



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**DETEKCE POŠKOZENÍ POVRCHU MATERIÁLU Z FO-
TOGRAFIE**

DETECTION OF MATERIAL SURFACE DAMAGE BASED ON A PHOTOGRAPH

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MAREK VALKO

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ DYK

BRNO 2023

Zadání bakalářské práce



146476

Ústav: Ústav inteligentních systémů (UITS)
Student: **Valko Marek**
Program: Informační technologie
Specializace: Informační technologie
Název: **Detekce poškození povrchu materiálu z fotografie**
Kategorie: Zpracování obrazu
Akademický rok: 2022/23

Zadání:

1. Prostudujte základy zpracování obrazu.
2. Vyberte vhodné metody a navrhňte řešení problému detekce poškození povrchu materiálu.
3. Navržené řešení implementujte v programovacím jazyce Python/C++.
4. Experimentujte s vaší implementací a případně navrhňte vlastní modifikace zvolených metod.
5. Porovnejte dosažené výsledky a navrhňte možnosti budoucího vývoje.

Literatura:

- ACHARYA, Tinku; RAY, Ajoy K. Image processing: principles and applications. John Wiley & Sons, 2005.
- Fišer M, Procházka S, Škvarek J. Hlavně palných zbraní. Univerzita obrany; 2006

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

Vedoucí práce: **Dyk Tomáš, Ing.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1.11.2022
Termín pro odevzdání: 10.5.2023
Datum schválení: 3.11.2022

Abstrakt

Táto záverečná práca sa zameriava na detekciu povrchových poškodení materiálu z fotografie s využitím techník počítačového videnia, konvolučných neurónových sietí a modelov pre detekciu objektov ako Faster R-CNN a YOLOv8. V práci sú podrobne popísané jednotlivé metódy detekcie povrchových poškodení, spracovanie obrazu, neurónové siete a strojové učenie. Práca porovnáva výkonnosť týchto modelov pri identifikácii povrchových poškodení na dreve a oceli s využitím rôznych augmentácií týchto dátových sád.

Abstract

This thesis focuses on the detection of surface defects from photograph using computer vision techniques, convolutional neural networks and models for object detection such as Faster R-CNN and YOLO. Different methods of surface damage detection, image processing, neural networks and machine learning are described in detail. The thesis also compares the performance of these models in identification of surface defects on wood and steel using different augmentations of these datasets.

Kľúčové slová

detekcia poškodení, yolov8, faster rcnn, počítačové videnie, automatická kontrola kvality, strojové učenie, konvolučné neurónové siete, extraktory príznakov

Keywords

damage detection, yolov8, faster rcnn, computer vision, automatic quality control, machine learning, convolutional neural networks, feature extractors

Citácia

VALKO, Marek. *Detekce poškození povrchu materiálu z fotografie*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Dyk

Detekce poškození povrchu materiálu z fotografie

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Tomáša Dyka. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Marek Valko

9. mája 2023

Podakovanie

Rád by som poďakoval svojmu vedúcemu práce Ing. Tomášovi Dykovi za pomoc, odborné vedenie, cenné rady a trpezlivosť.

Obsah

1	Úvod	2
2	Metódy detekcie povrchových poškodení	3
2.1	Kontrola kvality	3
2.2	Počítačové videnie	4
2.3	Spracovanie obrazu	7
3	Neurónové siete a strojové učenie	11
3.1	Umelé neurónové siete	11
3.2	Konvolučné neurónové siete	12
3.3	Metódy strojového učenia	15
3.4	Detektory založené na regiónoch	18
3.5	Jednokrokové detektory	21
4	Návrh riešenia	25
4.1	Zvolené modely	25
4.2	Trénovacie prostredie	25
4.3	Dátové sady	26
4.4	Metriky pre zrovnávanie	29
5	Experimenty	31
5.1	Porovnanie presnosti modelov na originálnych dátových sádach	31
5.2	Vplyv zmeny rozlíšenia dátových sád na presnosť modelov	34
5.3	Vplyv augmentácií dátových sád na presnosť modelov	35
6	Záver	40
	Literatúra	41
A	Obsah priloženej SD karty	46

Kapitola 1

Úvod

Jedným zo základných procesov vo výrobnom priemysle je kontrola kvality, ktorá zabezpečuje kvalitu vyrábaných výrobkov. Dôležitú úlohu pri tomto procese zohráva vizuálna kontrola kvality. Túto vizuálnu kontrolu zvyčajne ručne vykonávajú pracovníci vyškolení na identifikáciu komplexných povrchových chýb [28]. Avšak manuálna vizuálna kontrola je pracná a subjektívna, v mnohých prípadoch príliš pomalá a nákladná. V dôsledku toho sa v posledných desaťročiach dostávajú do popredia automatizované systémy vizuálnej kontroly. Medzi hlavné výhody takýchto systémov patria nestranné a reprodukovateľné výsledky kontroly, kompletná a detailná dokumentácia, rýchlosť kontroly a nižšie náklady [28].

V minulosti tieto systémy fungovali na princípe tradičných metód počítačového videnia, ktoré dokázali pokryť aspoň niektoré z problémov manuálnej vizuálnej kontroly [60]. V dnešnej dobe je potrebné aby bolo možné takúto metódu rýchlo adaptovať na nový výrobok a tradičné metódy počítačového videnia nie sú schopné poskytnúť takúto flexibilitu. Riešenie, ktoré umožňuje lepšiu flexibilitu a ponúka zníženie inžinierskeho úsilia je možné nájsť v metódach hlbokého učenia. Metódy hlbokého učenia sa učia potrebné funkcie priamo z konkrétnych dátových sád, čím sa pre inžiniera (programátora) znižuje úsilie pracne vytvárať takéto funkcie. Metódy hlbokého učenia v posledných rokoch dosiahli a prekonalí schopnosť ľudí riešiť úlohy spojené s obrazom, akými je napr. klasifikácia obrazu [11].

Táto práca sa zameriava na detekciu poškodenia materiálu z fotografie prostredníctvom rôznych techník počítačového videnia, konvolučných neurónových sietí a modelov pre detekciu objektov ako Faster R-CNN a YOLO. Cieľom je skúmať a porovnať výkon týchto metód pri identifikácii povrchových poškodení na drevných a oceľových materiáloch. V kapitole 2 je podrobne popísané spracovanie obrazu a jednotlivé metódy detekcie povrchových poškodení. V kapitole 3 sú následne opísané neurónové siete, strojové učenie a niektoré zo súčasťných detektorov objektov. Navrhnuté riešenie v kapitole 4 obsahuje výber vhodných modelov, opis tréningového prostredia, vybrané dátové sady a metriky na porovnávanie výsledkov experimentov. Výsledky experimentov v kapitole 5 nakoniec ukazujú účinnosť a potenciál navrhnutých metód v aplikáciách súvisiacich s kontrolou kvality materiálov.

Kapitola 2

Metódy detekcie povrchových poškodení

Táto kapitola sa zaoberá metódami detekcie povrchových poškodení a kontrolou kvality, pričom sa zameriava na využitie počítačového videnia. Sú tu predstavené metódy spracovania obrazu, ako je odstraňovanie šumu a normalizácia, a metódy spracovania obrazu, ako sú detektory hrán a rozpoznávanie obrazov. Jednotlivé metódy sú podrobne vysvetlené a popísané s cieľom poskytnúť prehľad o možnostiach, ktoré sú k dispozícii pre detekciu a povrchových poškodení z obrazových dát.

2.1 Kontrola kvality

Kontrola kvality je kritickým prvkom výrobných procesov, ktorý zabezpečuje, že výrobky sú vyrobené s presnosťou, spoľahlivosťou, kvalitou a spĺňajú očakávania zákazníkov. V rôznych odvetviach sa využívajú rôzne metódy kontroly kvality, od manuálnej kontroly až po pokročilé technológie založené na počítačovom videní a strojovom učení. Manuálna kontrola povrchových chýb je dôležitý, ale náročný proces, ktorý si vyžaduje vyškolený personál a veľa času. Navyše je tento proces náchylný na chyby, čo sťažuje zachovanie vysokej úrovne kvality. Ľudskí kontrolóri sú náchylní na únavu, čo môže časom viesť k nezrovnalostiam a zníženej presnosti tejto kontroly. Systémy automatickej vizuálnej detekcie povrchových chýb ponúkajú riešenie, ktoré dokáže tieto obmedzenia prekonať využitím pokročilých algoritmov a techník počítačového videnia na rýchlu a presnú identifikáciu chýb na obrázkoch. Táto technológia pomáha skrátiť čas potrebný na kontrolu a eliminuje možnosť ľudskej chyby. Jednou z najvýznamnejších výhod automatickej vizuálnej detekcie povrchových poškodení je jej schopnosť dôsledne a presne analyzovať poskytnuté dáta, čo vedie k lepšej spoľahlivosti a dôslednosti kontroly kvality.

Využitie automatickej kontroly kvality v praxi

Systémy automatickej kontroly kvality dokážu zabezpečiť monitorovanie výrobného procesu v reálnom čase, čo umožňuje okamžité odhalenie a odstránenie chýb. Takéto monitorovanie umožňuje včasnú reakciu a tým sa znižuje riziko následnej expedície chybných výrobkov. Okrem toho dokáže odhaliť aj tie najmenšie chyby, ktoré by mohli byť pri manuálnej kontrole prehliadnuté, čím sa zlepšuje celková kvalita výrobku.

Automatizácia detekcie povrchových chýb nájde uplatnenie aj v oblasti údržby a monitorovania infraštruktúry, ako sú cesty, mosty, tunely a budovy. Získavanie a digitalizácia

obrazu je v tomto kontexte ešte náročnejšia, pretože podmienky snímania sú často menej kontrolované a kvalita zachytených snímok môže byť ovplyvnená premenlivým osvetlením, poveternostnými podmienkami a faktormi prostredia. Pokročilé zobrazovacie technológie, akými sú LiDAR, multispektrálne a hyperspektrálne zobrazovanie umožňujú identifikovať širokú škálu chýb, ktoré nemusia byť viditeľné na tradičných fotografiách. Integrácia týchto technológií s algoritmi strojového učenia a umelej inteligencie ďalej zlepšuje možnosti detekcie a klasifikácie chýb v systémoch. Integrácia systémov detekcie poškodení s dronmi umožňuje efektívnejšie a lacnejšie kontroly veľkých a ťažko dostupných oblastí. Drony vybavené kamerami s vysokým rozlíšením alebo inými snímačmi môžu zachytávať snímky z rôznych uhlov a perspektív, čo poskytuje komplexné údaje na detekciu a analýzu chýb. Pri kontrole a údržbe infraštruktúry sa čoraz viac používa robotika v kombinácii s počítačovým videním a systémami na zisťovanie poškodení pretože robotické systémy majú prístup aj do človeku nedostupných a stiesnených priestorov. [34]

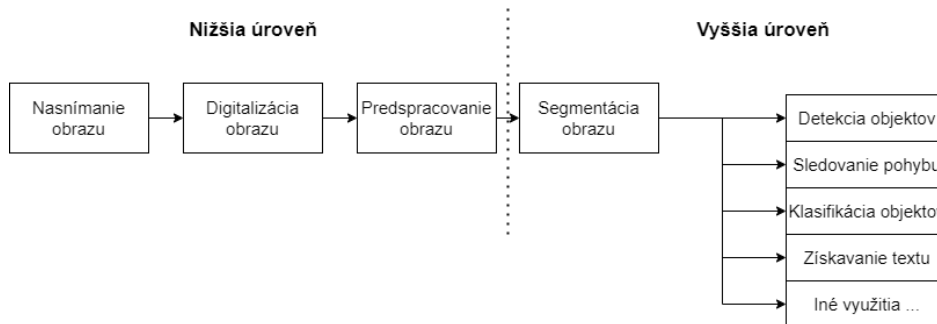
Systémy na detekciu povrchových chýb majú uplatnenie aj v oblasti leteckej a vesmírnej techniky [61]. V tejto oblasti sa používajú na detekciu a analýzu chýb na povrchu lietadiel, satelitov a iných vesmírnych lodí. Vzhľadom na extrémne podmienky, ktorým sú tieto objekty vystavené, je nutné predchádzať poškodeniam, ktoré by mohli mať vplyv na bezpečnosť týchto zariadení.

2.2 Počítačové videnie

Človek dokáže vnímať, rozpoznávať a následne reagovať na svoje okolie pomocou zmyslových orgánov. Jedným z najdôležitejších zmyslových orgánov je ľudský zrak, vďaka ktorému človek získava až 80% všetkých informácií zo svojho okolia [49]. Je preto pochopiteľné, že existuje stále väčšia snaha implementovať a vyvíjať túto vlastnosť aj v oblasti počítačovej techniky. Schopnosť vnímať a následne reagovať na svoje okolie predstavuje v počítačovej technike veľmi dôležitú úlohu.

Počítačové videnie je odvetvie, ktoré sa zaoberá získavaním dôležitých informácií zo zachyteného obrazu. Pod pojmom získať informácie je myslené správne rozoznanie skupiny objektov či javov a nasledovné využitie týchto informácií pre ďalšie spracovanie. Obrazové dáta môžu mať rôzne formy ako napríklad fotografie, video sekvencie, pohľady z viacerých kamier alebo viacrozmerné dáta z rôznych skenerov. Tieto dáta môžu byť uložené v pamäti počítača, alebo získané z pripojenej kamery. Spracovanie obrazu ako jedna z oblastí počítačového videnia má široké uplatnenie vo výrobnom priemysle, medicíne, vojenskom priemysle, bezpečnostných systémoch alebo dopravnom priemysle.

Pojem spracovanie obrazu predstavuje celý proces od digitalizácie nasnímaného obrazu až po konečné vyhodnotenie výsledkov. Systémy automatickej vizuálnej kontroly kvality zväčša pozostávajú zo štyroch základných komponentov: získanie a digitalizácia obrazu, pedspracovanie (angl. preprocessing), extrakcia príznakov a následná klasifikácia (obrázok 2.1).



Obr. 2.1: Celý proces spracovania obrazu.

2.2.1 Získanie a digitalizácia nasnímaného obrazu

Obraz sa dá definovať ako vizuálna scéna, ktorá je zachytená zobrazovacím systémom (napr. fotoaparátom) transformáciou z objektového (3D) priestoru objektu do obrazového (2D) priestoru. Obraz ako dvojrozmerný (2D) signál je matematicky popísaný spojitou alebo diskretnou obrazovou funkciou s dvoma premennými $f(x,y)$, kde x a y predstavujú rovinne priestorové súradnice a funkčná hodnota predstavuje hodnotu jasu v bode daných súradníc [29]. Pre spracovanie takéhoto obrazu digitálnymi metódami je potrebné najprv obraz zdigitalizovať.

Proces digitalizácie pozostáva z navzorkovania obrazu do matice bodov (pixelov) s rozmermi $M \times N$. Navzorkovaním obrazu sa zo spojitej funkcie stane funkcia diskretná. V súčasnosti sa všetko spracovanie obrazu vykonáva na počítači, takže obraz sa digitalizuje už v čase získavania údajov. Keďže je niekedy ťažké zabezpečiť linearitu prenosu signálu cez zobrazovací systém, obraz nemusí vždy presne zodpovedať reálnej scéne z hľadiska farieb, priestorového skreslenia a môže obsahovať šum alebo iné chyby. Tieto chyby môžu brániť kvalitnej interpretácii a vizuálnej analýze. Preto boli vyvinuté metódy, ktorými sa dajú tieto obrazové chyby potlačiť.

2.2.2 Predspracovanie obrazu

Predspracovanie obrazu je proces úpravy obrazu pred ďalšou analýzou a spracovaním. Jeho úlohou je pripraviť snímky na extrakciu a klasifikáciu príznakov. Cieľom predspracovania je zlepšiť kvalitu obrazu, odstrániť šum, skreslenie a napríklad tým zvýšiť presnosť detekcie poškodení. Výber správnych techník predspracovania obrazu môže v určitých prípadoch výrazne napomôcť úspešnosti detekcie povrchových chýb. Nasledujúca časť preto bližšie rozoberá niektoré z najpoužívanejších techník predspracovania obrazu.

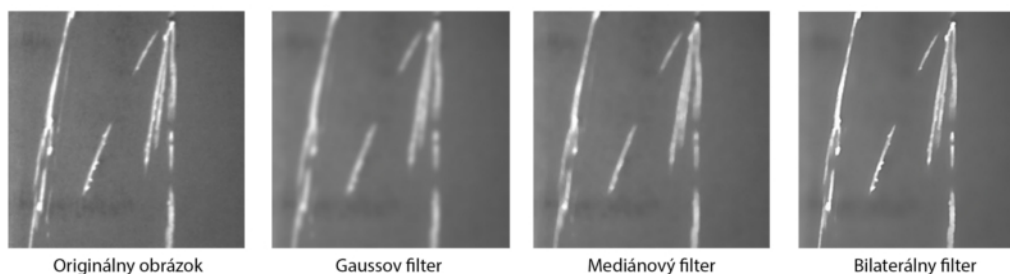
Odstránenie šumu

Hlavným cieľom filtrovania je zlepšiť kvalitu obrazu odstránením šumu a nežiaducich artefaktov, ale pri tom zachovať základné vlastnosti, ako sú hrany a textúry (obrázok 2.2). Nižšie sú zhrnuté najčastejšie používané typy filtrov. [16]

- **Gaussov filter** alebo Gaussovo vyhladzovanie, je lineárna filtračná technika, ktorá využíva Gaussovo jadro na konvolúciu so vstupným obrazom. Tento proces má za následok zníženie vysokofrekvenčného šumu a zároveň zachováva nízkofrekvenčné vlast-

nosti. Gaussovo filtrovanie sa využíva hlavne na vyhladzovanie obrazov bez toho, aby došlo k výraznému skresleniu alebo strate detailov.

- **Mediánový filter** je nelineárna metóda filtrovania, ktorá nahrádza hodnotu každého pixelu mediánom hodnôt jeho susedných pixelov. Táto technika je veľmi účinná pri odstraňovaní tzv. impulzného šumu a zároveň zachováva hrany a ostré prechody. Mediánové filtrovanie sa často používa pre svoju schopnosť zachovať štruktúru obrazu a zároveň účinne redukovať šum.
- **Bilaterálny filter** kombinuje priestorové informácie a informácie o intenzite na vyhladenie obrázkov. Na rozdiel od Gaussovho filtrovania, ktoré zohľadňuje len priestorovú vzdialenosť medzi pixelmi, bilaterálne filtrovanie berie do úvahy aj rozdiely v intenzite medzi pixelmi. Tieto dodatočné informácie pomáhajú zachovať hrany a jemné detaily v obraze a zároveň znižujú šum.



Obr. 2.2: Porovnanie filtrov.

Normalizácia

Normalizácia [16] je technika predspracovania obrazu, ktorá štandardizuje hodnoty intenzity obrazu. Cieľom je zabezpečiť, aby hodnoty intenzity spadali do určitého rozsahu, napríklad $<0, 1>$. Normalizácia pomocou transformácie intenzity obrazu zlepšuje konzistenciu a kompatibilitu obrazov pre následné kroky spracovania, ako je extrakcia a klasifikácia príznakov.

- **Min-max** normalizácia transformuje hodnoty intenzity obrazu tak, aby spadali do požadovaného rozsahu. Táto metóda mení každú hodnotu intenzity na základe lineárnej funkcie, ktorá mapuje pôvodný rozsah na cieľový rozsah. Ten môže byť definovaný ako rozsah od 0 do 1 alebo od 0 do 255. Normalizácia min-max zabezpečí výpočtom minimálnych a maximálnych hodnôt intenzity v obraze, že všetky hodnoty intenzity budú spadať do požadovaného rozsahu.
- **Vyrovňavanie histogramov** (angl. histogram equalization) je technika, ktorá upravuje hodnoty intenzity obrazu s cieľom dosiahnuť rovnomerné rozloženie intenzity. Na prerozdelenie intenzít sa používa kumulatívna distribučná funkcia (CDF) histogramu obrazu, ktorá tieto intenzity efektívne rozloží v celom rozsahu. Vyrovnanie histogramu má za následok zvýšenie kontrastu a zvýšenie viditeľnosti prvkov v obraze, čo zlepšuje celkovú kvalitu obrazu pre ďalšie kroky spracovania.

2.3 Spracovanie obrazu

Oblasť spracovania obrazu zastrešuje široké spektrum metód a algoritmov, ktoré umožňujú spracovávať fotografie a videá a využívať ich na rôzne účely. V súčasnosti je spracovanie digitálnych obrazov stále častejšie a má mnoho rôznych využití, od priemyselných procesov až po analýzu medicínskych obrazov. Nasledujúca časť je venovaná niektorým z najbežnejších technológií a algoritmov používaných v oblasti spracovania obrazu.

2.3.1 Detektory hrán

Detektory hrán umožňujú identifikovať v obraze oblasti, kde dochádza k rýchlemu zmenšovaniu intenzity a teda aj zmene kontrastu. Tieto oblasti sa vyskytujú na hranách objektov, kde nastáva prechod z jednej oblasti do druhej. Detekcia hrán preto hrá dôležitú úlohu pri rozpoznávaní tvarov a kontúr v obraze. V nasledujúcej časti sú bližšie porovnané najpoužívanejšie detektory hrán a ich princíp detekcie.

Cannyho detektor hrán

Cannyho detektor [4] je špeciálny typ detektora hrán v obraze, ktorý pozostáva z niekoľkých krokov. Tento komplexný algoritmus sa snaží maximalizovať pravdepodobnosť detekcie skutočných hrán a minimalizovať pravdepodobnosť detekcie falošných hrán spôsobených šumom. Jeho hlavnými kritériami správnej detekcie hrán sú:

- **Nízka chybovosť:** Cannyho detektor minimalizuje počet falošne detekovaných hrán spôsobeným šumom a maximalizuje pravdepodobnosť detekcie skutočných hrán.
- **Správna lokalizácia hrán:** detekované hrany by mali byť čo najbližšie k skutočným hranám, aby sa zabezpečila správna lokalizácia objektov v obraze.
- **Minimálny počet detekovaných hrán:** jedna skutočná hrana by nemala byť detekovaná viac než raz.

Samotný algoritmus Cannyho detektoru pozostáva z piatich krokov (obrázok 2.4):

1. **Vyhladzovanie obrazu:** Pri získavaní obrazových dát často dochádza k zašumeniu obrazu a preto je nutné tento šum odstrániť. Cannyho detektor tento problém odstraňuje využitím Gaussovho filtra. Filtrovanie je vykonané pomocou konvolúcie vstupného obrazu $f(x,y)$ a Gaussového jadra $G(x,y)$ nasledovne:

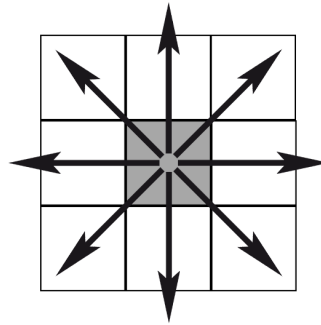
$$f_s(x, y) = G(x, y) * f(x, y) \quad (2.1)$$

2. **Získanie gradientu:** Hrana sa dá popísať ako vektor o určitom smere a veľkosti. V ďalšom kroku teda Cannyho detektor vypočíta veľkosť a smer gradientu v každom bode obrazu podľa rovníc 2.2 a 2.3. Výsledkom je obraz, ktorý reprezentuje veľkosť a smer gradientu.

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (2.2)$$

$$\vartheta(x, y) = \arctan\left(\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y}\right) \quad (2.3)$$

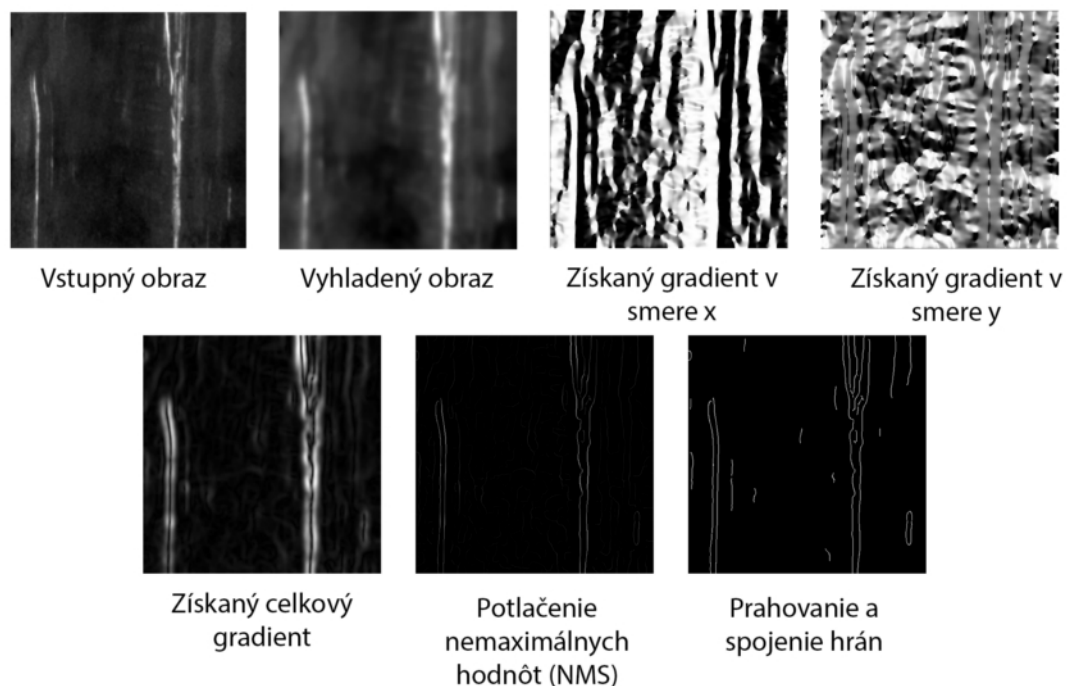
3. **Potlačenie nemaximálnych hodnôt:** Ďalším krokom je v podstate zúženie a doostrenie zistených hrán v obraze získanom z kroku 2. Toho sa dá dosiahnuť zachovaním lokálnych maxim hrany, t.j. hodnôt, ktoré najpresnejšie odpovedajú skutočnej hrane, keďže gradient je v tomto bode najväčší, a potlačením všetkých ostatných hodnôt [16]. Jedným zo spôsobov, ako túto operáciu vykonať, je zaokrúhliť smer gradientu (x,y) na najbližších 45° . Okolie jedného pixelu v každom 45° skoku je tvorené všetkými jeho priamo susediacimi pixelmi (obrázok 2.3).



Obr. 2.3: Susedné pixely a smery gradientov zaokrúhlené na 45° skoky.

Hodnota pixelu je vždy porovnávaná s hodnotami dvoch susediacich pixelov v smere aj proti smeru gradientu. Ak je hodnota vybraného pixelu väčšia ako hodnoty týchto dvoch susediacich pixelov, táto hodnota zostane zachovaná, inak je táto hodnota potlačená. Výsledkom tohto kroku je obraz detekovaných zúžených hrán.

4. **Dvojité prahovanie:** Je pravdepodobné, že medzi detekovanými hranami z predchádzajúceho kroku budú aj falošne detekované hrany, ktoré môžu byť spôsobené šumom. Pri tomto kroku sa využívajú dva prahy, T_H a T_L , pri ktorých platí že $T_H > T_L$. V prípade pixelov s hodnotou vyššou ako T_H sú tieto pixely označené ako tzv. silné hrany a zostanú ponechané. Pixely s hodnotou medzi T_H a T_L sú označené ako tzv. slabé hrany a tiež zostanú zachované, zatiaľ čo pixely s hodnotou menšou ako T_L sú potlačené.
5. **Spojenie hrán:** Posledným krokom Cannyho detektoru je spojenie detekovaných hrán. Keďže je nepravdepodobné, že by šum po prechádzajúcich krokoch mohol tvoriť silnú hranu, tieto hrany sú automaticky považované za skutočné hrany. Slabé hrany však môžu byť tvorené aj detekovaným šumom. Do výsledného obrazu sa preto zakomponujú len tie slabé hrany, ktorých pixely priamo susedia s pixelmi silných hrán.



Obr. 2.4: Ukážka algoritmu Cannyho detektoru hrán.

2.3.2 Rozpoznávanie obrazov

Cieľom procesu rozpoznávania obrazov (angl. image recognition) je identifikácia a kategorizácia objektov alebo vzorov v digitálnych obrazoch. Tento proces analyzuje obraz a rozpozná špecifické vlastnosti, ktoré umožňujú počítačovému systému rozlíšiť jeden objekt od druhého. Rozpoznávanie obrazov je dôležitou súčasťou oblasti počítačového videnia a má mnoho praktických využití ako napríklad analýza lekárskeho obrazov ako sú röntgenové snímky, detekcia ciest a prekážok v systémoch autonómnych vozidiel alebo práve detekcia poškodení v procese kontroly kvality. Medzi najčastejšie používané techniky patria klasifikácia obrazu, detekcia objektov, sémantická segmentácia a segmentácia inštancií.

Klasifikácia obrazu

Klasifikácia obrazu [26] je proces kategorizácie obrázkov do rôznych tried alebo kategórií. Táto technika sa zvyčajne používa v prípadoch, keď je potrebné identifikovať jednoduché a dobre definované objekty, ako sú napríklad zvieratá, rastliny alebo dopravné prostriedky. Klasifikácia obrazu spočíva v naučení modelu na trénovacej dátovej sade, kde každý obrázok je označený správnou kategóriou. Keď je model natrénovaný, môže sa použiť na predpovedanie triedy nových obrázkov.

Detekcia objektov

Detekcia objektov je pokročilejšia technika, ktorá na rozdiel od klasifikácie obrazu nielen identifikuje objekty na obrázku, ale ich aj lokalizuje. Hlavnou úlohou detekcie objektov je lokalizovať všetky prípady vopred definovaných tried a poskytnúť ich približné umiestnenie v rámci obrazu alebo videa [63]. Táto technika je užitočná najmä v prípadoch, kde

je dôležité poznať polohu objektu v obraze, napríklad v samojazdiacich autách alebo bezpečnostných systémoch. Trénovacia sada obsahuje označenia (angl. labels), ktoré určujú pravdivé umiestnenie všetkých objektov rôznych tried v rámci obrázkov.

Názorná ukážka klasifikácie obrazu a detekcie objektov je zobrazená na obrázku 2.5.



Obr. 2.5: Porovnanie klasifikácie obrazu a detekcie objektov, upravené z [25].

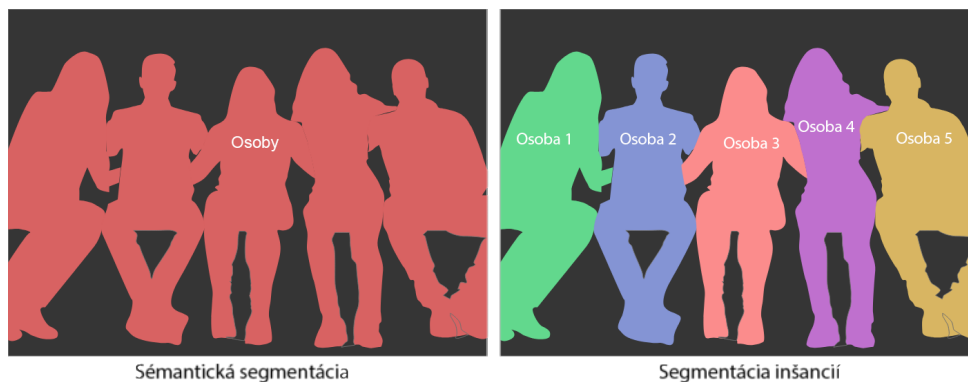
Sémantická segmentácia

Sémantická segmentácia [10] je proces, ktorý rozdeľuje obraz na rôzne oblasti (segmenty) na základe významu alebo kontextu pixelov. Cieľom sémantickej segmentácie je priradiť každému pixelu v obraze vopred definovanú triedu ako napríklad budova alebo strom. Sémantická segmentácia nerozlišuje rôzne výskyty tej istej triedy.

Segmentácia inštancií

Segmentácia inštancií je zložitejšia technika, ktorá kombinuje úlohy detekcie objektov a sémantickej segmentácie. Segmentácia inštancií nielen identifikuje a lokalizuje objekty v obraze, ale aj ich segmentuje na jednotlivé inštancie alebo objekty [46]. Na rozdiel od sémantickej segmentácie, ktorá zoskupuje pixely na základe ich spoločného sémantického významu sú algoritmy segmentácie inštancií navrhnuté tak, aby segmentovali každý výskyt objektu v obraze samostatne, aj keď sa v obraze vyskytuje viacero inštancií rovnakej triedy objektov.

Rozdiel medzi sémantickou segmentáciou a segmentáciou inštancií je znázornený na obrázku 2.6.



Obr. 2.6: Porovnanie sémantickej segmentácie a segmentácie inštancií, upravené z [7].

Kapitola 3

Neurónové siete a strojové učenie

V posledných rokoch dochádza k rapídному rozvoju oblasti umelej inteligencie a strojového učenia, ktoré sa stávajú stále dôležitejšími nástrojmi v mnohých odvetviach akým je aj spracovanie obrazu. Táto kapitola sa zameriava na popis fungovania rôznych typov neurónových sietí a metód strojového učenia, ktoré sa využívajú v detekcii povrchových chýb. Najviac pozornosti je venované konvulčným neurónovým sieťam, ktoré sú jedným z najpoužívanejších typov neurónových sietí v oblasti spracovania obrazu. Okrem toho sú v tejto kapitole opísané aj rôzne metódy strojového učenia, ako sú učenie s učiteľom a bez učiteľa a taktiež je vysvetlený princíp detektorov založených na regiónoch a jednokrokových detektorov.

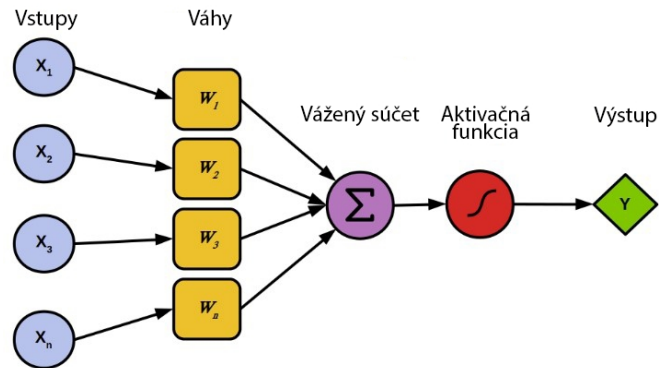
3.1 Umelé neurónové siete

Umelé neurónové siete (ANN) sú výpočtové modely inšpirované štruktúrou a fungovaním biologických neurónových sietí. Sú zložené z prepojených uzlov alebo neurónov, organizovaných do vrstiev, ktoré spracúvajú a prenášajú informácie na riešenie rôznych typov úloh, ako je napr. rozpoznávanie vzorov alebo spracovanie prirodzeného jazyka. ANN sa dokážu naučiť rozpoznávať zložité vzory, dokážu sa bez explicitného programovania prispôbiť novým informáciám a dokážu čerpať z predchádzajúcich skúseností.

Koncept ANN prvýkrát predstavili Warren McCulloch a Walter Pitts v roku 1943, keď navrhli zjednodušený model biologických neurónov, tzv. McCulloch-Pittsov neurón [41]. Neskôr pokračoval rozvoj ANN predstavením algoritmov učenia neurónových sietí, ako je perceptrónové učenie Franka Rosenblatta v roku 1958 [47] a algoritmus spätného šírenia chyby [48], ktoré predstavili v roku 1986 David Rumelhart, Geoffrey Hinton a Ronald Williams.

3.1.1 Perceptrón

Perceptron sa skladá z viacerých vstupov, váh, aktivačnej funkcie a jedného výstupu. V perceptróne sa každý vstup vynásobí príslušnou váhou a potom sa všetky tieto vážené súčiny spočítajú. Konečný výstup perceptronu určuje aktivačná funkcia, ktorá upravuje výsledný súčet. Perceptrón najčastejšie používa krokovú aktivačnú funkciu, ktorá na základe prahovej hodnoty posiela na výstup 0 ak je jej vstup menší ako táto prahová hodnota alebo 1 ak je jej vstup väčší ako prahová hodnota. Často sa využívajú aj iné aktivačné funkcie, ako napríklad hyperbolický tangens alebo sigmoid.



Obr. 3.1: Schéma perceptrónu.

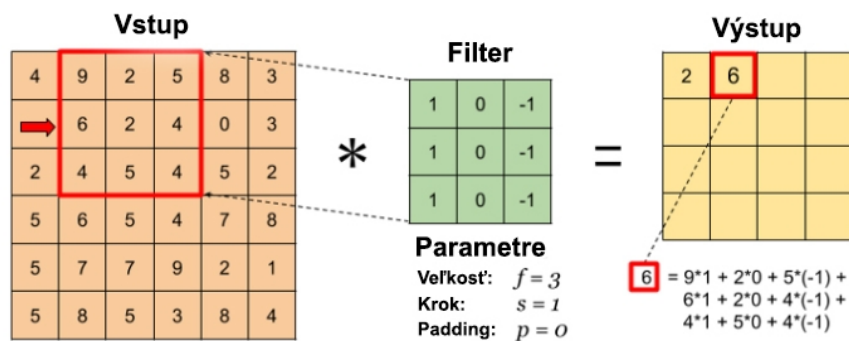
Perceptróny možno využiť na jednoduché klasifikačné úlohy, kde sú údaje lineárne oddeľiteľné. Ich schopnosť naučiť sa zložitejšie, lineárne neoddeľiteľné vzory je však obmedzená. Na riešenie zložitejších problémov boli vyvinuté komplexnejšie architektúry neurónových sietí [40] ako napríklad dopredné neurónové siete (feed-forward neural networks), rekurentné neurónové siete alebo konvolučné neurónové siete.

3.2 Konvolučné neurónové siete

Konvolučné neurónové siete (CNN) sú jedným z typov umelých neurónových sietí. Konvolučné neurónové siete majú rovnako ako klasické neurónové siete parametre (váhy), ktorých hodnoty sa učením menia. Ich architektúra sa skladá z vrstiev, ktoré obsahujú neuróny. Každý neurón počíta skalárny súčin svojho vstupu a parametrov a následne na základe nelineárnej funkcie aktivuje svoj výstup. V porovnaní s klasickými neurónovými sieťami obsahujú tzv. konvolučnú vrstvu, v dôsledku ktorej majú menej parametrov a výrazne menej spojení medzi vstupnými neurónmi, čo znamená jednoduchšie tréningovanie. Konvolučná neurónová sieť sa zvyčajne skladá z niekoľkých vrstiev, z ktorých každá plní špecifickú funkciu v celkovej činnosti siete. V nasledujúcich podkapitolách sú opísané hlavné typy vrstiev, ktoré sa nachádzajú v CNN.

3.2.1 Konvolučná vrstva

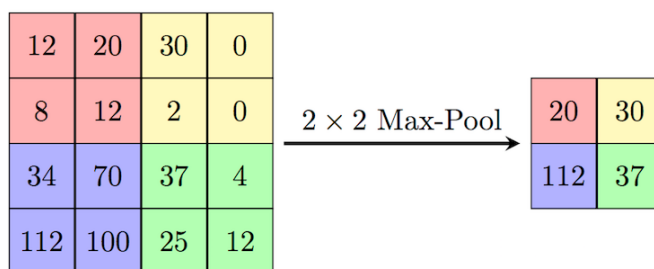
Parametre konvolučnej vrstvy pozostávajú zo sady filtrov, ktorých hodnoty sú predmetom učenia. Filter má definovanú šírku, výšku a hĺbku. Napríklad typický filter na prvej vrstve konvolučnej siete môže mať veľkosť $5 \times 5 \times 3$ (t. j. 5 pixelov na šírku a výšku a hĺbku 3 - farebné kanály RGB/HSV). Každý filter sa posúva (konvuluje) po vstupnej matici a počíta skalárny súčin medzi hodnotami filtra a vstupom na každej pozícii. Posúvaním filtra sa vytvorí dvojrozmerná aktivačná mapa, ktorá udáva odozvy daného filtra. Sieť sa intuitívne naučí filtre, ktoré sa aktivujú pri špecifickej situácii - napríklad na prvej vrstve to môže byť hrana alebo škvrna určitej farby, pri vyšších vrstvách to už môžu byť celé vzory pripomínajúce včelie plásty. Výsledkom tejto vrstvy je sada filtrov na každej konvolučnej vrstve (napríklad 12 filtrov) a každý z nich vytvára samostatnú dvojrozmernú aktivačnú mapu.[53]



Obr. 3.2: Operácia konvolúcie.

3.2.2 Pooling vrstva

Úlohou pooling vrstvy je postupne zmenšiť rozmery vstupu. Typickým spôsobom ako toto dosiahnuť je tzv. max-pooling [53]. Najčastejšou a najefektívnejšou formou je pooling vrstva s filtermi s veľkosťou 2x2 a krokom 2. Pri tejto operácii sa z oblasti 2x2 vyberie z každej oblasti maximum a následne sa z týchto hodnôt vytvorí nová výstupová matica.



Obr. 3.3: Max-pooling (2x2), prevzaté z [1].

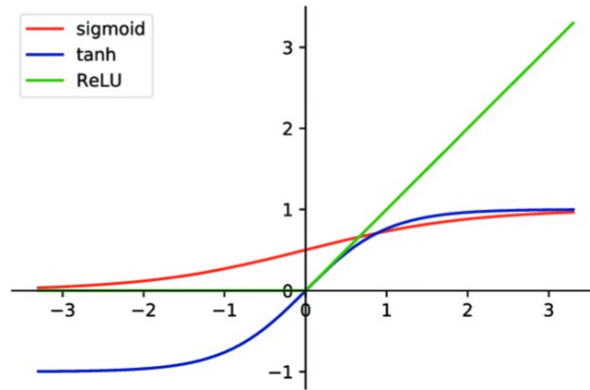
3.2.3 Plne prepojená vrstva

V plne prepojenej vrstve je každý neurón spojený so všetkými neurónmi z predchádzajúcej vrstvy. Táto vrstva je typická pre klasické neurónové siete, ale taktiež sa vyskytuje aj v konvolučných neurónových sieťach. Plne prepojené vrstvy bývajú náročné na výpočetný výkon pretože obsahujú veľmi vysoký počet spojení.

3.2.4 Aktivačná funkcia

Aktivačná funkcia určuje výstupné hodnoty jednotlivých neurónov a vnáša do modelu nelinearitu. V súčasnosti je v konvolučných sieťach najpoužívanejšou funkcia ReLU (Rectified Linear Unit), ktorá dokáže zastaviť tréning určitéch neurónov, čím výrazne zrýchľuje tréning vďaka jej nulovej derivácii pre záporné vstupy. Definícia funkcie ReLU je uvedená v rovnici 3.1. Historicky bola často používaná aj sigmoidová funkcia a hyperbolický tangens.

$$f(x) = \max(0, x) \tag{3.1}$$



Obr. 3.4: Aktivačné funkcie.

3.2.5 Stratová funkcia

Stratová funkcia (angl. loss function) je matematická funkcia, ktorá určuje rozdiel medzi predpovedaným výstupom a skutočným výstupom (cieľom) pre daný vstup. Hlavným cieľom v strojovom učení je minimalizovať stratovú funkciu, čo vedie k presnejším predpovediam a lepšej výkonnosti modelu. Stratové funkcie hrajú kľúčovú rolu pri tréňovaní neurónových sietí a iných modelov strojového učenia, pretože poskytujú metriku na optimalizáciu počas procesu učenia. Stratové funkcie možno všeobecne rozdeliť na dva typy: regresné stratové funkcie a klasifikačné stratové funkcie. Regresné stratové funkcie sa zväčša používajú, keď je cieľom predpovedať spojité hodnoty ako napríklad cenu nehnuteľností alebo teplotu, zatiaľ čo klasifikačné stratové funkcie sa používajú, keď je cieľom predpovedať diskkrétne triedy, napríklad určenie, či je na obrázku pes alebo mačka.

Správny výber stratovej funkcie môže byť kriticky dôležitý pre úspešné tréňovanie modelov strojového učenia, preto je dôležité poznať základné typy a ich použitie pre rôzne druhy problémov. Nasleduje príklad najbežnejších stratových funkcií.[42]

1. **Absolútna chyba (L1 Loss):** Používa sa pri regresných problémoch a vypočítava priemerný absolútny rozdiel medzi predpovedanými a skutočnými hodnotami. Absolútna chyba je definovaná rovnicou 3.2, kde n vyjadruje počet bodov, \hat{y}_i vyjadruje skutočnú hodnotu bodu a y_i vyjadruje predpovedanú hodnotu vrátenú modelom.

$$L1 = \frac{1}{n} \sum_{i=1}^n |y - \hat{y}_i| \quad (3.2)$$

2. **Stredná kvadratická chyba (MSE):** Používa sa pri regresných problémoch a vypočítava priemerný štvorcový rozdiel medzi predpovedanými a skutočnými hodnotami. MSE je popísaná rovnicou 3.3, kde n vyjadruje počet bodov, \hat{y}_i vyjadruje skutočnú hodnotu bodu a y_i vyjadruje predpovedanú hodnotu vrátenú modelom.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.3)$$

3. **Strata krížovej entropie (Cross-Entropy Loss):** Používa sa pri klasifikačných problémoch a meria rozdiel medzi dvoma rozdeleniami pravdepodobnosti, zvyčajne predpovedaným rozdelením pravdepodobnosti a skutočným rozdelením pravdepodobnosti.

Túto stratovú funkciu opisuje rovnica 3.4, kde $P^*(i)$ je skutočné rozdelenie pravdepodobnosti, $P(i)$ je predpovedané rozdelenie pravdepodobnosti a n značí počet tried.

$$Loss_{CEL} = -\frac{1}{n} \sum_{i=1}^n P^*(i) \log P(i) \quad (3.4)$$

4. **Hinge Loss:** Bežne sa používa pre metódu podporných vektorov (SVM) a niektoré ďalšie klasifikačné problémy, meria vzdialenosť medzi predpovedanými a skutočnými označeniami tried. Výpočet tejto vzdialenosti je vyjadrený rovnicou 3.5, kde n je počet bodov, \hat{y}_i predstavuje skutočnú hodnotu bodu a y_i predstavuje predpovedanú hodnotu vrátenú modelom.

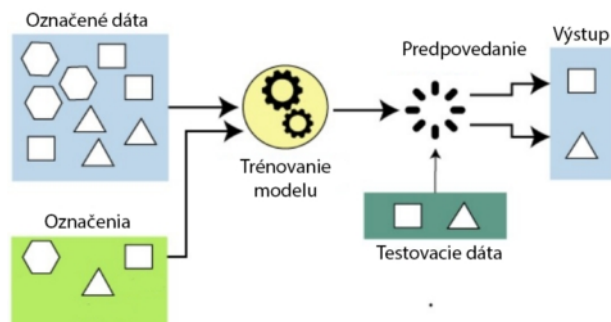
$$Loss_H = \frac{1}{n} \sum_{i=1}^n \max(0, 1 - \hat{y}_i y_i) \quad (3.5)$$

3.3 Metódy strojového učenia

Strojové učenie (angl. machine learning) je v súčasnosti vďaka zvýšenej dostupnosti dát, výkonnejším výpočtovým technológiám a vývoju nových algoritmov jednou z najrýchlejšie sa rozvíjajúcich oblastí v oblasti umelej inteligencie. Strojové učenie sa zaoberá algoritmami a technikami, ktoré umožňujú počítačom prispôbiť sa novým situáciám bez explicitného programovania. Medzi najpoužívanejšie metódy strojového učenia patria učenie s učiteľom (angl. supervised learning), učenie bez učiteľa (angl. unsupervised learning), učenie s čiastočným dohľadom (angl. semi-supervised learning) a učenie formou odmeňovania (angl. reinforcement learning).

3.3.1 Učenie s učiteľom

V prípade učenia s učiteľom sa dátová sada skladá z párových dvojíc vstup-výstup, pričom výstup sa často označuje ako label (označenie) alebo target (cieľ). Diagram takéhoto učenia je znázornený na obrázku 3.5. Neurónová sieť sa učí mapovať vstupy na správne výstupy minimalizovaním chyby medzi svojimi predpoveďami a skutočnými cieľovými hodnotami. Počas procesu tréningu sa upravujú váhy siete s cieľom zlepšiť jej výkonnosť na tréningových údajoch. Po natrénovaní sa sieť môže použiť na vytváranie predpovedí na nových dátach, ktoré neboli súčasťou tréningovej dátovej sady [26]. Učenie s učiteľom sa bežne používa v úlohách, ako je klasifikácia obrazov, rozpoznávanie reči a spracovanie prirodzeného jazyka. V tejto práci je využitý práve tento prístup. Vstupom bude fotografia, ktorá bude obsahovať rôzne triedy poškodení.



Obr. 3.5: Diagram učenia s učiteľom, upravené z [30].

3.3.2 Učenie bez učiteľa

Učenie bez učiteľa označuje techniku učenia, ktorá nevyužíva párové dvojice vstup-výstup, ani žiadne označené (cieľové) údaje. Cieľom je identifikovať základné štruktúry, vzory alebo korelácie vo vstupných dátach bez toho, aby sa spoliehala na vopred definované výstupné označenia. Techniky učenia bez učiteľa je možné využiť na rôzne účely, napríklad na zhľukovanie, redukciu dimenzionality [20] alebo extrakciu príznakov zo vstupných údajov. Príkladmi algoritmov učenia bez dohľadu sú k-means, analýza hlavných komponentov (PCA) a autoenkodéry.

3.3.3 Učenie so spätnou väzbou

Učenie so spätnou väzbou je technika, kde sa na začiatku vytvorí tzv. agent, ktorý následne interaguje s prostredím a učí sa robiť rozhodnutia. Agent za vykonané akcie v prostredí dostáva spätnú väzbu vo forme odmien alebo trestov. Cieľom je dostať sa z aktuálneho stavu prostredia do stavu, ktorý časom maximalizuje odmenu. Učenie s posilňovaním sa od učenia pod dohľadom líši tým, že sa nespolieha na vopred definovanú dátovú sadu ale namiesto toho sa agent učí metódou pokus-omyl a postupne zlepšuje svoj výkon prostredníctvom skúseností. Medzi algoritmy posilňovania učenia patria metódy Q-learning, Deep Q-Networks (DQN) a policy gradient [54]. Takáto forma učenia sa využíva pri robotike a autonómnych systémoch riadenia vozidiel.

3.3.4 Učenie s čiastočným dohľadom

Učenie s čiastočným dohľadom je kombinácia učenia s učiteľom a učenia bez učiteľa, pričom počas procesu učenia využíva údaje s označenými pravdivými výstupmi ako aj neoznačené údaje. Získanie označených údajov je často nákladné, pracné alebo časovo náročné, zatiaľ čo neoznačených údajov je často dostatok. Cieľom algoritmov učenia s čiastočným dohľadom je využiť informácie z neoznačených údajov na zlepšenie výkonu modelu na označených údajoch [6]. Medzi techniky používané pri čiastočne kontrolovanom učení patrí tzv. self-training (samo-štúdium), co-training (spoločné školenie) a metódy založené na grafoch.

Self-training

Pri self-trainingu [64] sa počiatočný klasifikátor trénuje na malej dátovej sade, ktorá obsahuje údaje s označenými pravdivými výstupmi. Tento klasifikátor sa neskôr použije na

predpovedanie označení pre neoznačené údaje. Po počiatočnom natrénovaní sa predpovede s najvyšším skóre dôveryhodnosti pridajú do trénovanej množiny údajov spolu s predpovedanými štítkami (cieľovými výstupmi). Klasifikátor sa potom znovu trénuje na tomto rozšírenom súbore údajov a proces sa iteratívne opakuje až kým sa nesplní kritérium pre zastavenie. Toto kritérium môže byť napríklad dosiahnutý maximálny počet iterácií alebo situácia, kedy za určitý počet iterácií nenastane žiadne výrazné zlepšenie výkonu modelu. Hlavným predpokladom pri self-trainingu je, že predpovede modelu s vysokým skóre dôveryhodnosti sú pravdepodobne správne, čo umožňuje zlepšiť jeho výkonnosť začlenením dodatočných označených údajov do trénovacej dátovej sady.

Co-training

Co-training [64] sa skladá z dvoch samostatných klasifikátorov, ktoré sú nezávisle od seba trénované na jednej označenej dátovej sade na základe dvoch rôznych príznakov. Napríklad by sa jeden klasifikátor mohol trénovať na základe farebných histogramov obrázkov, zatiaľ čo iný klasifikátor by sa mohol trénovať na základe textúrnych vlastností. Po počiatočnom trénovaní sa oba klasifikátory použijú na predpovedanie cieľových označení pre neoznačené údaje. Pre každý klasifikátor sa vyberie podmnožina predpovedí s najväčším skóre dôveryhodnosti a následne sa táto podmnožina pridá do trénovacej dátovej sady. Oba klasifikátory sa potom znovu trénujú na svojich rozšírených dátových sadách a proces sa iteratívne opakuje. Základným predpokladom pri spoločnom trénovaní je, že obe množiny príznakov poskytujú doplnkové informácie a oba klasifikátory môžu vzájomne zlepšiť ich výkonnosť zdieľaním svojich predpovedí s vysokým skóre dôveryhodnosti.

3.3.5 Prenesené učenie

Keďže trénovanie neurónovej siete môže byť časovo veľmi náročný proces, ktorý v niektorých prípadoch nemusí viesť k želaným výsledkom, používa sa ako „skratka“ prenesené učenie (angl. transfer learning). Prenesené učenie je technika, pri ktorej sa vopred natrénovaná neurónová sieť používa ako východiskový bod na trénovanie nového modelu na inej, ale príbuznej úlohe. Namiesto vytvárania modelu s náhodne inicializovanými váhami, ktoré sa neskôr trénujú, môže byť časť modelu inicializovaná s váhami už predtrénovanými na veľkých dátových sadách, ako je napríklad ImageNet. Využitím znalostí získaných z predtrénovaného modelu sa pri prenesenom učení skraca čas trénovania a znižujú sa požiadavky na množstvo dát potrebných pre novú úlohu, čo často vedie k zlepšeniu výkonu.

Hlavným princíp preneseného učenia je, že neurónové siete sa učia hierarchické príznaky, kde nižšie vrstvy zachytávajú nízkoúrovňové vlastnosti, akými sú hrany a textúry, zatiaľ čo vyššie vrstvy zachytávajú vlastnosti, ktoré sú špecifické pre danú úlohu. Pri úlohách, ktoré sú si podobné sa často dobre zovšeobecňujú nízkoúrovňové vlastnosti, čo umožňuje prenos znalostí medzi modelmi. Prenesené učenie sa zvyčajne využíva pri extrakcii príznakov (angl. feature extraction) alebo doladovaní neurónovej siete (angl. fine-tuning):

1. **Extrakcia príznakov:** Predtrénovaná sieť sa používa ako fixný extraktor príznakov a na základe týchto príznakov sa trénuje nový klasifikátor.
2. **Doladovanie:** Vopred natrénovaná sieť sa jemne doladí na novej úlohe, čo umožňuje prispôbenie sa charakteristikám špecifickým pre danú úlohu.

Extraktor príznakov ResNet a ResNetv2

Sieť ResNet vznikla v roku 2015, kedy sa skupina výskumníkov okolo Kaiminga Hea zaoberala otázkou, ako trénovať hlboké neurónové siete, ktoré by mali byť schopné dosahovať ešte vyššej presnosti pri klasifikácii a detekcii objektov v obrázkoch [18]. Jedným z problémov, ktorým sa výskumníci v tej dobe zaoberali, bol problém miznúceho gradientu. Tento problém sa týkal hlbokých sietí a spočíval v tom, že gradienty prechádzajúce sieťou sa s každou ďalšou vrstvou zmenšovali a mohli tak viesť k pomalému učeniu alebo k tzv. zaseknutiu sa v lokálnych extrémoch. He a jeho kolegovia navrhli riešenie tohto problému pomocou tzv. zvyškových spojení, ktoré umožňujú prenášať informácie o stave predošlej vrstvy cez aktuálnu vrstvu priamo na nasledujúcu vrstvu. Vďaka tomu je možné trénovať oveľa hlbšie siete, bez toho aby trpela kvalita výstupu alebo výpočtová náročnosť.

Hlavným vylepšením v ResNetv2 je zavedenie tzv. predaktivácie reziduálnych jednotiek a mapovania identity, čím sa zlepšil tok informácií a gradientov cez sieť. Mapovanie identity spočíva v tom, že zvyškové spojenia v reziduálnych blokoch prechádzajú priamo zo vstupu na výstup bez akejkoľvek úpravy, čo umožňuje sieti efektívnejšie sa učiť reziduálne funkcie [19]. Okrem toho ResNetv2 používa aj nové typy normalizačných vrstiev, vrátane tzv. *batch renormalizácie* [27], ktorá umožňuje lepšiu distribúciu parametrov siete počas tréningu.

Ďalšie známe extraktory príznakov

- **VGG (Visual Geometry Group)**: Sieť VGG, ktoré v roku 2014 predstavili Simonyan a Zisserman v článku [50], sa vyznačujú jednoduchou, ale hlbokou architektúrou pozostávajúcou z viacerých konvolučných vrstiev, po ktorých nasledujú max-pooling vrstvy.
- **Inception (GoogLeNet)**: Architektúra Inception, ktorú navrhol Szegedy a spol. v roku 2014 v článku [55], využíva viacero paralelných konvolučných vrstiev s rôznymi veľkosťami jadier na zlepšenie schopnosti siete zachytávať vlastnosti v rôznych mierkach.
- **DenseNet**: Systém DenseNet, ktorý predstavili Huang a spol. v roku 2016 v článku [22] spája každú vrstvu s každou ďalšou vrstvou spôsobom „feed-forward“, čo umožňuje efektívne opakované použitie funkcií a znižuje počet parametrov.
- **MobileNet** MobileNet vyvinutý Howardom a spol. v roku 2017 v článku [21] využíva hlbkovo oddeliteľné konvolúcie na zníženie výpočtovej zložitosti siete, vďaka čomu je vhodný pre prostredia s obmedzenými zdrojmi ako sú napríklad mobilné a „embedded“ zariadenia.

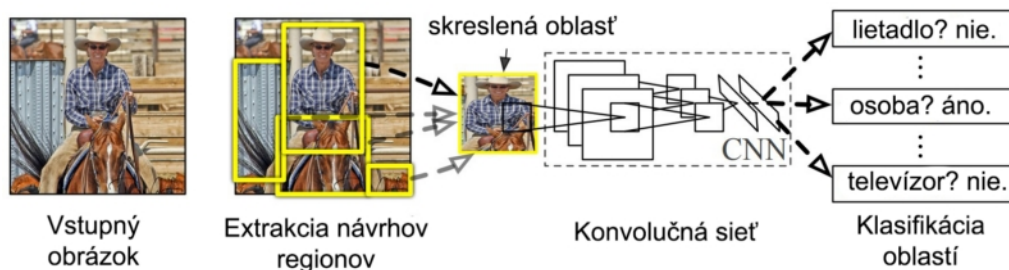
3.4 Detektory založené na regiónoch

Detektory založené na regiónoch sú v oblasti počítačového videnia často využívanou technikou na detekciu objektov v obraze. Tento typ detektorov môžeme rozdeliť do dvoch hlavných fáz. V prvej fáze sa na základe určitých prahových hodnôt a šablón vygenerujú z obrazu návrhy regiónov, ktoré by mohli obsahovať hľadaný objekt. V druhej fáze sa tieto regióny spracúvajú pomocou konvolučnej neurónovej siete, kde sa v týchto navrhnutých regiónoch detekujú a klasifikujú jednotlivé objekty. Tento prístup k detekcii objektov má vysokú presnosť, avšak zároveň aj vysokú výpočtovú náročnosť, pretože je nutné prechádzať každý vygenerovaný región.

3.4.1 R-CNN

Konvolučná neurónová sieť založená na regióne (R-CNN) je model na detekciu objektov v obrazových dátach, ktorý využíva vysoko-kapacitné konvolučné neurónové siete. Tento prístup pozostáva z dvoch krokov. V prvom kroku sa vygenerujú návrhy regiónov v obrázku, v ktorých sa predpokladá, že sa nachádza objekt. Na identifikáciu kandidátov na regióny sa používa algoritmus selektívneho vyhľadávania [56], ktorý z každého regiónu extrahuje vlastnosti určené na klasifikáciu. V druhom kroku sa každý návrh regiónu použije ako vstup do konvolučnej neurónovej siete, ktorá vyhodnotí prítomnosť objektu v danom regióne a zaradí ho do jednej z kategórií. Výstup každej CNN sa vloží do supervektora, ktorý použije na klasifikáciu algoritmus Support Vector Machine (SVM) [15]. Na konci tohto procesu sa použije lineárna regresia na presnejšie určenie ohraničujúceho rámčeka, ktorý objekt obklopuje. Fungovanie takejto siete je znázornené na obrázku 3.6.

Hoci R-CNN poskytuje relatívne vysokú presnosť pri detekcii objektov, existujú aj nevýhody, ktoré treba brať do úvahy. Trénovanie R-CNN je veľmi pomalé a výpočtovo náročné, pretože je potrebné prejsť každý návrh regiónu. Okrem toho, R-CNN potrebuje veľa miesta na disku a inferencia tiež trvá relatívne dlho a je výpočtovo náročná.

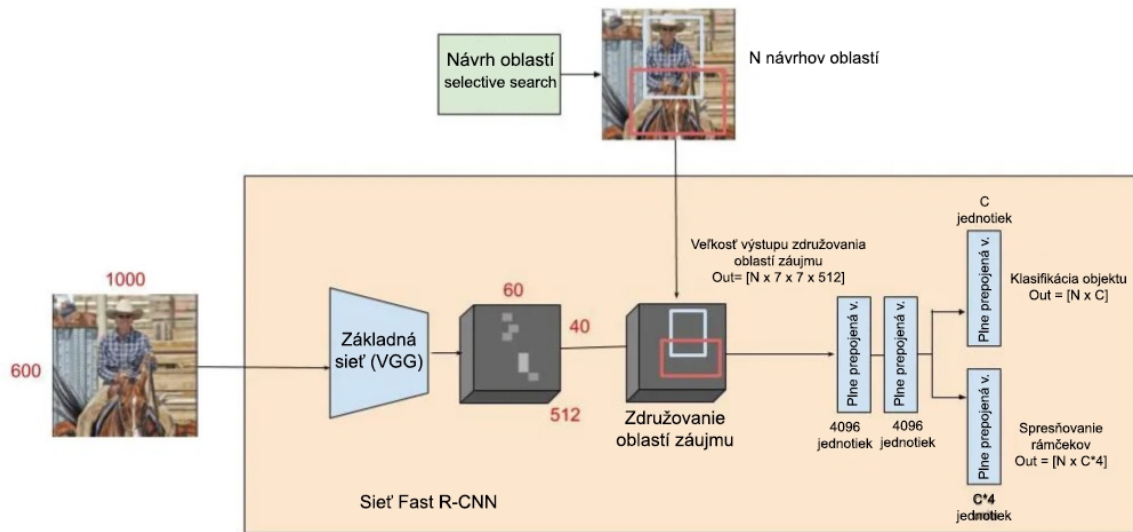


Obr. 3.6: Fungovanie siete R-CNN, upravené z [15].

3.4.2 Fast R-CNN

Tvorca R-CNN, Ross Girshick, vyvinul aktualizovanú verziu modelu s názvom Fast R-CNN [14], ktorej cieľom bolo zvýšiť rýchlosť detekcie objektov. Hoci základná myšlienka R-CNN zostala rovnaká, Fast R-CNN výrazne zlepšil a zrýchlil proces detekcie. Namiesto toho, aby sa CNN spúšťal nad každou oblasťou, Fast R-CNN berie celý obrázok a spúšťa CNN nad ním. Na ďalšie urýchlenie procesu bola zavedená nová technika nazvaná združovanie oblastí záujmu (angl. RoI pooling). Táto technika funguje tak, že oblasť záujmu sa rozdelí na sieť rovnako veľkých čiastkových okien a potom sa na každom čiastkovom okne vykoná nezávisle max pooling. To umožňuje efektívne spracovanie každej oblasti záujmu a predstavuje zlepšenie oproti predchádzajúcim metódam, ktoré vyžadovali samostatný priechod neurónovej siete pre každú oblasť.

Na odstránenie potreby vytvárania nového modelu bol algoritmus Support Vector Machine nahradený vrstvou softmax. Táto vrstva rozširuje sieť na predpovede a zapuzdruje regresiu, čo znamená, že Fast R-CNN používa len jeden model namiesto predchádzajúcich troch modelov (CNN, SVM, regresor). Zavedenie vrstvy softmax tiež zlepšilo presnosť Fast R-CNN v porovnaní s jeho predchodcom. Štruktúra siete Fast R-CNN je znázornená na obrázku 3.7.



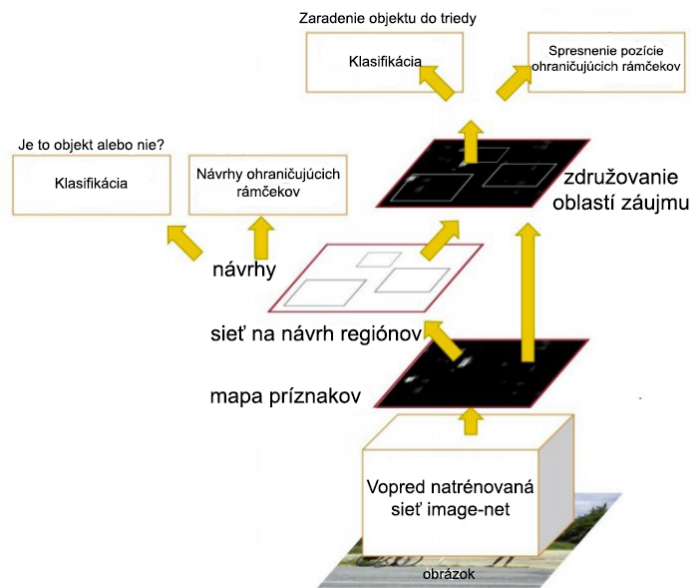
Obr. 3.7: Diagram štruktúry Fast R-CNN, upravené z [9].

3.4.3 Faster R-CNN

Ani po vylepšeniach, ktoré priniesla Fast R-CNN však stále nebolo možné rozpoznávanie objektov v reálnom čase. Zistilo sa, že hlavnou prekážkou je spôsob generovania návrhov regiónov. Na prekonanie tejto prekážky bol nahradený algoritmus selektívneho vyhľadávania neurónovou sieťou na návrh regiónov Region Proposal Network - RPN, čo viedlo ku vzniku Faster R-CNN [45].

RPN je plne konvolučná neurónová sieť, ktorá predpovedá návrhy príznakov. Táto sieť používa techniku posuvného okna, ktoré sa pohybuje po každej mape príznakov a vypočítava počet potenciálnych ohraničujúcich rámcov na každej pozícii. Každému ohraničujúcemu rámcu sa priradí skóre dôveryhodnosti, ktoré udáva pravdepodobnosť, že objekt je v rámcu správne umiestnený. Diagram fungovania siete Faster R-CNN je možné vidieť na obrázku 3.8.

Po procese generovania ohraničujúcich rámcov sa návrhy vložia do Fast R-CNN na klasifikáciu a spresnenie ohraničujúcich boxov. Kombinácia RPN a Fast R-CNN umožňuje Faster R-CNN dosiahnuť ešte rýchlejšiu a presnejšiu detekciu objektov v porovnaní s predchádzajúcimi metódami.



Obr. 3.8: Diagram štruktúry Faster R-CNN, upravené z [9].

3.5 Jednokrokové detektory

Novú generáciu detektorov založených na strojovom učení predstavujú jednokrokové detektory. Jednokrokové detektory sú založené na regresnej analýze a majú schopnosť priamo mapovať pixely obrazov na súradnice ohraničujúcich rámcov a poskytnúť pravdepodobnosti zaradenia objektov do jednotlivých tried pri jednom priechole obrazu sieťou.

Na rozdiel od detektorov založených na regiónoch zložených z viacerých častí, ktoré vyžadujú samostatné trénovanie je možné trénovať jednokrokové detektory ako celok, čím sa zjednodušuje proces trénovania a vývoja.

Výhodou jednokrokových detektorov je aj vysoká rýchlosť spracovania obrazu, keďže na detekovanie a lokalizáciu objektov stačí jeden prechod sieťou. V porovnaní s metódami založenými na regiónoch sú jednokrokové detektory výpočtovo nenáročnejšie a ich presnosť je porovnateľná až na detekovanie malých objektov, kde detektory založené na regiónoch dosahujú vyššiu presnosť. To činí jednokrokové detektory vhodnými pre aplikácie v oblastiach, ako je rozpoznávanie obrazov v reálnom čase.

3.5.1 YOLO - You Only Look Once

Model YOLO (You Only Look Once) [43] je založený na jednej konvolučnej sieti a umožňuje predpovedať ohraničujúce rámčeky a tzv. skóre pravdepodobnosti triedy, ktorá sa v nich nachádza jedným prechodom sieťou. Vstupný obrázok sa najskôr rozdelí na mriežku o veľkosti $S \times S$. Ak stred objektu spadá do bunky mriežky, táto bunka je zodpovedná za jeho detekciu. Každá bunka mriežky predpovedá niekoľko ohraničujúcich rámcov a skóre dôveryhodnosti pre každý tento rámček, ktoré reprezentuje mieru istoty, že obdĺžnik obsahuje objekt. Bunka tiež predpovedá pravdepodobnosti príslušnosti objektu k triedam, za predpokladu, že bunka obsahuje objekt. Séria modelov YOLO je špeciálne navrhnutá pre scenáre, kde je potrebné spracovanie obrazu v reálnom čase s úmyslom zvýšenia rýchlosti inferencie pri zachovaní presnosti.

YOLO prináša vďaka spracovaniu celých obrázkov naraz niekoľko výhod. Nepotrebuje komplexné zrefazované spracovanie, čo zaisťuje vysokú rýchlosť detekcie. Na rozdiel od posuvného okna a metód založených na návrhoch regiónov vidí YOLO pri tréovaní a testovaní celý obraz, čím implicitne kóduje kontextové informácie o triedach a ich vzhľade. Oproti Fast R-CNN má YOLO o polovicu menej chýb pri detekcii pozadia.

Avšak YOLO má problémy s presným určením polohy niektorých objektov, najmä malých. Model YOLO sa radí medzi najrýchlejšie metódy - napríklad model YOLOv5 Nano beží rýchlosťou až 230 FPS na GPU [52].

Vývoj modelov YOLO

- **YOLOv1** [43] využíval GoogLeNet[55] na extrakciu máp príznakov zo vstupných obrázkov s veľkosťou $448 \times 448 \times 3$. Príznakové mapy potom prešli cez dve plne prepojené vrstvy na konečný výstup, pričom sa na výrazné zníženie výpočetného výkonu a zvýšenie rýchlosti inferencie použila priama regresia.
- **YOLOv2** [44] použil ako základnú sieť (angl. backbone) DarkNet-19 s pridaním priedochodnej vrstvy na kombináciu sémantických informácií z vysokých a nízkych úrovní. Tým sa zlepšila schopnosť modelu detekovať malé objekty.
- **YOLOv3** [12] zaviedol modifikovanú chrbtovú sieť DarkNet53 a implementoval tzv. siete pyramidových príznakov (Feature Pyramid Networks - FPN) [37] na detekciu objektov pomocou viacrozmerých máp príznakov.
- **YOLOv4** [3] skombinoval techniku Cross Stage Partial (CSP)[58] a základnú sieť DarkNet-53 a vytvoril základnú sieť CSPDarkNet-53, ktorá poskytla bohatšie informácie o zlúčení gradientov. Model tiež použil architektúru Spatial Pyramid Pooling (SPP) [17] na rozšírenie receptívneho poľa a do „šije“ modelu (angl. neck) pridal na zníženie výpočetnej náročnosti modul Path Aggregation Network (PAN) [38].
- V **YOLOv5** [31] nahradila architektúru SPP vylepšená technika Spatial Pyramid Pooling - Fast (SPPF) [17] a na zlepšenie výkonu bol pridaný modul Focus. Model YOLOv5 tiež využíval namiesto open-source frameworku Darknet framework PyTorch.
- **YOLOv6** [36] využíval EfficientRep [59] ako chrbticovú sieť a do šije modelu začlenil Rep-PAN, založený na metódach PAN [38] a Rep [8]. Inšpiráciou bola architektúra YOLOX [13], konkrétne použitie oddelenej „hlavy“ (angl. head) z architektúry YOLOX v architektúre YOLOv6 .
- **YOLOv7** [57] zaviedol plánovaný preparametrizovaný konvolučný modul a koncept škálovania a rozširovania modelu pre zvýšenie efektivity parametrov.

YOLOv8

V januári 2023 sa spoločnosť Ultralytics, ktorá stála za vývojom úspešného YOLOv5 rozhodla vydať novú verziu svojho modelu pre detekciu objektov v obraze YOLOv8 [32]. YOLOv8 sa drží súčasného trendu a je bez kotiev (angl. anchors), čo znižuje počet potrebných predpovedí rámečkov a tým zrýchľuje proces post-processingu, konkrétne zrýchľuje algoritmus NMS (Non-Maximum Suppression). Okrem toho YOLOv8 využíva počas tréovania tzv. mozaikové rozšírenie, ktoré však môže byť niekedy škodlivé, ak sa používa počas celého procesu tréovania a preto je v posledných desiatich epochách vypnuté. Model YOLOv8 je

možné spustiť pomocou rozhrania príkazového riadka (CLI) alebo môže byť nainštalovaný ako balík PIP. Navyše, model YOLOv8 je dodávaný s viacerými integráciami na označovanie, tréning a nasadenie, čo uľahčuje jeho použitie v praxi. V nasledujúcej časti je opísaná architektúra tohto modelu, ktorá je zobrazená na obrázku 3.9.

Základná sieť (Backbone)

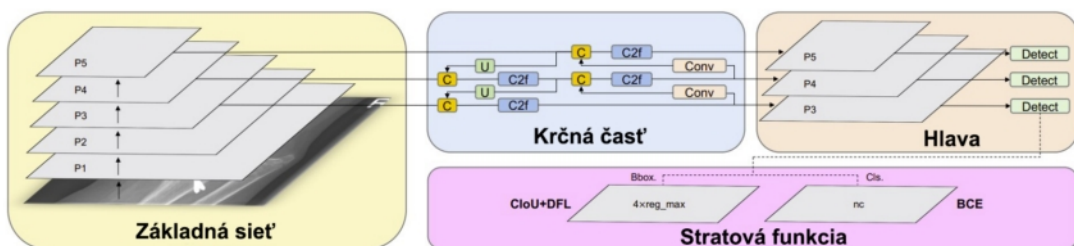
Model YOLOv8 používa techniku CSP [58] na rozdelenie mapy prvkov na dve časti, pričom jedna časť využíva konvolučné operácie a druhá je spojená s výstupom konvolučných operácií predchádzajúcej časti. Tým sa zlepšuje schopnosť modelu učiť sa a znižuje sa jeho výpočtová náročnosť. Namiesto modulu C3 použitého v YOLOv5 používa YOLOv8 modul C2f, čo umožňuje modelu získať bohatšie informácie o gradientnom toku. YOLOv8 tiež znižuje počet blokov v každej časti a na zvýšenie rýchlosti inferencie modelu využíva modul SPPF.[17]

Krčná časť (Neck)

Vo všeobecnosti hlbšie neurónové siete poskytujú viac informácií a vedú k lepším výsledkom. V prípade malých objektov však môže príliš veľa konvolučných operácií viesť k strate informácií. Na vyriešenie tohto problému je potrebná tzv. viacškálová fúzia vlastností [23] pomocou architektúr FPN [37] a PAN [38]. Krčná časť architektúry využíva viacškálové zlúčenie vlastností obrazov, pričom horné prvky majú viac informácií vďaka väčšiemu počtu vrstiev siete, zatiaľ čo dolné prvky si zachovávajú viac informácií o polohe vďaka menšiemu počtu konvolučných vrstiev.

Hlava (Head)

Na rozdiel od jeho predchodcov, ktorí používajú spojenú hlavu, YOLOv8 namiesto toho používa tzv. oddelenú hlavu (angl. decoupled head), ktorá oddeľuje klasifikačnú a detekčnú hlavu. YOLOv8 nahrádza prístup založený na kotvách tzv. Anchor-Free prístupom, ktorý tieto kotvy nepotrebuje a objekt lokalizuje podľa jeho stredu - predpovedá vzdialenosť od stredu k ohraničujúcemu rámčeku.



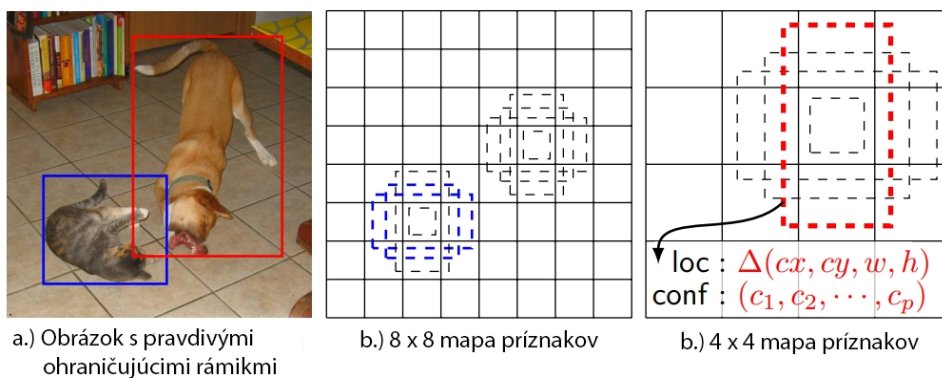
Obr. 3.9: Architektúra modelu YOLOv8, upravené z [33].

3.5.2 SSD: Single Shot MultiBox Detector

Model SSD [39] je založený na doprednej konvolučnej sieti, ktorá vytvára kolekciu ohraničujúcich rámčekov s pevnou veľkosťou a skóre dôveryhodnosti pre prítomnosť triedy objektov v týchto boxoch. Po tomto kroku nasleduje algoritmus potlačenia nemaximálnych hodnôt

(NMS), ktorý vyfiltruje prekrývajúce sa rámčeky, ktoré zdieľajú viac ako 50% oblasti vybraním rámčeka s najväčším skóre dôveryhodnosti pre danú triedu. Prvé vrstvy siete sú založené na štandardnej architektúre konvolučnej siete používanej pre klasifikáciu obrazu, ktorá sa nazýva základná sieť. Ide konkrétne o sieť VGG-16 ale autori tvrdia, že použitie podobných sietí by tiež malo produkovať dobré výsledky [39]. Na túto základnú sieť naväzujú postupne sa zmeňujúce konvolučné vrstvy, ktoré umožňujú detekciu objektov v rôznych mierkach. Na rozdiel od modelov YOLO, ktoré detekujú objekty iba na jednej vrstve príznakov, detekuje SSD na viacerých vrstvách, čo môže mať za následok vyšší počet celkových detekcií.

Vrstvy príznakov sú ďalej rozdelené do mriežky (obrázok 3.10). Každéj bunke v tejto mriežke je priradená sada pôvodných ohraničení, od ktorých sa následne spolu so skóre dôveryhodnosti predpovedá aj relatívny posun.



Obr. 3.10: Detekcia SSD. Na obrázku *a* je vstupný obrázok a pravdivé ohraničujúce rámy. Na obrázkoch *b* a *c* je mapa príznakov rozdelená na bunky podľa mierky. Pre každú bunku je odhadnuté skóre dôveryhodnosti pre danú triedu a posun ohraničenej oblasti, upravené z [39].

Kapitola 4

Návrh riešenia

V tejto kapitole sú najskôr zvolené modely pre detekciu povrchových poškodení a následne je opísané tréningové prostredie použité na tréning týchto modelov. Ďalej sú popísané dátové sady použité na tréning a testovanie modelov, ako aj metriky použité na zrovnávanie výsledkov. Táto kapitola umožňuje bližšie sa zoznámiť s nástrojmi a prostriedkami, ktoré boli v tomto výskume použité a ktoré umožnili dosiahnuť výsledky opísané v nasledujúcej kapitole.

4.1 Zvolené modely

Pre praktické vypracovanie tejto práce bol ako zástupca detektorov založených na regiónoch model Faster R-CNN so základnou sieťou ResNetv2 a to kvôli jeho vysokej presnosti a zároveň zníženej výpočtovej náročnosti tréningu v porovnaní s jeho predchodcami R-CNN a Fast-RCNN. Tento model a jeho postupný vývoj je bližšie opísaný v časti 3.4.3. Ako zástupca jedнокrokových detektorov bol zvolený model YOLOv8, kvôli jeho dosahovanej presnosti, jednoduchosti tréningu a jednoduchosti použitia. Tento model je viac opísaný v časti 3.5.1. Obidva zvolené modely sú implementované v programovacom jazyku Python.

4.2 Tréningové prostredie

V tejto práci som ako prostredie využíval dve služby. Pre výpočtovo menej náročné tréningovanie som kvôli bezplatnosti využíval služby prostredia Google Colab. Pre výpočtovo náročnejšie tréningovanie som používal cloudové prostredie Google Cloud Platform.

4.2.1 Google Colab

Google Colab je bezplatná online služba založená na Jupyter notebookoch, ktorá poskytuje výpočetné prostriedky, úložisko a integráciu s Google Drive pre tréningovanie modelov neurónových sietí. Táto platforma umožňuje efektívne tréningovanie a experimentovanie pre výskumníkov, študentov a odborníkov v oblasti strojového učenia. Vďaka poskytovaným výpočetným zdrojom ako sú grafické procesory (GPU) a tensorové procesory (TPU) sa znižuje čas tréningu a umožňuje rýchlejšie experimentovanie. Google Colab tiež podporuje širokú škálu knižníc strojového učenia ako sú TensorFlow, PyTorch a Keras. Služba poskytuje aj plán, ktorý je zdarma a umožňuje akceleráciu výpočtu pomocou grafických kariet Tesla T4 s pamäťou 16GB. Nevýhodou tohto prostredia je, že každý užívateľ môže mať

spustený maximálne jeden virtuálny stroj, musí byť aktívny, maximálna doba spustenia každého virtuálneho stroja je obmedzená a dáta na disku nie sú perzistentné.

4.2.2 Google Cloud Platform (GCP)

GCP je cloudové prostredie, ktoré okrem iného umožňuje aj tréning a nasadenie modelov strojového učenia. Poskytuje rôzne výpočtové zdroje, vrátane CPU, GPU a TPU, ktoré umožňujú efektívne tréning a nasadenie modelov. Hlavnou službou, ktorú som v tomto prostredí využíval bola Vertex AI, ktorá zjednodušuje prácu s modelmi strojového učenia a poskytuje možnosti pre správu modelov. Najviac som využíval jej súčasť Workbench, ktorá je podobne ako Google Colab založená na Jupyter notebookoch. Taktiež som využíval službu Cloud Storage, ktorá umožňuje rýchle a bezpečné ukladanie veľkých dátových súborov. Všetky tréningy v tejto službe prebiehali na virtuálnom stroji, ktorý poskytoval 4 CPU procesory a 26 GB pamäte RAM. Pre akcelerovanie výpočtov bola k tomuto virtuálnemu stroju pridaná aj grafická karta NVIDIA V100 s pamäťou 16GB.

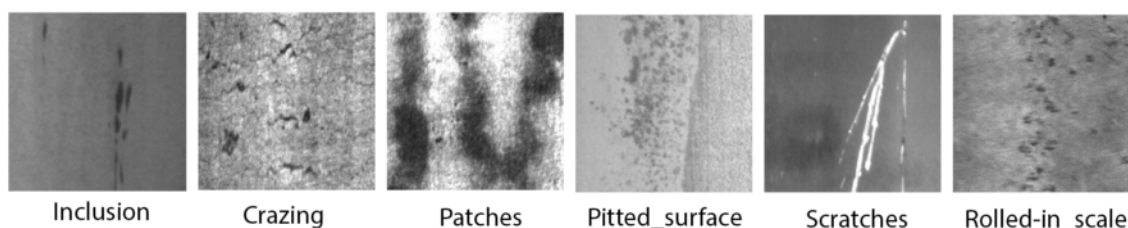
4.3 Dátové sady

Pre potreby tejto práce boli zvolené dve dátové sady poškodení. Prvá z nich sníma rôzne druhy poškodení na povrchu ocele, zatiaľ čo druhá zachytáva poškodenia na drevenom povrchu. Tieto dva typy materiálov boli zvolené z dôvodu ich širokého využitia v priemysle a potreby monitorovania kvality povrchu. Zvolené dátové sady poskytujú vhodný základ pre analýzu a hodnotenie modelov pre detekciu poškodení. V nasledujúcej časti sú tieto dátové sady bližšie predstavené.

4.3.1 NEU-DET

NEU-DET [51] je dátová sada povrchových chýb od Northeastern University (NEU), ktorá obsahuje šesť tried - šesť typov poškodení. Tieto povrchové chyby sú zozbierané na povrchu ocelového plechu valcovaného za tepla. Súbor údajov obsahuje 1800 obrázkov v odtieňoch šedej, t. j. 300 vzoriek v každej triede povrchových poškodení. Originálne rozlíšenie každého obrázku je 200×200 . Triedy poškodení sú zobrazené na obrázku 4.1 a v dátovej sade sú pomenované nasledovne:

- Inclusion (In): Inklúzia je typickým poškodením kovových povrchov a reprezentuje rôzne nekovové materiály, ktoré vniknú do ocele počas jej výroby na oceliarniach. Niektoré inklúzie sú voľné a ľahko odpadávajú, niektoré sú vtlačené do plechu.
- Cracking (Cr): Malé praskliny, ktoré vznikajú na povrchu materiálu.
- Patches (P): Časti kovu, ktoré sa od zvyšku odlišujú určitou vlastnosťou.
- Pitted surface (Ps): Dierkovaný povrch je forma korózie, ktorá sa vyskytuje na kovových povrchoch a preniká do vnútra kovov. Takéto poškodenie má spravidla malý priemer, ale zasahuje viac do hĺbky.
- Scratches (Sc): Škrabance sú stopy oderu na povrchu.
- Rolled-in scale (Rs): Vada vzniká, keď sa tzv. okuje, čo sú okysličené plátky kovu (oxidy železa) vznikajúce pri tepelnom tvarovaní kovu počas procesu valcovania zavalcujú do kovu.

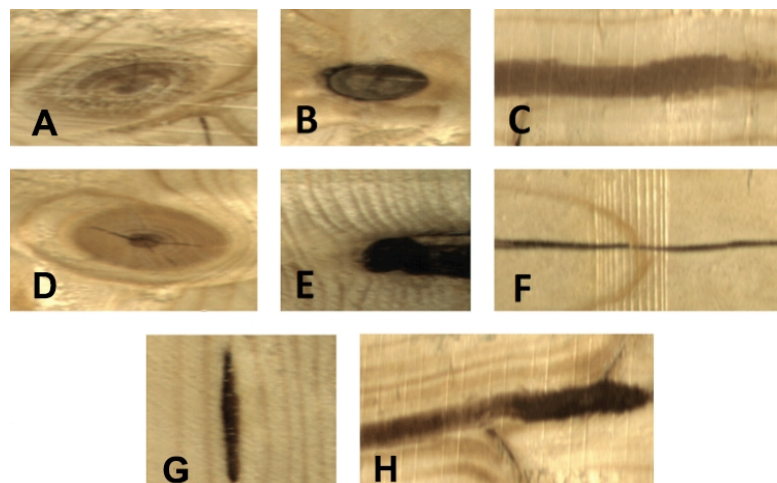


Obr. 4.1: Triedy povrchového poškodenia, ktoré obsahuje dátová sada NEU-DET.

4.3.2 Large Scale Image Dataset of Wood Surface Defects

Táto dátová sada [2] obsahuje zmenšené obrázky originálnej dátovej sady „*A large-scale image dataset of wood surface defects for automated vision-based quality control processes*“ [35]. Pôvodná dátová sada obsahovala fotografie s vysokým rozlíšením a veľkosťou až 12MB zatiaľ čo autor tejto dátovej sady zredukoval veľkosť jednej fotografie na približne 1MB a rozlíšenie 2800 x 1024. Obsah dátovej sady tvorí rozsiahly súbor 4000 autentických fotografií 8 typov najčastejších povrchových poškodení dreva z výrobnjej linky. Nasleduje prehľad obsiahnutých tried poškodení [5], [62]:

- Quartzity: kremenitosť je lokálna zmena štruktúry dreva v časti kmeňa vystavenej tlaku, ktorá sa prejavuje výrazným zväčšením zóny letokruhov. Táto vada sa vyskytuje len u ihličnatých drevín.
- Live Knot: živý uzol (živá hrča) je časť vetvy obrastená drevom. Ide o prirodzený jav spôsobený zarastaním konárov, ktoré sú základným funkčným orgánom každého stromu, do kmeňa. Živý uzol má zvyčajne svetlejšiu farbu a spravidla nikdy nevypadne.
- Dead Knot: mŕtvy uzol je uzol, ktorý stratil vláknité spojenie s okolitým drevom. Takýto uzol je tmavší a môže sa ľahko uvoľniť a vypadnúť, čo môže mať vplyv na pevnosť dreva.
- Knot with crack: uzol, ktorý obsahuje prasklinu alebo rozštiepenie.
- Knot missing: chýbajúci uzol je miesto v dreve, kde mal byť uzol, ale z nejakého dôvodu tam nie je. V niektorých prípadoch to môže byť žiaduce, pretože tak môže v dreve vzniknúť čistá oblasť bez chýb.
- Marrow: dreň, ktorá je vo veľkej vzdialenosti od geometrického stredu prierezu kmeňa.
- Resin: časť dreva, ktorá je impregnovaná živicom. Takáto oblasť vzniká mechanickým poškodením rastúceho stromu, napr. ťažbou živice, poranením zvieratami alebo niektorými hubami.
- Crack: bočná prasklina kmeňa spôsobená výrubom alebo prepravou dreva.

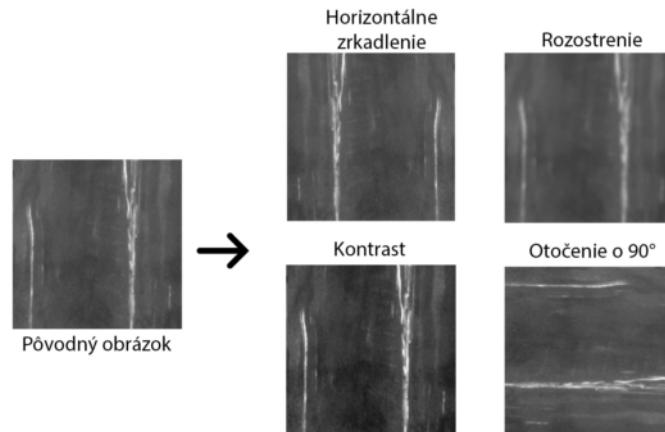


Obr. 4.2: Triedy povrchového poškodenia, ktoré obsahuje dátová sada Large Scale Image Dataset of Wood Surface Defects. (A) Live Knot, (B) Dead Knot, (C) Quartzity, (D) Knot with crack, (E) Knot missing, (F) Crack, (G) Resin, (H) Marrow. Upravené z [35].

4.3.3 Augmentácia dátovej sady

Metódy augmentácie dát sú silným nástrojom, ktorý môže pomôcť zabrániť pretrénovaniu neurónovej siete. Pretrénovanie je pojem, ktorý znamená, že daný klasifikátor podáva dobré výsledky na tréningových dátach, ale výrazne horšie výsledky na validačných (predtým nevidených) dátach. Princípom augmentácie dát je umelé zväčšovanie pôvodnej tréningovej dátovej sady. Ide napríklad o techniky ako rotácia pôvodného obrazu (zrotovanie obrázku o niekoľko stupňov), zrkadlenie (vertikálne, horizontálne), priblíženie, úpravu farieb, kontrastu, jasú, pridanie šumu alebo mozaikové kombinovanie tréningových obrazov. Existujú dva prístupy k augmentácii:

- **Offline augmentácia:** Predbežná augmentácia sa používa pred tréňovaním modelu a slúži na zväčšenie dátovej sady. Obrázky, ktoré prejdú touto augmentáciou, sa následne pridajú do dátovej sady. Pri väčšej dátovej sade však môže byť potrebné zohľadniť aj potrebné zvýšené miesto na disku, ktoré takáto dátová sada zaberá.
- **Online augmentácia:** Augmentácia v reálnom čase sa na rozdiel od offline augmentácie aplikuje priamo počas tréňovania modelu. Cieľom tejto augmentácie je vytvoriť väčšiu rozmanitosť medzi obrázkami, čo vedie k lepšej príprave neurónovej siete. Táto augmentácia sa uplatňuje na jednotlivé obrázky alebo dávku obrázkov s využitím náhodných hodnôt. To znamená, že model uvidí pri každej epoche rôzne obrázky, čím sa dátová sada umelo zväčší niekoľkonásobne bez potreby ukladania augmentovaných dát, čím sa šetrí miesto na disku.



Obr. 4.3: Techniky augmentácie dátovej sady.

4.4 Metriky pre zrovnávanie

Intersection over Union (priesečník nad zjednotením, ďalej len IoU) meria prekryvanie predpovedaného ohraničujúceho rámčeka alebo masky s pravdivým ohraničujúcim rámčekom alebo masku pre daný objekt na obrázku. IoU sa vypočíta vydelením prieniku predpovedaných a pravdivých oblastí zjednotením týchto oblastí. Výsledná hodnota sa pohybuje od 0 do 1, pričom hodnota 1 znamená dokonalú zhodu medzi predpovedanými a pravdivými oblasťami.

$$\text{IoU} = \frac{\text{Oblasť prieniku}}{\text{Oblasť zjednotenia}}$$

Obr. 4.4: Definícia metriky IoU

Pre určitý dataset môže byť vopred definovaný prah IoU (napríklad 0,5), podľa ktorého je možné rozdeliť predikcie do nasledujúcich tried [24].

- **skutočne pozitívna** (TP) - správne detekovaný objekt ($\text{IoU} > \text{prah}$)
- **falošne pozitívna** (FP) - nesprávne detekovaný objekt ($\text{IoU} < \text{prah}$)
- **skutočne negatívna** (TN) - správne nedetekovaný objekt
- **falošne negatívna** (FN) - nesprávne nedetekovaný objekt

Na základe týchto tried je možné určiť ďalšie dôležité metriky (medzi odborníkmi v obore je zvykom využívať anglické formy) - precision (presnosť), recall (citlivosť) a average precision (priemerná presnosť).

Precision udáva presnosť predpovedí, t. j. percento pravdivých predpovedí.

$$Precision = \frac{TP}{TP + FP} \quad (4.1)$$

Recall meria koľko zo všetkých skutočne pozitívnych prípadov bolo predpovedané správne.

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

Average precision (AP), je metrika, ktorá sa používa na meranie presnosti objektového detektoru. Táto metrika, ktorá sa určuje zvlášť pre každú triedu detekovaných objektov udáva priemer maximálnych hodnôt precision pre jednotlivé hodnoty recall, tým pádom nadobúda hodnoty z intervalu $\langle 0,1 \rangle$, kde 1 označuje najlepší prípad. Princíp AP je rovnaký ako počítanie plochy pod precision-recall krivkou [24].

Mean average precision (mAP) sa vypočíta určením AP pre každú triedu detekovaných objektov a spočítaním ich priemeru. Najčastejšie sa používa mAP s prahom IoU 0,5 (mAP(0,50)) a mAP s prahom IoU 0,95 (mAP(0,95)).

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (4.3)$$

Ohraničujúce rámčeky

Pre každý ohraničujúci rámček (angl. bounding box) existuje päť predpovedí: x , y , w , h , c , kde (x, y) sú súradnice stredu detekovaného objektu, (w, h) sú šírka a výška vzhľadom na veľkosť obrázka a c je skóre dôveryhodnosti, ktoré predstavuje *Intersection over union* (IOU) medzi základným pravdivým a predpovedaným ohraničujúcim rámčekom. Skóre dôveryhodnosti je formálne definované ako:

$$c = P(Object) * IOU(0, x) \quad (4.4)$$

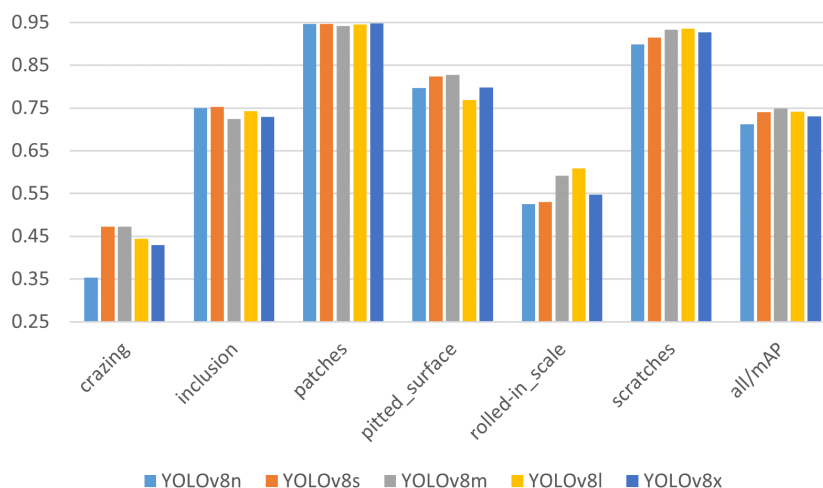
Kapitola 5

Experimenty

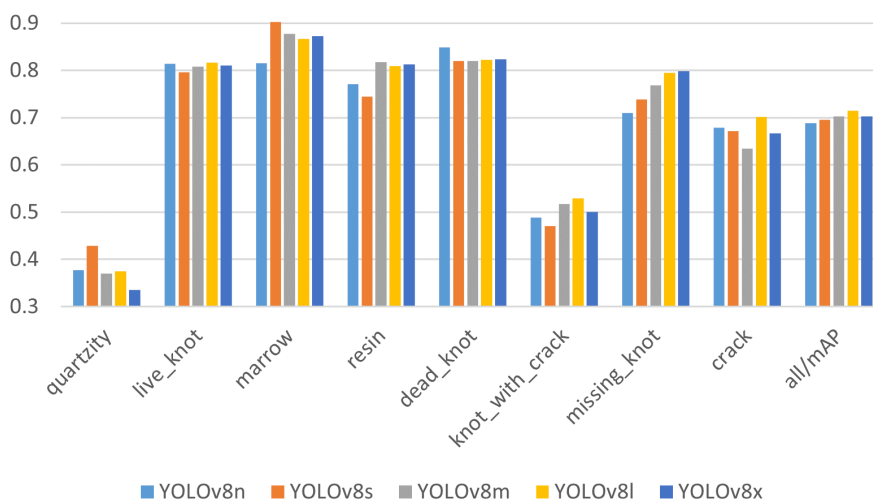
Táto kapitola sa zaoberá sériou experimentov zameraných na porovnávanie presnosti dvoch vybraných modelov detekcie objektov, Faster RCNN a YOLOv8. Experimenty prebiehali na rôzne upravených dátových sadách, aby sme získali hlbšie pochopenie o ich výkonnosti a presnosti. Prvý experiment sa zaoberal porovnaním presnosti týchto modelov. Následne bola skúmaná ich presnosť na upravených dátových sadách so zmeneným rozlíšením. Nakoniec boli porovnané presnosti modelov pri použití rôznych druhov augmentácií dátových sád.

5.1 Porovnanie presnosti modelov na originálnych dátových sadách

Cieľom tohto experimentu bolo zistiť, ktorý z modelov poskytuje lepšiu presnosť detekcie poškodení na vybraných testovacích dátových sadách. V prvom kroku experimentu som natreinoval a porovnal rôzne veľkosti YOLOv8 modelu obidvoch dátových sadách. Na základe výsledkov tohto porovnania som pre ďalšie experimenty vybral model YOLOv8s, ktorý sa ukázal ako najlepší kompromis medzi výkonnosťou a efektívnosťou, nakoľko tréning väčších modelov bolo náročnejšie na výpočetný výkon.



Obr. 5.1: Porovnanie metriky AP(0,50) YOLOv8 modelov na dátovej sade *NEU-DET*.



Obr. 5.2: Porovnanie metriky AP(0,50) YOLOv8 modelov na dátovej sade *Large Scale Image Dataset of Wood Surface Defects*.

V ďalšej fáze experimentu som natrénoval modely Faster R-CNN na tých istých dátových sadách. Tieto modely boli použité ako alternatívne riešenie pre porovnanie s vybraným YOLO modelom s cieľom zistiť ich silné a slabé stránky. Na základe výsledkov tohto porovnania bolo možné určiť, ktorý model je pre konkrétnu dátovú sadu vhodnejší. Tieto výsledky budú tiež slúžiť ako základ pre ďalšie experimenty.

Z analýzy tabuľky 5.1 možno vyvodit, že model YOLOv8s dosiahol výrazne lepšie výsledky v kategórii *cracking*, zatiaľ čo Faster R-CNN mal mierne lepšie hodnoty v kategórii *rolled-in_scale*. V ostatných kategóriách boli rozdiely medzi modelmi menej výrazné. V celkovej metrike mAP dosahoval model YOLOv8s lepšie výsledky v oboch prípadoch.

	Faster R-CNN		YOLOv8s	
	<i>AP(0,50)</i>	<i>AP(0,95)</i>	<i>AP(0,50)</i>	<i>AP(0,95)</i>
<i>cracking</i>	0,317	0,092	0,472	0,193
<i>inclusion</i>	0,777	0,395	0,752	0,405
<i>patches</i>	0,918	0,558	0,946	0,586
<i>pitted_surface</i>	0,786	0,475	0,824	0,553
<i>rolled-in_scale</i>	0,594	0,258	0,53	0,224
<i>scratches</i>	0,931	0,585	0,915	0,576
<i>all/mAP</i>	0,72	0,394	0,74	0,423

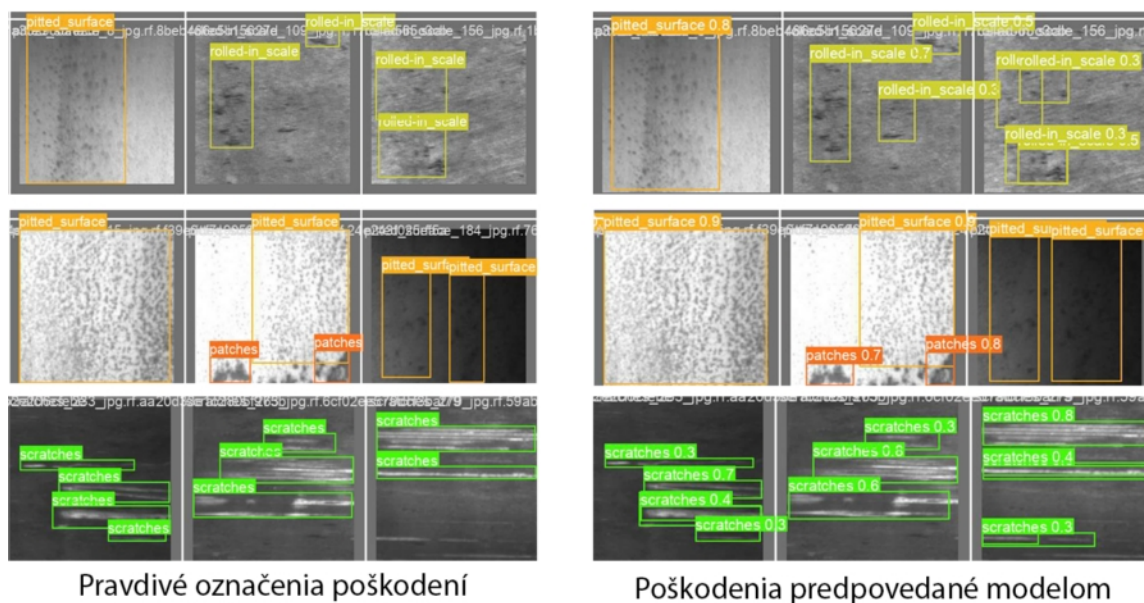
Tabuľka 5.1: Porovnanie presnosti modelu Faster R-CNN a YOLOv8s na dátovej sade *NEU-DET*

Z tabuľky 5.2 vyplýva, že model YOLOv8s je výrazne lepší v detekcii tried *quartzity* a *marrow*. V ostatných kategóriách boli rozdiely medzi modelmi menej výrazné. Celková hodnota mAP dosiahnutá modelom YOLOv8s (0,696) bola vyššia v porovnaní s hodnotou dosiahnutou modelom Faster R-CNN (0,618), z čoho vyplýva, že YOLOv8s má lepšiu celkovú výkonnosť aj na tejto dátovej sade.

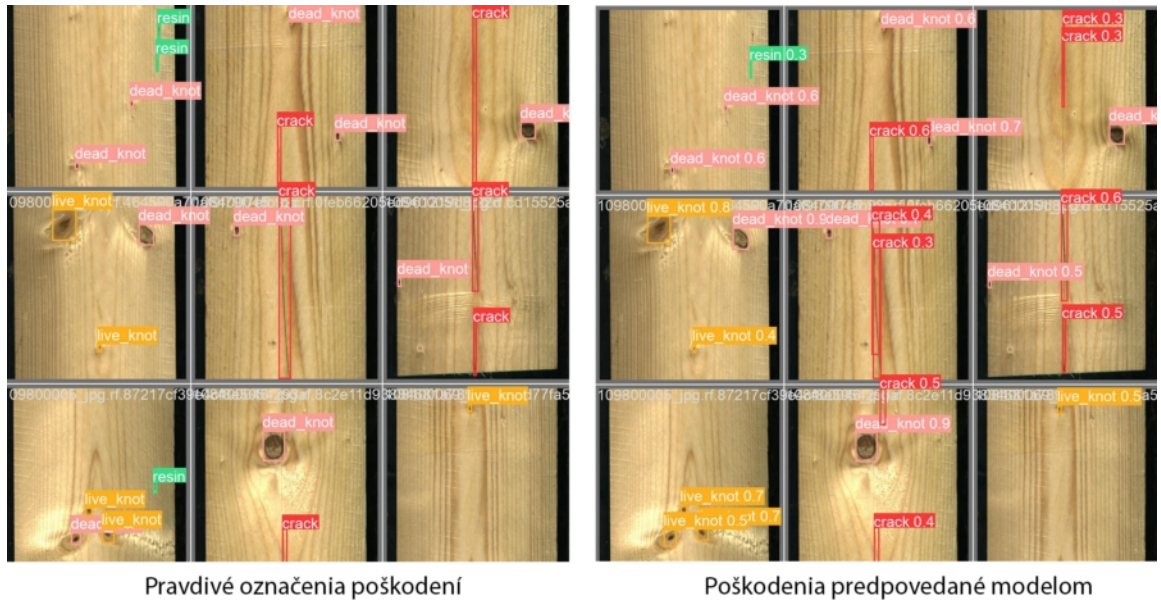
	Faster R-CNN		YOLOv8s	
	$AP(0,50)$	$AP(0,95)$	$AP(0,50)$	$AP(0,95)$
<i>quartzity</i>	0,141	0,086	0,429	0,305
<i>live_knot</i>	0,759	0,31	0,796	0,379
<i>marrow</i>	0,729	0,508	0,902	0,611
<i>resin</i>	0,752	0,324	0,744	0,356
<i>dead_knot</i>	0,779	0,362	0,82	0,446
<i>knot_with_crack</i>	0,467	0,29	0,471	0,312
<i>missing_knot</i>	0,704	0,373	0,738	0,338
<i>crack</i>	0,616	0,321	0,672	0,362
<i>all/mAP</i>	0,618	0,322	0,696	0,389

Tabuľka 5.2: Porovnanie presnosti modelu Faster R-CNN a YOLOv8s na dátovej sade *Large Scale Image Dataset of Wood Surface Defects*.

Na obrázkoch 5.3 a 5.4 je znázornené porovnanie vstupných a výstupných obrázkov modelu YOLOv8 na obidvoch dátových sadách.



Obr. 5.3: Porovnanie pravdivých označení a označení predpovedaných YOLOv8 modelom na dátovej sade *NEU-DET*.



Obr. 5.4: Porovnanie pravdivých označení a označení predpovedaných YOLOv8 modelom na dátovej sade *Large Scale Image Dataset of Wood Surface Defects*.

5.2 Vplyv zmeny rozlíšenia dátových sád na presnosť modelov

Cieľom tohto experimentu bolo zistiť ako zmena rozlíšenia dátových sád ovplyvňuje výkon modelov Faster R-CNN a YOLOv8 a ako sa tieto zmeny prejavujú v porovnaní s predchádzajúcimi výsledkami.

5.2.1 Zväčšenie rozlíšenia sady NEU-DET

Dátovej sade NEU-DET som s cieľom zlepšiť presnosť detekcie zvýšil rozlíšenie z pôvodných 200×200 na 416×416 pixelov. Následne som na tejto upravenej dátovej sade natrénoval modely Faster R-CNN a YOLOv8. Týmto krokom som sa snažil zistiť, či takáto zmena môže pomôcť dosiahnuť lepšiu presnosť detekcie poškodení. Presnosť týchto natrénovaných modelov som následne porovnal.

	Faster R-CNN		YOLOv8s	
	$AP(0,50)$	$AP(0,95)$	$AP(0,50)$	$AP(0,95)$
<i>crazing</i>	0,422	0,137	0,485	0,206
<i>inclusion</i>	0,763	0,375	0,782	0,419
<i>patches</i>	0,923	0,57	0,943	0,598
<i>pitted_surface</i>	0,748	0,476	0,798	0,55
<i>rolled-in_scale</i>	0,674	0,25	0,573	0,281
<i>scratches</i>	0,93	0,573	0,916	0,568
<i>all/mAP</i>	0,743	0,397	0,749	0,437

Tabuľka 5.3: Porovnanie presnosti modelu Faster R-CNN a YOLOv8s na dátovej sade *NEU-DET* zväčšenej na rozlíšenie 416×416 .

Z porovnania výsledkov pôvodnej dátovej sady (tabuľka 5.1) a dátovej sady so zvýšeným rozlíšením (tabuľka 5.3) možno pozorovať určité zlepšenia v presnosti pri triedach *crazing* a *rolled-in_scale*, kde sa zaznamenala zvýšená presnosť AP(0,50) pre obidva modely. Celková presnosť sa pre oba modely zvýšila, avšak nie výrazne.

5.2.2 Zníženie rozlíšenia sady Large Scale Image Dataset of Wood Surface Defects

Experimentovanie s druhou dátovou sadou *Large Scale Image Dataset of Wood Surface Defects* malo za cieľ znížiť výpočetnú náročnosť tréningu modelov na tejto dátovej sade znížením rozlíšenia dátovej sady z pôvodných 2800×1024 na 640×640 . Na takto upravenej dátovej sade som natrénoval obidva modely a následne som porovnal výsledky presnosti týchto modelov s výsledkami z predchádzajúceho experimentu (tabuľka 5.2).

	Faster R-CNN		YOLOv8s	
	AP(0,50)	AP(0,95)	AP(0,50)	AP(0,95)
<i>quartzity</i>	0,129	0,083	0,252	0,108
<i>live_knot</i>	0,794	0,342	0,807	0,407
<i>marrow</i>	0,723	0,444	0,713	0,418
<i>resin</i>	0,73	0,333	0,707	0,363
<i>dead_knot</i>	0,826	0,401	0,849	0,464
<i>knot_with_crack</i>	0,589	0,369	0,663	0,463
<i>missing_knot</i>	0,643	0,325	0,796	0,444
<i>crack</i>	0,59	0,25	0,61	0,298
<i>all/mAP</i>	0,628	0,318	0,675	0,371

Tabuľka 5.4: Presnosť modelov Faster R-CNN a YOLOv8s na dátovej sade *Large Scale Image Dataset of Wood Surface Defects* pri rozlíšení 640×640 .

V porovnaní s pôvodným rozlíšením (2800×1024) dosiahli modely zmiešané výsledky. Zatiaľ čo u modelu Faster R-CNN sa dosiahlo zlepšenie hodnoty mAP(0,50) z pôvodných 0,618 na 0,628, model YOLOv8s zaznamenal zhoršenie tejto hodnoty na 0,675 (z pôvodných 0,696). Z týchto výsledkov vyplýva, že zníženie rozlíšenia dátových sád negatívne ovplyvňuje presnosť modelov, avšak pri menšom rozlíšení (640×640) sa presnosť modelu YOLOv8s zhoršila iba mierne. Zníženie rozlíšenia však dokázalo výrazne znížiť výpočetné nároky na tréning a tým tento proces urýchliť.

5.3 Vplyv augmentácií dátových sád na presnosť modelov

V rámci tretieho experimentu som skúmal vplyv rôznych augmentácií dátových sád na presnosť modelov. Vychádzal som pritom z dátových sád z minulého experimentu, keďže sa tým pri dátovej sade NEU-DET mierne zvýšila presnosť a pri druhej dátovej sade zas zmenšila výpočetná náročnosť tréningu.

5.3.1 Odtiene sivej

Prvou z aplikovaných augmentácií bolo prevedenie upravenej dátovej sady dreva s rozlíšením 640×640 z experimentu 5.2.2 na odtiene sivej, s cieľom ešte viac zjednodušiť tréning

modelov. Modely boli následne natrénované na upravenej dátovej sade a výsledky týchto zmien boli následne porovnané s výsledkami z predchádzajúcich experimentov.

	Faster R-CNN		YOLOv8s	
	$AP(0,50)$	$AP(0,95)$	$AP(0,50)$	$AP(0,95)$
<i>quartzity</i>	0,135	0,077	0,127	0,057
<i>live_knot</i>	0,739	0,326	0,808	0,393
<i>marrow</i>	0,682	0,408	0,723	0,426
<i>resin</i>	0,694	0,296	0,715	0,361
<i>dead_knot</i>	0,774	0,375	0,853	0,456
<i>knot_with_crack</i>	0,546	0,33	0,675	0,466
<i>missing_knot</i>	0,749	0,382	0,74	0,44
<i>crack</i>	0,571	0,281	0,621	0,298
<i>all/mAP</i>	0,611	0,31	0,658	0,362

Tabuľka 5.5: Porovnanie presnosti modelov Faster R-CNN a YOLOv8s na upravenej dátovej sade *Large Scale Image Dataset of Wood Surface Defects* zmenšenej na rozlíšenie 640×640 a prevedenej do odtieňov sivej.

Pri porovnaní výsledkov dátovej sady prevedenej na odtiene sivej (tabuľka 5.5) s výsledkami dátovej sady z minulého experimentu s rozlíšením 640×640 (tabuľka 5.4) bol zaznamenaný mierny pokles celkovej mAP pre obidva modely, z 0,628 na 0,611 u Faster R-CNN a z 0,675 na 0,658 u YOLOv8s modelu.

V porovnaní dosiahnutých výsledkov s výsledkami z originálnej neupravenej dátovej sady (tabuľka 5.2) si model Faster R-CNN dokázal zachovať vysokú celkovú mAP (0,611 oproti pôvodnej 0,618). Výsledky modelu YOLOv8s na druhú stranu ukazujú výraznejšie zníženie celkovej mAP aj keď v niektorých kategóriách, ako napríklad *knot_with_crack* a *dead_knot*, sa presnosť mierne zvýšila. Napriek tomu, dosiahnutá presnosť stále umožňuje použitie modelu pre prototypovanie. Tieto zistenia ukazujú, že zníženie rozlíšenia a prevod na odtiene sivej môžu byť v určitých prípadoch vhodným kompromisom medzi výpočtovou náročnosťou a presnosťou detekcie.

5.3.2 Rozšírenie dátovej sady

V ďalšej časti som skúmal vplyv offline augmentácie dátovej sady NEU-DET z experimentu 5.2.1 na presnosť modelov. Pri tejto offline augmentácii bola použitá iba technika otočenia o 90° . Touto technikou sa podarilo rozšíriť dátovú sadu z pôvodných 1800 obrázkov na dvojnásobok. Treba poznamenať, že techniky zrkadlenia ako napríklad zrkadlenie sa už využívajú v online augmentáciách pri každom experimente.

Experiment ukázal mierny nárast presnosti modelov na augmentovanej dátovej sade v porovnaní s pôvodnou dátovou sadou, avšak nie výrazný. Podrobnejšie výsledky sú uvedené nižšie.

	Faster R-CNN		YOLOv8s	
	$AP(0,50)$	$AP(0,95)$	$AP(0,50)$	$AP(0,95)$
<i>crazing</i>	0,406	0,112	0,506	0,197
<i>inclusion</i>	0,762	0,366	0,79	0,426
<i>patches</i>	0,934	0,557	0,923	0,577
<i>pitted_surface</i>	0,806	0,491	0,792	0,501
<i>rolled-in_scale</i>	0,525	0,204	0,655	0,277
<i>scratches</i>	0,915	0,521	0,89	0,538
<i>all/mAP</i>	0,725	0,375	0,759	0,419

Tabuľka 5.6: Presnosti modelov Faster R-CNN a YOLOv8 na rozšírenej dátovej sade *NEU-DET* s rozlíšením 416×416 .

Výsledky experimentu ukazujú, že offline augmentácia dátovej sady priniesla zmiešané výsledky. Pri modeli Faster R-CNN sa celková presnosť (mAP) znížila z 0,743 (0,397 pri $AP(0,95)$) v pôvodnej dátovej sade na 0,725 (0,375 pri $AP(0,50)$) v augmentovanej dátovej sade. Model YOLOv8 zaznamenal mierne zvýšenie celkovej presnosti z 0,749 (0,437 pri $AP(0,95)$) na 0,759 (0,419 pri $AP(0,95)$). Tieto zmeny v presnosti modelov môžu byť spôsobené rozdielnou účinnosťou augmentácií na rôzne triedy objektov alebo interakciami medzi jednotlivými augmentáciami. V niektorých prípadoch mohla augmentácia zvýšiť zložitosť dát, čo zhoršilo presnosť modelov. Napríklad pri kategórii *rolled-in_scale* sa presnosť modelu Faster R-CNN zlepšila, zatiaľ čo presnosť modelu YOLOv8 v tejto kategórii poklesla.

5.3.3 Zvýšenie kontrastu

V tomto experimente som skúmal vplyv zvýšenia kontrastu na presnosť detekcie. V prvom prípade som vychádzal z dátovej sady z experimentu 5.2.1. Tejto dátovej sade bol pomocou nástroja Pinetools¹ na hodnotu 50. Následne som na takto upravenej dátovej sade natrénoval modely a výsledky porovnal s výsledkami (tabuľka 5.3) z experimentu 5.2.1.

	Faster R-CNN		YOLOv8s	
	$AP(0,50)$	$AP(0,95)$	$AP(0,50)$	$AP(0,95)$
<i>crazing</i>	0,413	0,128	0,535	0,209
<i>inclusion</i>	0,752	0,375	0,763	0,392
<i>patches</i>	0,919	0,578	0,898	0,56
<i>pitted_surface</i>	0,781	0,439	0,842	0,524
<i>rolled-in_scale</i>	0,499	0,196	0,646	0,293
<i>scratches</i>	0,920	0,55	0,922	0,557
<i>all/mAP</i>	0,714	0,378	0,767	0,423

Tabuľka 5.7: Presnosti modelov Faster R-CNN a YOLOv8 na upravenej dátovej sade *NEU-DET* so zvýšeným kontrastom.

Experiment ukázal rozdielny vplyv zvýšenia kontrastu na jednotlivé triedy poškodení. U modelu Faster R-CNN sa v porovnaní s výsledkami z minulého experimentu (tabuľka 5.3) celková presnosť znížila a úprava kontrastu nemala pozitívny vplyv. Naopak, pri modeli YOLOv8s sa celková presnosť zvýšila, ale pri vyššom prahu presnosti sú výsledky menej konzistentné. Pri porovnaní výsledkov modelov YOLO môžeme pozorovať pri triedach *crazing*

¹<https://pinetools.com/bulk-batch-change-image-contrast>

a *pitted_surface* výraznejší nárast presnosti. Tento nárast môže byť dôsledkom zvýraznenia jemných detailov poškodenia v dôsledku zvýšenia kontrastu. Pri modeli Faster R-CNN bol zaznamenaný pokles presnosti (AP 0,50) hlavne pri detekcii *rolled-in_scale* z 0,674 na 0,499. Tento pokles môže byť spôsobený tým, že zvýšenie kontrastu zmenilo textúru a vlastnosti obrazu takým spôsobom, že model bol menej schopný správne detekovať poškodenie.

Rovnakú zmenu kontrastu som vykonal aj na upravenej dátovej sade poškodení dreva z experimentu 5.3.1. Po úprave som na takto zmenenej dátovej sade natrénoval modely a ich výsledky porovnal s výsledkami (tabuľka 5.5) z toho istého experimentu.

	Faster R-CNN		YOLOv8s	
	<i>AP(0,50)</i>	<i>AP(0,95)</i>	<i>AP(0,50)</i>	<i>AP(0,95)</i>
<i>quartzity</i>	0,129	0,054	0,174	0,055
<i>live_knot</i>	0,8	0,343	0,814	0,399
<i>marrow</i>	0,73	0,437	0,745	0,435
<i>resin</i>	0,678	0,295	0,739	0,348
<i>dead_knot</i>	0,838	0,416	0,823	0,444
<i>knot_with_crack</i>	0,527	0,326	0,665	0,474
<i>missing_knot</i>	0,701	0,336	0,701	0,379
<i>crack</i>	0,533	0,228	0,601	0,287
<i>all/mAP</i>	0,617	0,304	0,658	0,352

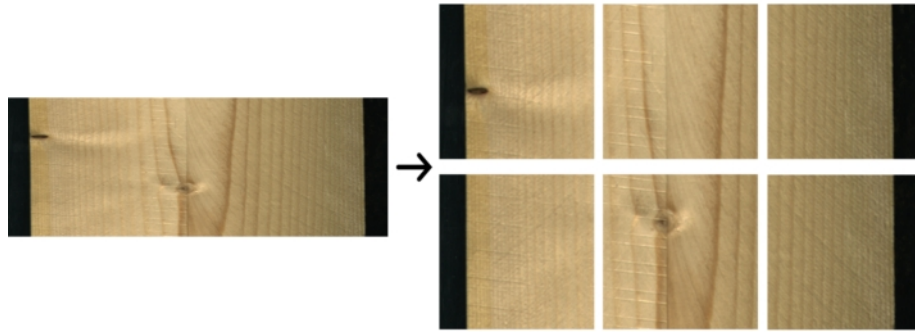
Tabuľka 5.8: Presnosti modelov Faster R-CNN a YOLOv8 na upravenej dátovej sade *Large Scale Image Dataset of Wood Surface Defects* so zvýšeným kontrastom.

Porovnaním výsledkov dátovej sady so zvýšeným kontrastom (tabuľka 5.8) a výsledkov modelov natrénovaných na pôvodnej dátovej sade (tabuľka 5.5) z experimentu 5.3.1 môžeme vidieť, že celková presnosť modelu Faster R-CNN sa mierne zvýšila, zatiaľ čo celková presnosť modelu YOLOv8 sa pri AP(0,50) nijako nezmenila a pri AP(0,95) sa mierne zhoršila. V tomto prípade môže byť kontrast medzi objektom a pozadím už dostatočne výrazný a zvýšenie kontrastu celého obrázka môže skôr spôsobuje straty detailov alebo zvýšenie šumu, čo negatívne ovplyvňuje presnosť modelu.

5.3.4 Metóda tiling

V poslednom experimente som sa pokúsil zlepšiť presnosť detekcie poškodení na dreve pomocou metódy nazývanej „tiling“. Cieľom tohto experimentu bolo overiť, či použitie tejto metódy môže prispieť k lepšej identifikácii malých poškodení, ktoré môžu byť pri väčších rozlíšeniach ťažšie rozpoznateľné.

Pôvodný obrázok s rozlíšením 2800×1024 som rozdelil na 6 menších častí, z ktorých každá mala rozlíšenie 416×416. Príklad takejto úpravy je znázornený na obrázku 5.5.



Obr. 5.5: Príklad jedného obrázku z upravenej dátovej sady.

Týmto som dosiahol zväčšenie malých poškodení. Následne som na takto upravenej dátovej sade natrénoval modely a ich presnosť porovnal s presnosťou modelov natrénovaných na originálnej dátovej sade.

	Faster R-CNN		YOLOv8s	
	<i>AP(0,50)</i>	<i>AP(0,95)</i>	<i>AP(0,50)</i>	<i>AP(0,95)</i>
<i>quartzity</i>	0,104	0,048	0,193	0,096
<i>live_knot</i>	0,707	0,3	0,735	0,366
<i>marrow</i>	0,617	0,381	0,551	0,345
<i>resin</i>	0,569	0,256	0,679	0,362
<i>dead_knot</i>	0,76	0,38	0,791	0,433
<i>knot_with_crack</i>	0,435	0,25	0,556	0,367
<i>missing_knot</i>	0,545	0,27	0,557	0,346
<i>crack</i>	0,516	0,213	0,538	0,239
<i>all/mAP</i>	0,532	0,262	0,578	0,319

Tabuľka 5.9: Presnosti modelov Faster R-CNN a YOLOv8 na upravenej dátovej sade *Large Scale Image Dataset of Wood Surface Defects* v odtieňoch sivej, kde bol každý obrázok rozdelený na 6 menších s rozlíšením 416×416 .

Z výsledkov vyplýva, že použitie metódy „tiling“ v tomto konkrétnom prípade neprišlo očakávané zlepšenie presnosti detekcie. Naopak, uplatnenie tilingu spôsobilo výrazný pokles presnosti pri oboch testovaných modeloch. Tieto zistenia naznačujú, že pri zlepšovaní detekcie malých poškodení táto metóda nie je účinná. Taktiež sa rozdelením jedného obrázku na 6 menších výrazne zvýšila výpočtová náročnosť tréningu.

Pokles presnosti mohla spôsobiť strata informácií o kontexte, keďže každá časť obrazu sa spracovávala samostatne. Taktiež sa niektoré poškodenia mohli ocitnúť na hranici medzi dvoma časťami obrazu a tým znemožnili ich detekciu. Tieto prípady by sa dali odstrániť zložitejšou implementáciou tejto metódy, ktorá by brala ohľad na širší kontext jednotlivých poškodení a zamedzila by rozdeleniu poškodenia na dva obrázky.

Kapitola 6

Záver

Táto záverečná práca bola zameraná na detekciu povrchového poškodenia materiálu na základe fotografie. Na úvod som si naštudoval a opísal základy spracovania obrazu, ktoré sú nevyhnutné pre pochopenie tejto problematiky. Následne som sa zameril na fungovanie neurónových sietí s dôrazom na konvolučné neurónové siete. Opísané boli rôzne metódy strojového učenia a detektory objektov. Tieto znalosti som využil pri výbere vhodných modelov na riešenie problému detekcie poškodenia, konkrétne som sa rozhodol pre modely Faster R-CNN a YOLOv8. Tieto modely boli následne tréňované na vybratých dátových sadách poškodení dreva a ocele, ktoré sú v práci podrobne opísané spolu s tréňovacím prostredím. Posledná časť bola venovaná experimentom, kde som vyhodnocoval výkon použitých modelov a skúmal vplyv rôznych augmentácií dátových sád na presnosť modelov.

Pri dátovej sade poškodení dreva sa podarilo výrazne znížiť výpočetné nároky na tréňovanie prevedením tejto dátovej sady do odtieňov sivej a zmenšením rozlíšenia. Pri týchto úpravách sa presnosť siete znížila, ale iba mierne. Výsledky ukazujú, že takéto úpravy môžu byť v určitých prípadoch vhodným kompromisom medzi výpočetnou náročnosťou a presnosťou detekcie a môžu nájsť využitie napríklad pri prototypovaní. Najvyššia presnosť bola zaznamenaná pri originálnej dátovej sade bez augmentácií s využitím modelu YOLOv8 a to 0,696 pri AP(0,50) a 0,389 pri AP(0,95). Pri dátovej sade povrchových poškodení ocele bola najvyššia presnosť zaznamenaná pri zvýšení rozlíšenia tejto dátovej sady a zvýšenom kontraste zároveň. V tomto prípade model YOLOv8 zaznamenal presnosť AP(0,50) 0,767 a AP(0,95) 0,423.

Ďalší vývoj v tejto oblasti by mohol smerovať k vytvoreniu špecifických metód augmentácie zameraných na konkrétne charakteristiky poškodenia materiálu. Ďalšie zlepšenie by mohlo priniesť zväčšenie rozsahu dátových sád a v niektorých prípadoch aj presnejšie označenie pravdivých označení. K lepšej presnosti a rýchlosti detekcie by mohlo prispieť aj použitie iných architektúr neurónových sietí alebo implementácia vlastných špecializovaných architektúr.

Literatúra

- [1] *Max-pooling* [online]. [cit. 2023-01-12]. Dostupné z: https://computersciencewiki.org/index.php/Max-pooling/_/Pooling.
- [2] AHSAN, N. *Large Scale Image Dataset of Wood Surface Defects*. 2022. Dostupné z: <https://www.kaggle.com/datasets/nomihsa965/large-scale-image-dataset-of-wood-surface-defects>.
- [3] BOCHKOVSKIY, A., WANG, C. a LIAO, H. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *CoRR*. 2020, abs/2004.10934. Dostupné z: <https://arxiv.org/abs/2004.10934>.
- [4] CANNY, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986, PAMI-8, č. 6, s. 679–698. DOI: 10.1109/TPAMI.1986.4767851.
- [5] CARPENTER, R. D. *Defects in hardwood timber*. US Department of Agriculture, Forest Service, 1989.
- [6] CHAPELLE, O., SCHOLKOPF, B. a ZIEN, A. Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews]. *IEEE Transactions on Neural Networks*. 2009, zv. 20, č. 3, s. 542–542.
- [7] DIAZ, A. *5 Aspects of Fully Convolutional Networks for Semantic Segmentation* [online]. [cit. 2023-01-17]. Dostupné z: <https://www.marscrowd.com/blog/text/fully-convolutional-networks-for-semantic-segmentation/>.
- [8] DING, X., ZHANG, X., MA, N., HAN, J., DING, G. et al. Repvgg: Making vgg-style convnets great again. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, s. 13733–13742.
- [9] DU, L., ZHANG, R. a WANG, X. Overview of two-stage object detection algorithms. *Journal of Physics: Conference Series*. Máj 2020, zv. 1544, s. 012033. DOI: 10.1088/1742-6596/1544/1/012033.
- [10] DWIVEDI, P. *Semantic Segmentation — Popular Architectures* [online]. [cit. 2023-03-16]. Dostupné z: <https://towardsdatascience.com/semantic-segmentation-popular-architectures-dff0a75f39d0>.
- [11] ERCAN OZTEMELI, S. G. Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*. 2018, č. 31, s. 127–182. Dostupné z: https://www.researchgate.net/profile/Samet-Gursev/publication/326584283_Literature_review_of_Industry_4_0_and_related_technologies/links/

[5c8e37bea6fdcc38175916f1/Literature-review-of-Industry-40-and-related-technologies.pdf](https://arxiv.org/abs/5c8e37bea6fdcc38175916f1/Literature-review-of-Industry-40-and-related-technologies.pdf).

- [12] FARHADI, A. a REDMON, J. Yolov3: An incremental improvement. In: Springer Berlin/Heidelberg, Germany. *Computer vision and pattern recognition*. 2018, sv. 1804, s. 1–6.
- [13] GE, Z., LIU, S., WANG, F., LI, Z. a SUN, J. *YOLOX: Exceeding YOLO Series in 2021*. 2021.
- [14] GIRSHICK, R. B. Fast R-CNN. *CoRR*. 2015, abs/1504.08083. Dostupné z: <http://arxiv.org/abs/1504.08083>.
- [15] GIRSHICK, R. B., DONAHUE, J., DARRELL, T. a MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*. 2013, abs/1311.2524. Dostupné z: <http://arxiv.org/abs/1311.2524>.
- [16] GONZALEZ, R. C. a WOODS, R. E. *Digital image processing (3rd Edition)*. Upper Saddle River, N.J.: Pearson Education, 2007. ISBN 978-0131687288.
- [17] HE, K., ZHANG, X., REN, S. a SUN, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*. IEEE. 2015, zv. 37, č. 9, s. 1904–1916.
- [18] HE, K., ZHANG, X., REN, S. a SUN, J. Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [19] HE, K., ZHANG, X., REN, S. a SUN, J. *Identity Mappings in Deep Residual Networks*. 2016.
- [20] HINTON, G. E. a SALAKHUTDINOV, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science*. 2006, zv. 313, č. 5786, s. 504–507. Dostupné z: <https://www.science.org/doi/abs/10.1126/science.1127647>.
- [21] HOWARD, A. G., ZHU, M., CHEN, B., KALENICHENKO, D., WANG, W. et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017.
- [22] HUANG, G., LIU, Z., MAATEN, L. van der a WEINBERGER, K. Q. Densely Connected Convolutional Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [23] HUANG, L., CHEN, C., YUN, J., SUN, Y., TIAN, J. et al. Multi-Scale Feature Fusion Convolutional Neural Network for Indoor Small Target Detection. *Frontiers in Neurorobotics*. 2022, zv. 16. DOI: 10.3389/fnbot.2022.881021. ISSN 1662-5218. Dostupné z: <https://www.frontiersin.org/articles/10.3389/fnbot.2022.881021>.
- [24] HUI, J. *MAP (mean Average Precision) for Object Detection* [online]. 2018 [cit. 2023-01-19]. Dostupné z: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>.

- [25] HULSTAERT, L. *A Beginner's Guide to Object Detection* [online]. [cit. 2023-01-17]. Dostupné z: <https://www.datacamp.com/tutorial/object-detection-guide>.
- [26] IAN GOODFELLOW, A. C. *Deep Learning*. MIT Press, 2016. ISBN 978-0-262-03561-3.
- [27] IOFFE, S. *Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models*. 2017.
- [28] J. BEYERER, F. P. L. a. C. F. *Machine Vision : Automated Visual Inspection: Theory, Practice and Applications*. Springer, 2016. ISBN 978-3-662-47794-6.
- [29] JAN, J. *Číslíková filtrace, analýza a restaurace signálů*. Vyd. 2. Brno: VUTIUM, 2002. ISBN 80-214-1558-4.
- [30] JAVATPOINT. *Supervised Machine Learning* [online]. 2021 [cit. 2023-03-02]. Dostupné z: <https://www.javatpoint.com/supervised-machine-learning>.
- [31] JOCHER, G. *YOLOv5 by Ultralytics*. 2020. Dostupné z: <https://github.com/ultralytics/yolov5>.
- [32] JOCHER, G., CHAURASIA, A. a. QIU, J. *YOLOv8 by Ultralytics*. Január 2023. Dostupné z: <https://github.com/ultralytics/ultralytics>.
- [33] JU, R.-Y. a. CAI, W. *Fracture Detection in Pediatric Wrist Trauma X-ray Images Using YOLOv8 Algorithm*. 2023.
- [34] KOCH, C., GEORGIEVA, K., KASIREDDY, V., AKINCI, B. a. FIEGUTH, P. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*. 2015, zv. 29, č. 2, s. 196–210. DOI: <https://doi.org/10.1016/j.aei.2015.01.008>. ISSN 1474-0346. Infrastructure Computer Vision. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1474034615000208>.
- [35] KODYTEK, P., BODZAS, A. a. BILIK, P. A large-scale image dataset of wood surface defects for automated vision-based quality control processes [version 2; peer review: 2 approved]. *F1000Research*. 2022, zv. 10, č. 581. DOI: 10.12688/f1000research.52903.2.
- [36] LI, C., LI, L., JIANG, H., WENG, K., GENG, Y. et al. *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*. 2022.
- [37] LIN, T.-Y., DOLLÁR, P., GIRSHICK, R., HE, K., HARIHARAN, B. et al. Feature pyramid networks for object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, s. 2117–2125.
- [38] LIU, S., QI, L., QIN, H., SHI, J. a. JIA, J. Path aggregation network for instance segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, s. 8759–8768.
- [39] LIU, W., ANGUELOV, D., ERHAN, D., SZEGEDY, C., REED, S. et al. Ssd: Single shot multibox detector. In: Springer. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. 2016, s. 21–37.

- [40] MARVIN MINSKY, S. P. *Perceptrons : an introduction to computational geometry*. Expanded ed., 4th print. Cambridge (Mass.): MIT press, 1990. ISBN 0-262-63111-3.
- [41] MCCULLOCH, W. S. a PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*. Springer. 1943, zv. 5, s. 115–133.
- [42] NIELSEN, M. A. *Neural networks and deep learning*. Determination press, 2015.
- [43] REDMON, J., DIVVALA, S. K., GIRSHICK, R. B. a FARHADI, A. You Only Look Once: Unified, Real-Time Object Detection. *CoRR*. 2015, abs/1506.02640. Dostupné z: <http://arxiv.org/abs/1506.02640>.
- [44] REDMON, J. a FARHADI, A. YOLO9000: Better, Faster, Stronger. *CoRR*. 2016, abs/1612.08242. Dostupné z: <http://arxiv.org/abs/1612.08242>.
- [45] REN, S., HE, K., GIRSHICK, R. a SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*. 2015, zv. 28.
- [46] RICHARD, S. *Computer Vision: Algorithms and Applications*. Springer London, Limited, 2010. ISBN 9781848829343.
- [47] ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*. American Psychological Association. 1958, zv. 65, č. 6, s. 386.
- [48] RUMELHART, D. E., HINTON, G. E. a WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*. Nature Publishing Group UK London. 1986, zv. 323, č. 6088, s. 533–536.
- [49] SIKUDOVA, E., CERNEKOVA, Z., BENESOVA, W., BERGER HALADOVA, Z. a KUCEROVA, J. *Počítačové videnie: Detekcia a rozpoznávanie objektov. - 1. vydanie*. Január 2014. 268 s. ISBN 978-80-87925-06-5.
- [50] SIMONYAN, K. a ZISSERMAN, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015.
- [51] SONG, K. a YAN, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*. Elsevier. 2013, zv. 285, s. 858–864.
- [52] SOVIT RATH, V. G. *Performance Comparison of YOLO Object Detection Models – An Intensive Study* [online]. [cit. 2023-04-14]. Dostupné z: <https://learnopencv.com/performance-comparison-of-yolo-models/>.
- [53] STANFORDUNIVERSITY. *Convolutional Neural Networks for Visual Recognition* [online]. [cit. 2023-01-11]. Dostupné z: <https://cs231n.github.io/convolutional-networks/>.
- [54] SUTTON, R. a BARTO, A. Reinforcement Learning: An Introduction. *IEEE Transactions on Neural Networks*. 1998, zv. 9, č. 5, s. 1054–1054.

- [55] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S. et al. Going Deeper With Convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.
- [56] UIJLINGS, J. R. R., SANDE, K. E. A. van de, GEVERS, T. a SMEULDERS, A. W. M. Selective Search for Object Recognition. *International Journal of Computer Vision*. 2013, zv. 104, č. 2, s. 154–171. Dostupné z: <https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013>.
- [57] WANG, C.-Y., BOCHKOVSKIY, A. a LIAO, H.-Y. M. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. 2022.
- [58] WANG, C.-Y., LIAO, H.-Y. M., WU, Y.-H., CHEN, P.-Y., HSIEH, J.-W. et al. CSPNet: A new backbone that can enhance learning capability of CNN. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, s. 390–391.
- [59] WENG, K., CHU, X., XU, X., HUANG, J. a WEI, X. EfficientRep: An Efficient Repvgg-style ConvNets with Hardware-aware Neural Network Design. *CoRR*. 2023, abs/2302.00386. DOI: 10.48550/arXiv.2302.00386. Dostupné z: <https://doi.org/10.48550/arXiv.2302.00386>.
- [60] XIANGHUA, X. A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques. *Electronic Letters on Computer Vision and Image Analysis*. 2008, č. 7, s. 1–22. Dostupné z: <https://pdfs.semanticscholar.org/b809/841f0f45fb91e70f9623cf70bd4514cf4f91.pdf>.
- [61] YASUDA, Y. D., CAPPABIANCO, F. A., MARTINS, L. E. G. a GRIPP, J. A. Aircraft visual inspection: A systematic literature review. *Computers in Industry*. 2022, zv. 141, s. 103695. DOI: <https://doi.org/10.1016/j.compind.2022.103695>. ISSN 0166-3615. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0166361522000926>.
- [62] ZEIDLER, A. *Lexikon vad dřeva*. 2011. Dostupné z: http://r.fld.czu.cz/vyzkum/multimedia/lexikon_vad/mechanicke_poskozeni.htm.
- [63] ZHAO, Z.-Q., ZHENG, P., XU, S.-T. a WU, X. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*. 2019, zv. 30, č. 11, s. 3212–3232.
- [64] ZHU, X. a GOLDBERG, A. Introduction to Semi-Supervised Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*. 2009, zv. 3, č. 1.

Príloha A

Obsah priloženej SD karty

Priložená SD karta obsahuje:

- models/ - priečinok s natrénovanými modelmi
- tex/ - priečinok so zdrojovými súbormi textovej časti práce
- bp.pdf - textová časť práce