



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

AUTOMATIZACE VIDEOEXTENZOMETRŮ S VYUŽITÍM METOD UMĚLÉ INTELIGENCE

AUTOMATION OF VIDEO EXTENSOMETERS USING ARTIFICIAL INTELLIGENCE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Vít Leinweber

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Bořek Ščerba

BRNO 2024

Zadání diplomové práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	Bc. Vít Leinweber
Studijní program:	Inženýrská mechanika a biomechanika
Studijní obor:	Inženýrská mechanika
Vedoucí práce:	Ing. Bořek Ščerba
Akademický rok:	2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Automatizace videoextenzometrů s využitím metod umělé inteligence

Stručná charakteristika problematiky úkolu:

Digitální korelace obrazu, jakožto bezkontaktní metoda pro měření povrchových deformací, nalézá čím dál častěji uplatnění v různých oblastech průmyslu, zejména pro testování mechanických vlastností materiálů. Pro tento účel se využívají videoextenzometry a s jejich rostoucí popularitou uživatelé kladou čím dál více důraz na jejich automatické a intuitivní ovládání. Jedním ze způsobů automatizace videoextenzometrů je rozpoznání typu měřeného vzorku a umístění vhodného softwarového nástroje do správné polohy na vzorek v reálném čase.

Cíle diplomové práce:

V rámci práce bude provedena rešerše metod umožňující "real-time" rozpoznání typu měřeného vzorku a určení jeho polohy. Z dostupných metod bude vybrána nejvhodnější a implementována s využitím databáze výrobce videoextenzometrů X-Sight. Práce bude doplněna experimentální částí, která jednak rozšíří dostupnou databázi, jednak bude sloužit k validaci implementované metody.

Seznam doporučené literatury:

WU, Xiongwei, Doyen SAHOO a Steven C.H. HOI. Recent advances in deep learning for object detection. Neurocomputing (Amsterdam) [online]. Elsevier B.V, 2020, 396, 39-64 [cit. 2022-10-17]. ISSN 0925-2312. Dostupné z: doi:10.1016/j.neucom.2020.01.085

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jiří Hlinka, Ph.D.
děkan fakulty

ABSTRAKT

Tato diplomová práce se zabývá automatizací použití videoextenzometrů s využitím metod umělé inteligence, a to konkrétně implementací vhodného algoritmu pro rozpoznání typu měřeného vzorku a umístění vhodného softwarového nástroje do správné polohy na vzorek v reálném čase. V teoretické části práce je popsána digitální korelace obrazu, počítačové vidění se zaměřením na rozpoznávání objektů v obraze a strojové učení se zaměřením na hluboké konvoluční neuronové sítě a jejich architektury pro detekci objektů v obraze. Na základě poznatků z teoretické části práce je zvolen algoritmus YOLOv8 jako nejvhodnější pro klasifikaci typu vzorku a určení jeho polohy v reálném čase. V praktické části práce je pomocí databáze výrobce videoextenzometrů vytvořen dataset obsahující tři typy těles určených k detekci. Tento dataset je dále rozšířen o vlastní snímky vzorků a augmentován. Zvolený algoritmus je pomocí vytvořeného datasetu trénován, optimalizován a testován. V rámci optimalizace je použit genetický algoritmus a náhodné prohledávání prostoru hyperparametrů. Nejlepší natrénované modely algoritmu YOLOv8 jsou navzájem porovnány na dvou testovacích množinách, a je vybrán nejlepší z nich. Dále je popsána práce s tímto modelem jako detektorem integrovaným do příslušného softwaru pro práci s videoextenzometry. Na závěr je navržen postup na korekci vlivu nezarovnaných vzorků na snímcích vstupujících do detektoru.

Klíčová slova

videoextenzometr, DIC, počítačové vidění, strojové učení, konvoluční neuronové sítě, YOLOv8

ABSTRACT

This thesis deals with the automation of the use of video extensometers using artificial intelligence methods, specifically the implementation of a suitable algorithm for the recognition of the type of sample to be measured and the placement of a suitable software tool in the correct position on the sample in real-time. The theoretical part of the thesis describes digital image correlation, computer vision with a focus on object recognition, and machine learning with a focus on deep convolutional neural networks and their architectures for object detection. Based on the findings from the theoretical part of the work, the YOLOv8 algorithm is chosen as the most suitable algorithm for classifying the type of sample and determining its location in real-time. In the practical part of the thesis, a dataset containing three types of samples is created using the database of video extensometer manufacturer. The dataset is extended with images containing samples to be detected and augmented. The selected algorithm is trained, optimized, and tested using the created dataset. A genetic algorithm and a random search of the hyperparameter space are used in the optimization process. The best trained models of the YOLOv8 algorithm are compared with each other on two test sets and the best one is selected. Furthermore, the work with this algorithm as a detector integrated into the corresponding software for working with video extensometers is described. Finally, a method for correcting the effect of potential misalignment of samples in the images entering the detector is proposed.

Key words

video extensometers, DIC, computer vision, machine learning, convolutional neural networks, YOLOv8

BIBLIOGRAFICKÁ CITACE

LEINWEBER, Vít. *Automatizace videoextenzometrů s využitím metod umělé inteligence*. Brno, 2024. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/154057>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Bořek Ščerba.

PROHLÁŠENÍ

Prohlašuji, že jsem Diplomovou práci na téma *Automatizace videoextenzometrů s využitím metod umělé inteligence* vypracoval samostatně s použitím odborné literatury a pramenů, uvedených v seznamu, který tvoří přílohu této práce.

17. května 2024

Vít Leinweber

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat své rodině a všem, kteří mě během studia podporovali. Dále bych rád poděkoval Ing. Janu Pokornému Ph.D. za poskytnutí externí pracovní plochy a společnosti X-Sight za poskytnutí přístupu do jejich databáze.

Největší dík však patří mému vedoucímu Ing. Bořkovi Ščerbovi za zkušené vedení, cenné rady a neskonalou trpělivost.

OBSAH

ÚVOD	11
1 Problémová situace, problém a cíle práce	12
2 Současný stav poznání	13
2.1 Digitální korelace obrazu	13
2.1.1 2D Digitální korelace obrazu	14
2.1.2 3D Digitální korelace obrazu	15
2.1.3 Hlavní zdroje chyb při použití digitální korelace obrazu	17
2.2 Počítačové vidění	19
2.2.1 Typické úlohy počítačového vidění	19
2.2.2 Rozpoznávání objektů	20
2.2.3 Tradiční metody rozpoznávání objektů	22
2.3 Strojové učení	23
2.3.1 Učení s učitelem	23
2.3.2 Učení bez učitele	24
2.3.3 Zpětnovazební učení	24
2.3.4 Základní úlohy učení s učitelem	24
2.3.5 Algoritmy učení s učitelem	24
2.3.6 Neuronové sítě	26
2.3.7 Konvoluční neuronové sítě	27
2.3.8 Učení neuronových sítí	29
2.3.9 Architektury konvolučních neuronových sítí pro rozpoznávání objektů	32
3 Analýza problému a volba metod pro jeho řešení	40
3.1 Nástroje použité pro implementaci	40
4 Tvorba datasetu	41
4.1 Databáze X-Sight	41
4.2 Doporučení autorů YOLOv8	42
4.3 Tvorba třídy tahových vzorků	42
4.4 Tvorba třídy vzorků betonářské výztuže	44
4.5 Tvorba třídy vrubovaných těles	46
4.6 Augmentace snímků	48
4.7 Výsledný dataset	50
4.7.1 Doplnění datasetu o snímky bez objektů	51
4.8 Validáčnı́ a testovacı́ množiny	51
4.9 Anotace snímků	52
5 Implementace modelu	54
5.1 Volba velikosti modelu	54
5.2 Trénink základního modelu	55

5.2.1	Porovnání optimalizačních algoritmů	60
5.3	Optimalizace hyperparametrů	61
5.3.1	Prostor hyperparametrů	62
5.3.2	Genetický algoritmus	62
5.3.3	Náhodné prohledávání prostoru hyperparametrů	64
5.4	Porovnání natrénovaných modelů	66
5.4.1	Porovnání na testovací množině v2	66
5.4.2	Porovnání na testovací množině v3	68
5.5	Volba nejlepšího modelu	69
5.6	Vyhodnocení nejlepšího modelu	71
6	Práce s modelem	77
6.1	Zpracování výstupu z detektoru	78
6.1.1	Třída tahových vzorků	79
6.1.2	Třída betonářské výztuže	80
6.1.3	Třída vrubovaných těles	80
6.2	Korekce nezarovnaných vzorků	82
7	Diskuse	86
8	Závěr	88
	BIBLIOGRAFIE	90
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	96
	PŘÍLOHY	98

ÚVOD

V současné době nacházejí metody umělé inteligence stále větší uplatnění napříč různými oblastmi inženýrství. Jednou z těchto oblastí je metoda digitální korelace obrazu. Digitální korelace obrazu je poměrně jednoduchá optická metoda, která mimo jiné umožňuje bezkontaktní měření povrchových deformací, což je výhodné v mnoha průmyslových aplikacích.

Jednou z průmyslových aplikací digitální korelace obrazu jsou videoextenzometry, které umožňují bezkontaktní testování mechanických vlastností materiálů. S videoextenzometry přichází i software k jejich ovládání a vyhodnocení výsledků mechanických zkoušek. Při vývoji komerčního průmyslového softwaru je mimo jiné kladen důraz na automatické a intuitivní ovládání. V případě průmyslového softwaru k ovládání videoextenzometrů může být jedním ze způsobů automatizace rozpoznávání typu měřeného vzorku a umístění vhodného softwarového nástroje do správné polohy na vzorku, a to v reálném čase. Tento požadavek otevírá možnosti použití vhodných metod počítačového vidění, konkrétně s využitím strojového učení, které nabízí nepřehledné množství algoritmů pro řešení takových úloh. Pro uživatele může být tak práce s videoextenzometry efektivnější a jejich výrobce s automatizovaným softwarem může být konkurenceschopnější.

Tato práce se zabývá automatizací použití videoextenzometrů s využitím metod umělé inteligence, konkrétně rozpoznáním typu měřeného vzorku a určení jeho polohy pro umístění vhodného softwarového nástroje v reálném čase. Teoretická část pojednává o současném stavu poznání oblastí, do kterých práce zasahuje. Praktická část se pak zabývá implementací vhodných metod a jejich validací.

1 Problémová situace, problém a cíle práce

Problémové situace

Automatizace použití videoextenzometrů, respektive softwaru k práci s nimi, pomocí rozpoznání typu měřeného vzorku a určení jeho polohy pro umístění vhodného softwarového nástroje s sebou přináší několik úskalí. První z nich je volba vhodné metody, pomocí níž je určen typ vzorku a vzorek je lokalizován. V dnešní době se za tímto účelem používají v drtivé většině algoritmy na bázi strojového učení, což s sebou přináší i nutnost práce s daty a vytvoření dostatečného datasetu pro jejich implementaci. Druhým úskalím je implementace vhodného algoritmu, která mimo jiné souvisí i s jeho validací a testováním. Třetím úskalím je vytvoření postupu pro práci s implementovaným algoritmem jako nástrojem integrovaným do softwaru pro práci s videoextenzometry.

Problém

Určení, implementace a validace vhodné metody počítačového vidění na bázi umělé inteligence pro rozpoznání typu měřeného vzorku a lokalizaci jeho polohy.

Cíle práce

- Rešerše metod umožňujících „real-time“ rozpoznání typu měřeného vzorku a určení jeho polohy.
- Nalezení nejvhodnější metody a její implementace s využitím databáze výrobce videoextenzometrů X-Sight.
- Rozšíření dostupné databáze, které bude sloužit k validaci implantované metody.
- Validace a testování implementované metody.

2 Současný stav poznání

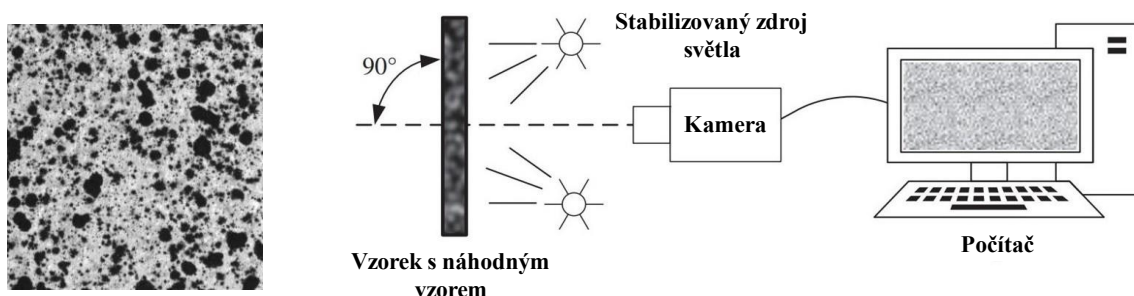
Tato kapitola se zabývá současným stavem poznání v oborech, do kterých práce zasahuje. Je v ní pojednáno o digitální korelaci obrazu, počítačovém vidění se zaměřením na rozpoznávání objektů v obraze a strojovém učení.

2.1 Digitální korelace obrazu

Tradiční způsoby měření přetvoření, například pomocí tenzometrů, jsou vhodné pro měření uniformní deformace v diskrétních bodech vyšetřovaných oblastí. Nicméně v technické praxi většinou k uniformní deformaci nedochází. Často dochází k heterogenní deformaci tělesa. Využití tenzometrů je pak limitované jejich lokálním přístupem, na základě kterého může být obtížné obecně popsat deformaci tělesa pomocí diskrétních bodů [1]. Z tohoto důvodu bylo postupem času vyvinuto mnoho metod, jenž umožňují měření přetvoření v celé vyšetřované oblasti. Jedním z odvětví těchto metod jsou metody optické. Optické metody používají pro vyšetřování přetvoření informaci, kterou nese povrch tělesa, respektive náhodný vzor na povrch nanesený. Mezi optické bezkontaktní metody patří například holografická interference, elektronická tečkovaná interferometrie a digitální korelace obrazu [1].

Digitální korelace obrazu, dále pouze DIC, je optická, nedestruktivní a bezkontaktní metoda v dnešní době v praxi hojně používaná, a to díky mnoha výhodám, jenž přináší. Mezi naznačené výhody patří např. relativně jednoduchá příprava měřícího systému, poměrně nízké nároky na prostředí experimentu, možnost automatizace metody, možnost automatizace zpracování změřených dat a dále možnost použití v mikroskopickém i makroskopickém měřítku [1].

S použitím DIC souvisí i splnění určitých předpokladů. Prvním předpokladem je přítomnost tzv. náhodné struktury („*speckle pattern*“) na povrchu tělesa, které slouží jako nositel informace o deformaci povrchu [1]. Tato struktura povrchu může být na vzorku přítomná přirozeně nebo může být nanášena pomocí technik nanášení náhodného vzoru. Často používanou technikou je nanášení bílé barvy na povrch tělesa a přidání náhodného kontrastního vzoru barvou černou např. sprejováním. Při nanášení náhodného vzoru je nutné dbát na to, aby naneseným vzorem nebyly ovlivněny mechanické vlastnosti tělesa. Správně vytvořená náhodná struktura by měla být silně kontrastní, stochastická a izotropní [1]. Druhým předpokladem použití DIC je dostatečně silný stabilizovaný světelný zdroj [1]. Třetím předpokladem je použití zařízení pro záznam obrazu s dostatečným rozlišením [1]. Na obrázku 1 vlevo je znázorněna náhodná struktura vytvořená výše zmíněnou technikou.



Obrázek 1: Speckle pattern (vlevo) a sestava 2D-DIC (vpravo, převzato z [1] upraveno)

2.1.1 2D Digitální korelace obrazu

Použití 2D digitální korelace obrazu (2D-DIC) přidává ke třem zmíněným obecným předpokladům další dva požadavky. První požadavek spočívá v dodržení rovinnosti měřené plochy vzorku. Druhý požadavek pak spočívá v dodržení kolmosti měřené plochy vzorku na optickou osu. Při nedodržení těchto požadavků může dojít k zanesení chyb měření [1]. Na obrázku 1 vpravo je znázorněno schéma měřící sestavy 2D-DIC.

Podstata DIC spočívá ve zpracování dvou po sobě jdoucích snímků pořízených v různých krocích zatěžování. Snímky jsou pak rozděleny do menších podoblastí. Principem DIC je sledování pozic podoblastí v referenčním (výchozím) snímku a hledání odpovídajících pozic v cílovém snímku, a to pomocí korelačních algoritmů [1]. Schéma principu 2D-DIC je zobrazeno na obrázku 2.

Podrobněji, podoblast o $(2M+1) \times (2M+1)$ pixelů ve výchozím (referenčním) snímku obsahuje referenční středový bod P , který je na obrázku 2 znázorněn červeným křížkem. Tento bod je vstupem do korelačních algoritmů a je hledána jeho pozice ve snímku cílovém. K popisu lineární transformace bodů podoblastí je definována bázová funkce prvního řádu a to následovně:

$$x^* = u + \frac{\delta u}{\delta x} \Delta x + \frac{\delta u}{\delta y} \Delta y \quad (1)$$

$$y^* = v + \frac{\delta v}{\delta x} \Delta x + \frac{\delta v}{\delta y} \Delta y \quad (2)$$

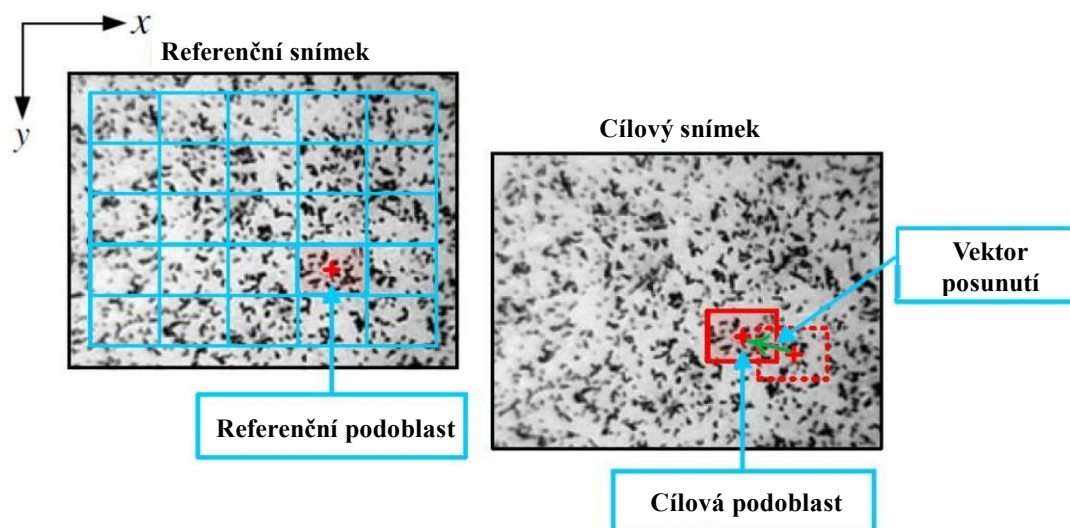
kde x^* a y^* jsou výsledné posuvy podoblasti, u a v jsou posuvy referenčního středového bodu P , Δx a Δy jsou vzdálenosti obecného bodu Q v podoblasti vůči referenčnímu bodu P a $\frac{\delta u}{\delta x}$, $\frac{\delta u}{\delta y}$, $\frac{\delta v}{\delta x}$, $\frac{\delta v}{\delta y}$ jsou gradienty posunutí podoblasti [1].

Pro určení míry podobnosti mezi výchozím a cílovým snímkem, respektive k nalezení podoblasti z referenčního snímku ve snímku cílovém, je určena korelační funkce pro všechny možné polohy podoblasti v cílovém snímku a vybrána její hodnota, která nabývá největší shody. Nejpoužívanější korelační funkce jsou ZNSSD („Zero-mean normalized sum-of-square difference criterion“) a ZNCC („Zero-mean normalized cross-correlation criterion“) díky jejich odolnosti vůči změnám světlosti a kontrastu snímků [1]. Zmíněné korelační funkce jsou převzaty z [1] a popsány vztahy:

$$C_{ZNSSD}(p) = \sum_{i=-M}^M \sum_{j=-M}^M \left[\frac{f(x, y) - f_m}{\sqrt{\sum_{i=-M}^M \sum_{j=-M}^M [f(x, y) - f_m]^2}} - \frac{g(x^*, y^*) - g_m}{\sqrt{\sum_{i=-M}^M \sum_{j=-M}^M [g(x, y) - g_m]^2}} \right]^2 \quad (3)$$

$$C_{ZNCC}(p) = \sum_{i=-M}^M \sum_{j=-M}^M \left[\frac{[f(x, y) - f_m] \times [g(x^*, y^*) - g_m]}{\sqrt{\sum_{i=-M}^M \sum_{j=-M}^M [f(x, y) - f_m]^2 \times \sum_{i=-M}^M \sum_{j=-M}^M [g(x^*, y^*) - g_m]^2}} \right] \quad (4)$$

kde $f(x, y)$ jsou hodnoty stupně šedi v jednotlivých pixelech v podoblasti referenčního snímku, $g(x^*, y^*)$ jsou hodnoty stupně šedi v jednotlivých pixelech podoblasti v cílovém snímku a f_m a g_m jsou průměrné hodnoty stupně šedi v podoblasti v referenčním snímku a podoblasti cílového snímku [1].



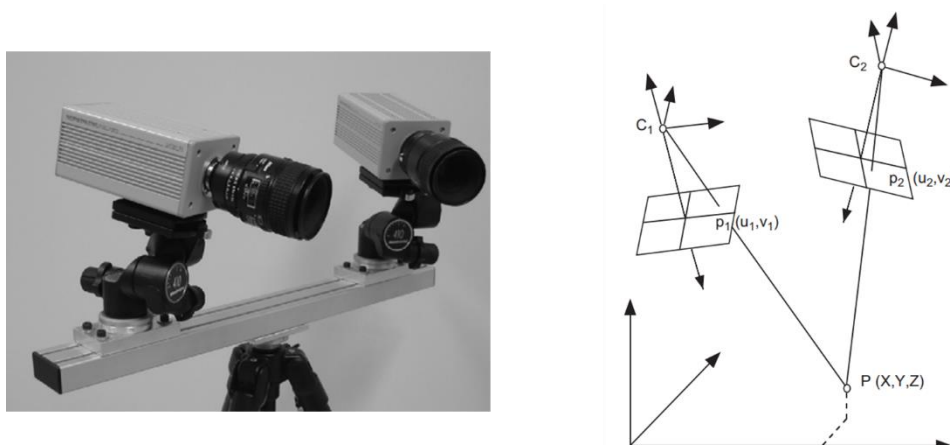
Obrázek 2: Schéma principu 2D-DIC (převzato z [1] a upraveno)

Nalezením odpovídajících poloh podoblastí referenčního snímku ve snímku cílovém lze definovat vektor posunutí podoblastí, respektive jejich středových bodů, a definovat tak vektorové pole deformace celé oblasti [1].

2.1.2 3D Digitální korelace obrazu

Na rozdíl od 2D-DIC, která pro měření mechanických vlastností na povrchu tělesa využívá jednu kameru a klade požadavky na rovinnost vzorku a kolmost optické osy vůči vzorku, 3D digitální korelace obrazu (3D-DIC) dokáže určit mechanické vlastnosti na povrchu obecného 3D tělesa. Jinými slovy nemusí být dodržena zmíněná rovinnost vzorku a kolmost vzorku na optickou osu kamery, a to díky použití dvou nebo více kamer (obrázek 3 vlevo) a vytvoření tzv. stereovize [2].

Binokulární stereovize je metoda počítačového vidění pro rekonstrukce tvaru 3D objektu pomocí dvou úhlů pohledu. V kontextu 3D-DIC, dvou snímků pořízených pomocí dvou kamer v prostoru. Binokulární stereovize je znázorněna na obrázku 3 vpravo, kde bod $P(X, Y, Z)$ je obecný bod v prostoru, $p_1(u_1, v_1)$ a $p_2(u_2, v_2)$ jsou stereo projekce ve snímcích z kamer s optickými středy v bodech C_1 a C_2 . Z dvojice snímků je pak možné vypočítat souřadnice

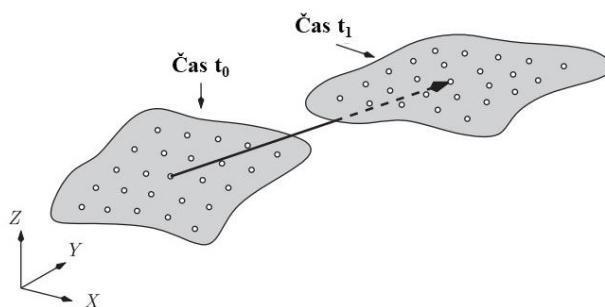


Obrázek 3: Dvojice kamero pro 3D-DIC (vlevo) a schéma binokulární stereovize (vpravo). Oba obrázky jsou převzaty z [2].

obecného bodu v prostoru pomocí triangulace, a to za splnění dvou předpokladů. Prvním předpokladem je, že relativní pozice a orientace obou kamer je známá. Druhým pak je, že projekce p_1 a p_2 jsou projekcemi právě jednoho bodu [2].

První z dvou zmíněných předpokladů úzce souvisí s kalibrací kamer. Kalibrace stereovize, respektive kamer jenž ji vytváří, spočívá v určení vnitřních parametrů každé kamery a určení jejich relativní pozice a vzájemné orientace. Jednou z efektivních metod kalibrace je zaznamenání 10 až 20 snímků daného rovinného vzoru (šablony) v různých prostorových orientacích. Použitím fotogrammetrického vyrovnání svazku („*photogrammetric bundle adjustment*“) lze pak určit všechny požadované kalibrační parametry. Výhodou této metody je, že neklade velký důraz na kvalitu šablony a není třeba znát relativní pohyb šablony mezi jednotlivými kalibračními snímky. Další výhodou je, že metoda bere v úvahu i distorzi čočky kamery [2].

Druhý zmíněný předpoklad souvisí s problémem hledání vazby, respektive odpovídajících projekčních bodů (stereo projekcí), mezi dvojicí snímků. Existuje mnoho metod pro řešení tohoto problému a obecně se dají rozdělit do dvou kategorií. První kategorií jsou metody, které pro přiřazení odpovídajících stereo projekcí používají ze snímků extrahované primitivy jako jsou hrany a obrysy. Tyto metody jsou obvykle rychlé, protože extrahováním primitivních rysů z obrazu dojde ke snížení množství pixelů, se kterými jsou prováděny výpočty. Nevýhodou však je, že vznikne velmi hrubá mapa přiřazených bodů a tím pádem je znalost deformace povrchu tělesa omezena jen na určená místa [2]. Druhou kategorií jsou metody, které hledají odpovídající podoblasti ve dvojici snímků na základě porovnávání hodnot stupně šedi a hledání jejich vnitřní podobnosti. Podoblast z prvního snímku se na druhém snímku nachází s největší pravděpodobností v místě, kde korelační funkce dosahuje maxima s ohledem na to, že podoblast, jenž se v prvním snímku jeví jako čtverec, může být v druhém snímku obecně jiného tvaru kvůli prostorovému charakteru tělesa. Při znalosti vzájemné orientace kamer, jejich relativní polohy vůči sobě a souřadnic bodu P v obou snímcích je možno pomocí triangulace dopočítat jeho souřadnice v prostoru. Při opakování tohoto postupu vyhledávání podoblasti z prvního snímku ve snímku druhém pro pokrytí celého společného zorného pole obou kamer je možno vytvořit prostorová rekonstrukce snímaného tělesa. Porovnáním pozic bodů povrchu snímaného tělesa v rozdílném čase jsou pak určeny posuvy těchto bodů podobně jako u 2D-DIC [2]. Z dvojice snímků pořízených v čase t_0 je možno popsáním postupem vytvořit prostorový tvar tělesa v čase t_0 . Z dvojice snímků pořízených v čase t_1 je analogicky možno vytvořit prostorový tvar tělesa v čase t_1 . Porovnáním



Obrázek 4: Výpočet 3D-DIC posuvů (převzato z [2]
a upraveno)

odpovídajících bodů v jednotlivých časech je možno určit posuvy těchto bodů podobně jako u 2D-DIC (obrázek 4) [2].

Metoda 3D digitální korelace obrazu je vybudována na základu metody 2D digitální korelace obrazu a rozšiřuje ho o použití metod stereo vize, čímž se staví do pozice (s ohledem na uvedená omezení) univerzálního nástroje pro bezkontaktní určování mechanických vlastností povrchu prostorového tělesa [2].

2.1.3 Hlavní zdroje chyb při použití digitální korelace obrazu

Využití experimentálních metod přináší i určitá omezení v jejich přesnosti z důvodu výskytu chyb měření. V kontextu DIC mohou být potenciální zdroje chyb shrnuty do dvou skupin. První skupinou jsou chyby související s výpočetním algoritmem DIC. Druhou skupinou potenciálních chyb jsou pak chyby související s procesem pořizování digitálních snímků [1].

Zdroje chyb spojené s výpočetním algoritmem DIC souvisí s volbou vhodného řádu bázové funkce pro popis transformace bodů v podoblastech, přesností korelačního kritéria, přesností sub-pixelové interpolace funkce stupně šedi a velikostí vyhodnocovaných podoblastí. Často používané bázové funkce pro popis transformace bodů v podoblastech jsou bázové funkce prvního řádu. Lava a spol. v článku [4] navrhli, že pro silně heterogenní pole deformace na povrchu těles je vhodnější používat bázové funkce vyššího řádu pro přesnější popis pole deformace, čímž dojde ke snížení chyby, nicméně ke zvýšení výpočtové náročnosti. K docílení sub-pixelové přesnosti je použita interpolace mezi hodnotami stupně šedi v jednotlivých pixelech. Ideální interpolační funkce by byla sinusoida, nicméně tato funkce není vhodná pro numerické kalkulace, proto jsou pro interpolaci často používány B-spline interpolační funkce nebo bikubické interpolační polynomy, čímž může dojít k zanesení systematických chyb [1]. Velký vliv na přesnost DIC má zvolená velikost vyšetřované podoblasti hledané mezi referenčním a cílovým snímkem. Velikost takové podoblasti by měla být dostatečně velká, aby obsahovala dostatečné množství unikátních vzorů k dosažení spolehlivému přiřazení odpovídající podoblasti mezi snímky. Nicméně příliš velká podoblast může významně navýšit výpočetní čas a může dojít k chybnému zachycení gradientu deformace za předpokladu, že velikost podoblasti je větší než velikost oblasti deformace [1].

Zdroje chyb spojené s procesem pořizování digitálních snímků vstupujících do DIC souvisí s kvalitou náhodného vzoru na povrchu tělesa, distorzí obrazu způsobenou čočkou kamery, pohybem kamery, šumem v experimentálním prostředí a primárně pro 2D-DIC, pohybem vzorku nerovnoběžně s rovinnou kolmou na optickou osu kamery [1]. Dostatečná kvalita nanášeného náhodného vzoru na povrchu tělesa je základem použití metody DIC. Pro ohodnocení kvality tohoto vzoru existují lokální a globální parametry. Lokální parametry reflektují kvalitu náhodného vzoru na úrovni podoblastí snímku, a to pomocí zprůměrování rozdílů hodnot intenzity osmi sousedních pixelů nebo součtem čtverců gradientu intenzity podoblasti v horizontálním a vertikálním směru [1]. Globální parametry reflektují kvalitu náhodného vzoru celého povrchu tělesa. Jednoduchým parametrem je průměrná velikost prvku náhodného vzoru [1]. Distorze obrazu způsobená čočkou kamery úzce souvisí s kvalitou systému, respektive kamery, pro snímání obrazu. Pro vysoce přesná měření je nutná kalibrace pro kompenzaci tohoto vlivu [1]. Další možný zdroj chyb související s kamerou, respektive její čočkou, je zahřívání aparatury vlivem ohřevu měřicího systému. V literatuře je uvedena závislost mezi chybou měřeného přetvoření a teplotou, která je u delších měření nutná kompenzovat předehřátím kamery před začátkem měření na provozní teplotu a to i po dobu 1

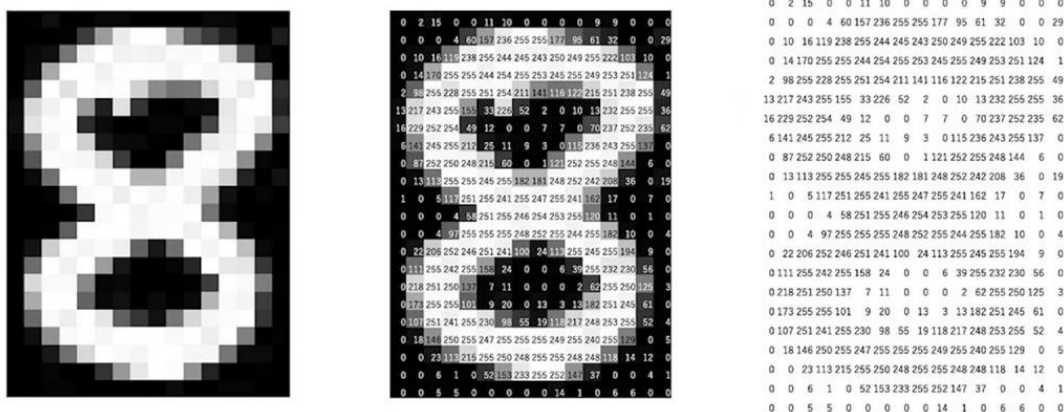
až 2 hodin [1]. Zdroj chyb v DIC může být i šum experimentálního prostředí a měřicího systému. Potenciální zdroje šumu jsou např. kolísající intenzita osvětlení, kabeláž soustavy a termální šum prostředí. V kontextu 2D-DIC je jeden z velkých potenciálních zdrojů chyb pohyb mimo rovinu vzorku, což je vliv jdoucí přímo proti předpokladům splnění při použití 2D-DIC. V jistém slova smyslu tomuto jevu nelze nikdy zabránit a jeho vliv se projevuje relativně vůči rozsahu a kontextu měření [1]. Pokud lze vyhodnotit, že během experimentu dochází k nezanedbatelnému pohybu mimo rovinu vzorku, je možno tento vliv částečně kompenzovat použitím telecentrické čočky, případně přistoupit k použití stereovize, respektive 3D-DIC [1].

2.2 Počítačové vidění

Počítačové vidění je obor, který se zabývá získáváním a analýzou obrazu a následnou extrakcí a pochopením informací, jenž obraz přenáší, a to pomocí počítače [5]. Z inženýrského hlediska se počítačové vidění zabývá vytvářením metod a algoritmů pro zpracování obrazu, které počítači umožňují zpracovat vizuální informace podobně jako lidský zrakový systém. Díky tomu je možno automatizovat mnoho úkonů, jenž vyžadují vizuální pochopení informací v obraze [6].

Počítačové vidění je velice rozsáhlý obor, který staví na základech z mnoha odvětví vědy a techniky. Primárním předmětem této práce je rozpoznávání typu měřeného vzorku a určení jeho polohy, což je klasická úloha rozpoznávání objektů, což je popsáno dále. Z toho důvodu tato podkapitola uvádí stručný teoretický základ napříč oborem počítačového vidění, ale do hloubky se zabývá výhradně uvedenými tématy, jenž souvisí s předmětem práce.

Počítačové vidění pracuje s digitálním obrazem. Digitální obraz se dá definovat jako diskrétní reprezentace analogového obrazu do matice, jejíž elementy se nazývají pixely [5]. V případě, že se jedná o obraz černobílý, hodnoty pixelů reprezentují úroveň stupně šedi, respektive snímanou intenzitu světla pomocí obrazového snímače (čipu). V barevném obraze je každý pixel, respektive jeho hodnota, reprezentovaná vektorem čísel, jenž představuje komponenty barev (kanály). Typickým barevným modelem je model RGB, který má tři kanály, jeden pro červenou komponentu, jeden pro zelenou komponentu a jeden pro modrou komponentu [5]. Na obrázku 5 je vlevo zobrazen digitální obraz čísla 8, uprostřed jsou zobrazeny spolu s odstíny šedi i číselné hodnoty jednotlivých pixelů a vpravo jsou zobrazeny pouze číselné hodnoty úrovně šedi jednotlivých pixelů.



Obrázek 5: Digitální reprezentace obrazu (převzato z [3])

2.2.1 Typické úlohy počítačového vidění

Počítačové vidění nachází uplatnění v mnoha oborech od medicíny, autonomního řízení vozidel, počítačové grafiky až po různá odvětví inženýrství. Mezi typické úlohy počítačového vidění patří rekonstrukce obrazu, analýza pohybu, rekonstrukce scény nebo 3D objektu a rozpoznávání objektů [5]. Jednou z dalších úloh je i vytvoření stereovize, o které je pojednáno v podkapitole 2.1.2.

Rekonstrukce obrazu úzce souvisí s odstraněním nežádoucích jevů v obraze. Mezi nežádoucí jevy patří například šum (multiplikativní, aditivní a impulzní) a rozmazání obrazu

vlivem pohybu obrazového snímače. Jedním z používaných způsobů, jak odstranit zmíněné nežádoucí jevy je filtrace obrazu [5, 62].

Analýza pohybu v obraze souvisí se sledováním sekvence snímků pořízených v čase a určování rychlostí sledovaných podoblastí, případně objektů v obraze nebo pohybu snímacího zařízení (kamery) vůči snímané scéně [5].

Rekonstrukce scény nebo 3D objektu souvisí s určováním 3D modelu pomocí sekvence snímků. Kromě stereovize je další metodou poměrně jednoduchá a přesná metoda rekonstrukce pomocí strukturovaného osvětlování, která spočívá v nasvícení předem definovaného známého vzoru na povrch tělesa nebo oblast scény. Pomocí deformace nasvíceného vzoru lze pak spočítat hloubku a geometrické parametry nasvícené entity [7].

Typickou úlohu v počítačovém vidění je rozpoznávání objektů. Tato úloha úzce souvisí s předmětem této práce, a proto je o ní pojednáno samostatně v následující podkapitole.

2.2.2 Rozpoznávání objektů

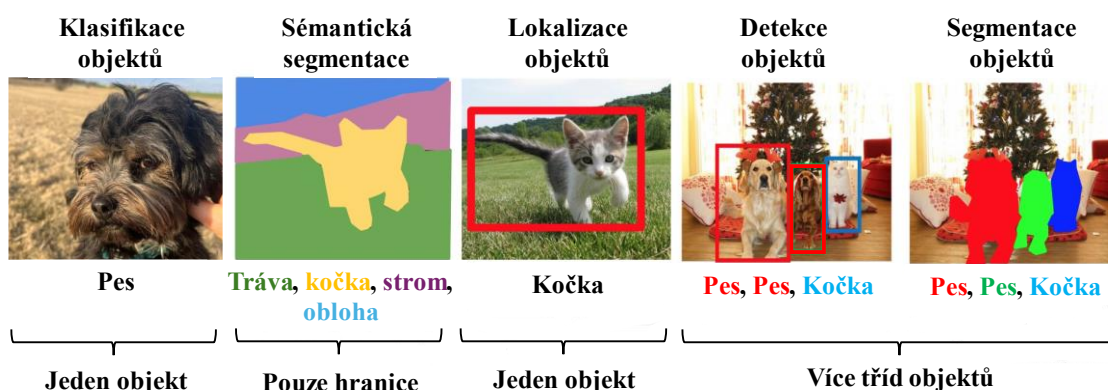
Klasickou úlohou počítačového vidění je rozpoznávání objektů v obraze. Rozpoznávání objektů lze rozdělit na několik dílčích přístupů, jež spolu úzce souvisí. Tyto přístupy jsou klasifikace objektů („*object classification*“), lokalizace objektů („*object localization*“), detekce objektů („*object detection*“), sémantická segmentace („*semantic segmentation*“) a segmentace objektů („*instance segmentation*“) [8,61].

Klasifikace objektů se zabývá určením třídy objektu z množiny definovaných tříd. Na vstupu do klasifikačního algoritmu je obrázek s jedním objektem a výstupem je jeho označení (jeho třída) [8,61]. Klasifikace je znázorněna na obrázku 6 vlevo.

Lokalizace objektů umožňuje umístit hraniční obdélník („*bounding box*“) do místa hledaného objektu. Na vstupu do lokalizačního algoritmu je obrázek s jedním nebo více objekty jedné třídy a na výstupu je jeden nebo více hraničních obdélníků lokalizujících jejich polohu [8,61]. Lokalizace je zobrazena na obrázku 6 uprostřed.

Detekce objektů umožňuje kombinaci klasifikace objektů a jejich lokalizace. Vstupem do detekčního algoritmu je obrázek s jedním nebo více objekty a výstupem jsou pak hraniční obdélníky s příslušnými označeními, jež reprezentují jejich třídu příslušnosti [8,61]. Detekce objektů je znázorněna na obrázku 6 na druhém místě zprava.

Sémantická segmentace umožňuje rozdělit obraz na jednotlivé spojitě související podoblasti. Na rozdíl od detekce, sémantická segmentace nerozeznává mezi jednotlivými



Obrázek 6: Znázornění jednotlivých přístupů rozpoznávání objektů v obraze (převzato z [60] a upraveno)

objekty stejné třídy, obrázek pouze rozděluje na podoblasti stejné sémantické příslušnosti [8,61]. Sémantická segmentace je znázorněna na obrázku 6 na druhém místě zleva.

Segmentace objektů (též segmentace instancí z anglického termínu „*instance segmentation*“) je složení detekce objektů a sémantické segmentace. Cílem segmentace objektů je rozpoznat jednotlivé objekty v obraze, uvést jejich příslušné označení (třídu) a znázornit jejich polohu pomocí pixelové masky [8]. Jinými slovy lze říct, že segmentace objektů je speciální případ detekce objektů, kdy je poloha objektu v obraze reprezentovaná jeho tvarem (pixelovou maskou) místo hraničního obdélníku [8]. Segmentace objektů je znázorněna na obrázku 6 vpravo.

Výkonnostní metriky související s problematikou rozpoznávání objektů

S popisovanou problematikou souvisí definování metrik pro hodnocení přesnosti a kvality rozpoznávání. Mezi základní metriky patří přesnost (P, „*precision*“) a citlivost (R, „*recall*“). Přesnost je bezrozměrná metrika nabývající hodnot z intervalu $\langle 0; 1 \rangle$ a je definována takto:

$$P = \frac{TP}{TP + FP} \quad (5)$$

kde *TP* („*True positives*“) představuje počet skutečných pozitivů a *FP* („*False positives*“) počet falešných pozitivů [5,8]. Citlivost je stejně jako přesnost bezrozměrná metrika nabývající hodnot z intervalu $\langle 0; 1 \rangle$ a je definována takto:

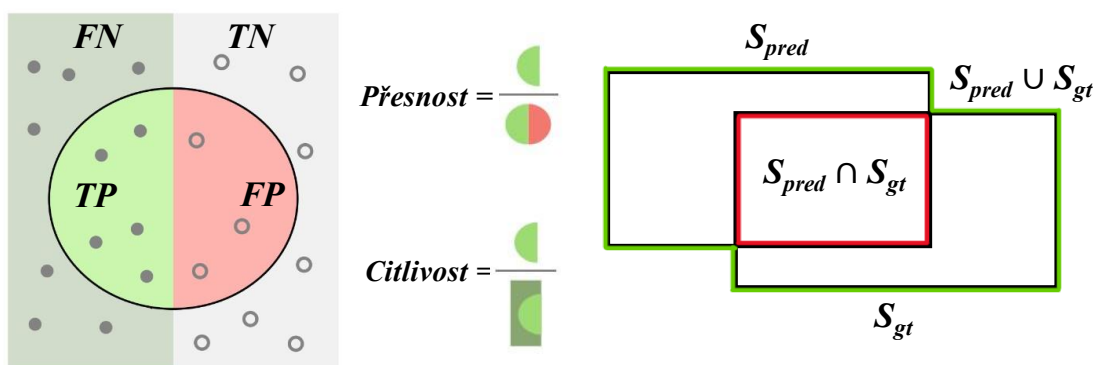
$$R = \frac{TP}{TP + FN} \quad (6)$$

kde *FN* („*False negatives*“) představuje počet falešných negativů [5,8]. Jinými slovy, *FN* představuje počet nerozpoznaných objektů, které by měly být rozpoznány. Přesnost a citlivost jsou graficky znázorněny v levé části obrázku 7.

V kontextu hraničních obdélníků, případně pixelové masky, je pro určení přesnosti a citlivosti nutno definovat parametr IoU („*Intersection over Union*“). IoU je dáno vztahem:

$$IoU = \frac{S_{pred} \cap S_{gt}}{S_{pred} \cup S_{gt}} \quad (7)$$

kde S_{pred} představuje plochu určeného hraničního obdélníku (objektu) a S_{gt} představuje plochu skutečného hraničního obdélníku (objektu) [8]. Čítec v rovnici 7 představuje průnik ploch



Obrázek 7: Znázornění přesnosti a citlivost (vlevo, převzato z [59] a upraveno), určení IoU (vpravo).

hraničních objektů (na obrázku 7 vpravo vyznačeno červeně) a jmenovatel představuje jejich sjednocení (na obrázku 7 vpravo vyznačeno zeleně). Pro určení, zda se jedná o správnou nebo špatnou lokalizaci hraničního objektu se určuje mezní hodnota IoU (prahová hodnota IoU). Často používanou prahovou hodnotou IoU je hodnota 0.5 [8].

2.2.3 Tradiční metody rozpoznávání objektů

Metody rozpoznávání objektů by se daly obecně rozdělit do dvou kategorií. Metody založené na hlubokém učení a tradiční metody rozpoznávání objektů [8]. V této podkapitole je stručně pojednáno o tradičních metodách rozpoznávání objektů, a to se zaměřením na detekci objektů, která je kombinací lokalizace a klasifikace. V dnešní době se však v drtivé většině případů používají metody umělé inteligence, a to především na bázi hlubokých konvolučních neuronových sítí, které jsou popsány v následující kapitole [8].

Tradiční metody pro detekci objektů lze rozdělit do tří kroků [8]. Cílem prvního kroku je vytyčit podoblasti obrazu, kde by se mohl vyskytovat hledaný objekt, respektive hledané objekty. Tyto podoblasti se označují ROI („*Regions of interest*“). Jedním z možných přístupů je rozdělit obraz do podoblastí pomocí posouvajícího se okna, kde každá takto vzniklá podoblast může obsahovat hledaný objekt, jinými slovy, může se jednat o ROI. Pro zachycení objektů různých velikostí může být velikost posouvajícího se proměnná [8]. Cílem druhého kroku je extrakce vektoru charakteristických rysů („*feature vector*“, někdy též příznaků), které obsahují sémantické informace o jednotlivých ROI. Toto je provedeno pomocí vizuálních deskriptorů („*visual descriptors*“), což jsou algoritmy určené k detekci a popisu lokálních rysů v obraze [8]. Mezi často používané vizuální deskriptory patří SIFT („*Scale Invariant Feature Transform*“) a SURF („*Speeded Up Robust Features*“). V posledním kroku je cílem přiřadit oblastem zájmu jednotlivé třídy, pokud se jedná o hledané objekty. Ke klasifikaci tříd objektů v ROI z vektorů charakteristických rysů jsou často použity algoritmy SVM („*Support vector machines*“) [8]. SVM se ve své podstatě zařazují do strojového učení, nicméně jejich použití se datuje před ostatní metody tohoto oboru (podkapitola 2.3.5).

Většina tradičních metod pro detekci objektů v obraze má určité limity. Při prvním kroku, kdy dochází k vytyčení ROI, může být těžké rozlišit objekty, které jsou blízko u sebe. Dalším omezením může být i to, že vizuální deskriptory používané u tradičních metod jsou často ručně a heuristicky navrženy a nemají dostatečně zobecňovací schopnosti [8].

2.3 Strojové učení

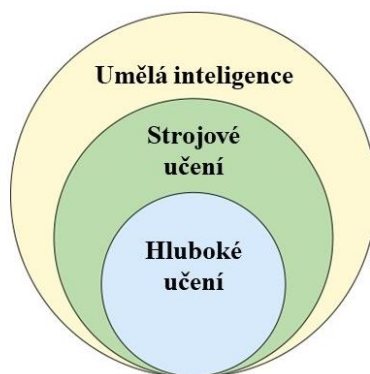
Strojové učení („*Machine learning*“) je obor z oblasti umělé inteligence (obrázek 8), který se zabývá vytvářením algoritmů a technik, jež se dokážou učit z poskytnutých dat [10]. Procesem učení je rozuměno získávání znalostí, dovedností a zkušeností na základě kterých lze zobecňovat informace a provádět úkony dle stanovených požadavků bez explicitních instrukcí [9,10]. Strojové učení v dnešní době nachází uplatnění téměř ve všech oborech, a to především na místech, kde je nevýhodné aplikovat tradiční algoritmy [11].

Strojové učení, jak již název napovídá, je proces, při kterém se počítač učí. To obsahuje analýzu a hledání vzorů v datech. Pro jednoduché úlohy je vhodné vytvořit algoritmus, který počítači přesně řekne, jak dané úlohy splnit a při tomto procesu k žádnému učení nedochází, nicméně pro splnění komplexních úloh, při kterých je nutné porozumět složitým procesům nebo pochopit vnitřní reprezentaci určitých systémů, může být složité vytvořit požadovaný algoritmus. V tomto případě je výhodné použít metody strojového učení tak, aby si počítač vytvořil vhodný algoritmus sám [12]. Strojové učení by se dalo jinými slovy popsat jako proces, při které se počítač snaží napodobit lidský mozek [13].

Metody strojového učení se dají obecně rozdělit do tří kategorií dle paradigmat učení. Zmíněné kategorie jsou učení s učitelem, učení bez učitele a zpětnovazební učení.

2.3.1 Učení s učitelem

Při učení s učitelem jsou modelu předkládána data, která obsahují vstupní parametry s jejich odpovídajícími výstupními hodnotami (někdy také vzory) [14]. Jinými slovy, data jsou označena (anotována). Model se takto učí hledat závislost a vazby mezi vstupy a výstupy. Pro zjednodušení si lze model, jenž je podroben procesu učení s učitelem, představit jako funkci o jedné proměnné s neznámým funkčním předpisem, přičemž do procesu učení vstupují dvojice jednotlivých hodnot proměnných s odpovídajícími funkčními hodnotami. Předkládáním takových dvojic dat je model pomocí učicího algoritmu schopen pochopit závislost mezi vstupními proměnnými a výstupními funkčními hodnotami a dále určovat výstupy na základě vstupů, které nebyly obsaženy v učících (trénovacích) datech, jinými slovy model je schopen zobecňovat i mimo učící (trénovací) množinu [10]. Učení s učitelem úzce souvisí s tématem této práce, proto je jeho problematice věnováno několik dalších podkapitol (podkapitola 2.3.5 a dále).



Obrázek 8: Strojové učení jako podobor umělé inteligence (převzato z [15])

2.3.2 Učení bez učitele

Na rozdíl od učení s učitelem při učení bez učitele nejsou data označena. Není znám požadovaný výstup. Cílem je najít vnitřní reprezentaci dat. Typickou úlohou pro využití algoritmů na bázi učení bez učitele je shlukování, což je proces, při kterém jsou objekty zařazovány do skupin s podobnými vlastnostmi. Mezi další úlohy patří například hledání anomálií v datech nebo odhadování hustot pravděpodobnosti [10].

2.3.3 Zpětnovazební učení

Zpětnovazební učení je proces učení, kde data nejsou označena stejně jako u učení bez učitele, nicméně při provedení určité akce je známá míra jejího úspěchu [61]. Algoritmus se takto učí, jak se chovat na základě pozorování určitého prostředí. Každou akci vůči tomuto prostředí provází určitá reakce, u níž lze stanovit ohodnocení (míru úspěchu). Proces učení je pak snaha maximalizovat míru úspěchu provedených akcí [13]. Pro prostředí je často reprezentováno jako Markovský rozhodovací proces [61]. Zpětnovazební učení se používá například v teorii her, teorii řízení nebo při optimalizaci na základě pozorování prostředí [10].

2.3.4 Základní úlohy učení s učitelem

Mezi dvě základní úlohy učení s učitelem patří regresní úloha a klasifikační úloha. Cílem obou úloh je naučit daný algoritmus vazbu mezi vstupy a výstupy, přesněji řešeno najít matematickou závislost mezi vstupy a výstupy.

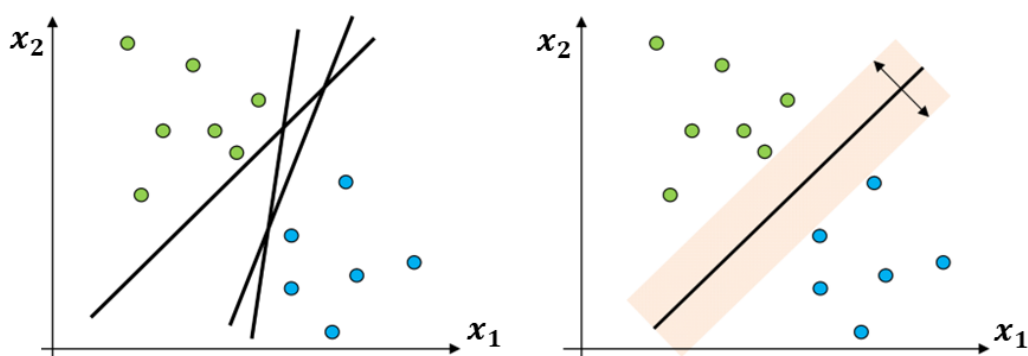
Výstupem regresní úlohy je spojitá proměnná (případně vektor proměnných), jejichž hodnoty závisí na vstupních proměnných [13]. Příkladem regresní úlohy může být stanovení průhybu volného konce z jedné strany vetknutého ocelového nosníku při působení síly na volném konci. Vstupními parametry jsou síla, geometrie nosníku a materiálové charakteristiky. Výstupem je pak průhyb volného konce [61].

Výstupem klasifikační úlohy je příslušnost do definovaných tříd, jinými slovy, výstupem této úlohy je diskretní proměnná, jež reprezentuje označení vstupu (příslušnost) [13]. Příkladem klasifikační úlohy může být filtrování emailů. Vstupní entitou je příchozí email. Výstupem je označení, zda se jedná nebo nejedná o spam.

2.3.5 Algoritmy učení s učitelem

V oboru učení s učitelem je mnoho algoritmů, které se využívají v praxi pro regresní i klasifikační úlohy [13]. V rámci této podkapitoly jsou stručně popsány pouze vybrané, jež se hojně využívají. V dnešní době asi nejznámějším algoritmem učení s učitelem jsou neuronové sítě („*Neural networks*“, zkratka NN). Neuronové sítě a jejich modifikace, jelikož jsou pro tuto práci velmi významné, jsou popsány v samostatných podkapitolách.

Support vector machines (SVM) neboli metoda podpůrných vektorů je algoritmus učení s učitelem, který se používá zejména ke klasifikaci (řešení klasifikační úlohy), ale lze ho využít i pro regresní analýzu (regresní úlohu). Základní úlohou SVM je lineární klasifikace do dvou tříd příslušnosti [18]. Cílem úlohy je oddělit jednotlivé třídy pomocí vhodné nadroviny, která prostor příznaků trénovací množiny dat rozděluje do tříd ležících v opačných poloprostorech. Ve 2D je hledanou nadrovinou přímka, ve 3D pak rovina. Hledaná nadrovina je optimální za předpokladu, že minimum vzdáleností bodů od této nadroviny je co největší [16,61]. Na obrázku 9 vlevo jsou znázorněny možné nadroviny pro lineární separaci dat ve 2D. Na



Obrázek 9: SVM ve 2D (převzato z [61] a upraveno)

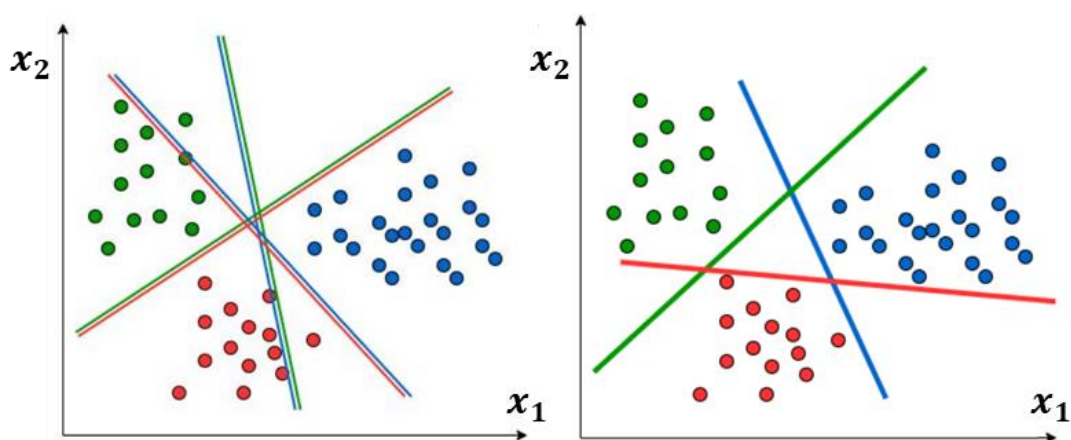
zmíněném obrázku vpravo je pak hledaná optimální nadrovina rozdělující data do příslušných tříd (příslušnost do tříd reprezentována barvou).

Pro klasifikaci více tříd (N tříd) je nutné použít $\frac{N(N-1)}{2}$ support vector machines, respektive N support vector machines podle toho, jestli se jedná o typ klasifikace jedna na jedna („one to one“) nebo klasifikace jedna na ostatní („one to rest“) [18]. Oba typy klasifikace jsou zobrazeny na obrázku 10.

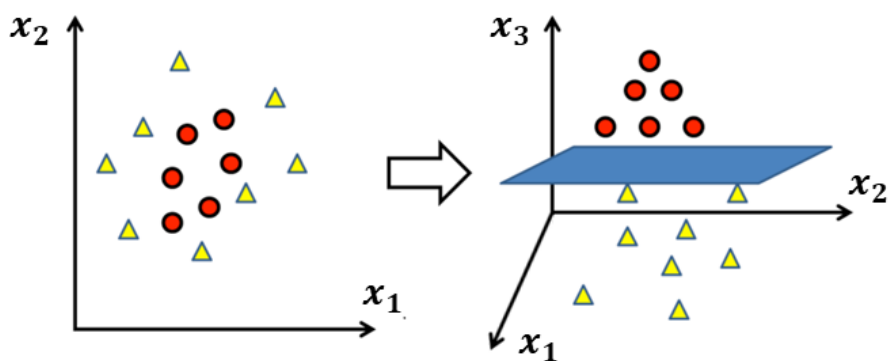
Pro řešení lineárně neseparabilních úloh se využívá jádrová transformace („kernel transformation“, někdy taky „kernel trick“). Vstupní data (vstupní vektory) se transformují přes jádro, což může být nelineární funkce, čímž dojde ke zvýšení dimenze úlohy. Ve vyšší dimenzi lze úlohu separovat lineárně [16,18]. Jádrová transformace je znázorněna na obrázku 11.

SVM se často uplatňují například při rozpoznávání textu nebo při rozpoznávání objektů v obraze, což bylo uvedeno v kapitole o tradičních metodách rozpoznávání objektů [16].

Mezi další algoritmy učení s učitelem patří například Lineární regrese, Logistická regrese nebo Naivní Bayesovský klasifikátor. **Lineární regrese** je klasická úloha hledání lineární závislosti mezi vstupy a výstupem [13]. Lineární regrese se často využívá k hledání závislosti mezi parametry, u kterých je očekáváno lineární chování. **Logistická regrese** se zabývá odhadem pravděpodobnosti určitého jevu na základě skutečností, jež mohou ovlivnit jeho výskyt. Logistická regrese nachází využití například při zpracování přirozeného jazyka [19].



Obrázek 10: Klasifikace jedna na jedna (vlevo) a klasifikace jedna na ostatní (vpravo) (obrázek převzat z [63] a upraven)



Obrázek 11: Jádrová transformace (převzato z [64] a upraveno)

Naivní Bayesovský klasifikátor pracuje s pravděpodobnostním modelem, který používá podmíněné pravděpodobnosti jevů. Jeho výhodou je, že dobře pracuje s daty obsahující šum [13]. Naivní Bayesovský klasifikátor se používá například pro filtrování emailů [61].

2.3.6 Neuronové sítě

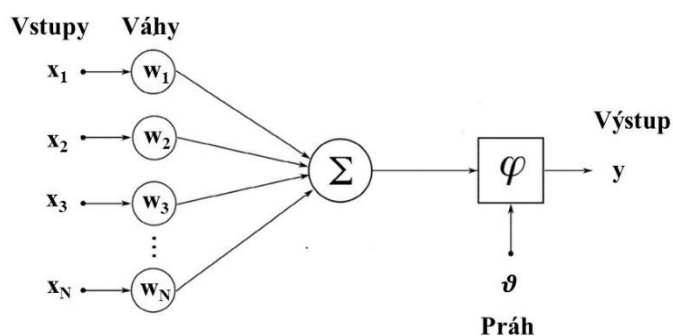
V dnešní době asi nejznámější algoritmy strojového učení s učitelem jsou neuronové sítě, které jsou zjednodušeným modelem neuronů a jejich propojení v lidském mozku [20]. Základním stavebním prvkem neuronové sítě je neuron. Často používaný model neuronu je znázorněn na obrázku 12. Vztah mezi vstupy a výstupem tohoto neuronu je následující:

$$y = \varphi \left(\sum_{i=1}^N (w_i \cdot x_i) - \vartheta \right) \quad (8)$$

kde x_i jsou vstupy („inputs“) do neuronu, w_i jsou váhy („weights“) a ϑ je práh neuronu („bias“). Funkce φ je pak aktivační funkce neuronu a její funkční hodnota je výstup („output“) neuronu. Suma součinů vah a vstupů v argumentu aktivační funkce se nazývá potenciál neuronu (p) [20].

Samotný neuron s Heavisideovou funkcí (jednotkový skok) jako aktivační funkcí je binární klasifikátor, též označovaný jako perceptron. Aktivační funkce je typicky nelineární, monotónní, neklesající a diferencovatelná. Požadavek nelinearity je z toho důvodu, aby byla do systému vnesena nelinearita a výstupy tak nebyly pouze lineární kombinací vstupů [20]. Přehled nejpoužívanějších aktivačních funkcí je uveden v tabulce 1.

Neuronová síť, jak již název napovídá, se skládá z více než jednoho neuronu. Typicky z několika neuronů v jedné nebo více skrytých vrstvách. Způsob propojení jednotlivých neuronů a vrstev udává topologie neuronové sítě. Často používanými topologiemi jsou plně propojené dopředné sítě a rekurentní neuronové sítě. Dopředná, plně propojená neuronová síť



Obrázek 12: Model neuronu (převzato z [61] a upraveno)

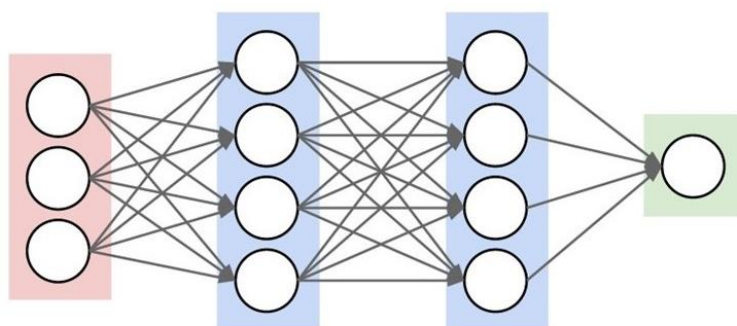
s dvěma skrytými vrstvami je znázorněna na obrázku 13. Na základě počtu vrstev, respektive skrytých vrstev, se rozlišuje klasické učení od tzv. hlubokého učení („Deep learning“). Obecně se jedná o hluboké učení ve chvíli, kdy má neuronová síť více než 2 skryté vrstvy. Podstata hlubokého učení spočívá v tom, že skryté vrstvy hledají v datech postupně komplexnější rysy (příznaky) [20].

Tabulka 1: Používané aktivační funkce [21] doplněné o charakteristiku z [61]

Aktivační funkce	Předpis	Obor hodnot	Charakteristika a číslo vztahu
Sigmoida	$\varphi(p) = \frac{1}{1 + e^{-p}}$	$\varphi(p) \in (0,1)$	Snadno diferencovatelná. (9)
Hyperbolický tangens	$\varphi(p) = \tanh(p)$	$\varphi(p) \in (-1,1)$	Snadno diferencovatelná. (10)
ReLU	$\varphi(p) = \max(0, p)$	$\varphi(p) \in \langle 0, \infty \rangle$	Často se používá u CNN. (11)
Softmax	$\sigma(p_i) = \frac{e^{p_i}}{\sum_{j=1}^n e^{p_j}}$	$\sigma(p) \in (0,1)$	Používá se u klasifikačních úloh na výstupní vrstvě. (12)

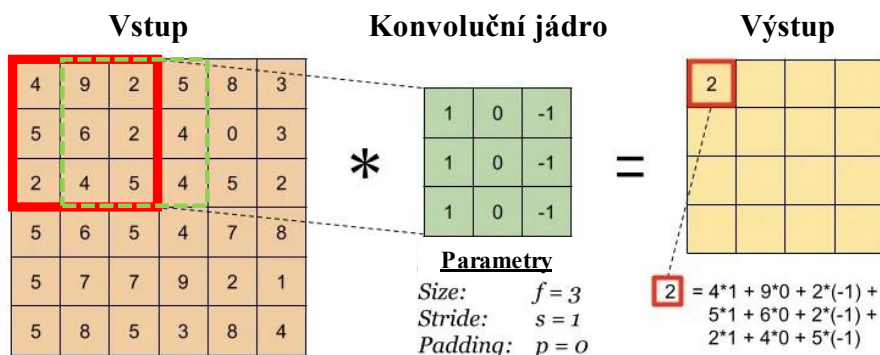
2.3.7 Konvoluční neuronové sítě

Jednou z modifikací klasické neuronové sítě, jež byla popsána v předchozí podkapitole je konvoluční neuronová síť (CNN z anglického „Convolutional neural network“). Princip sítě jako takové zůstává stejný s tím rozdílem, že jsou přidány tzv. konvoluční vrstvy a poolingové vrstvy. Konvoluční neuronové sítě se typicky využívají v případě, když je vstup vícerozměrný, například při zpracování více rozměrných signálu (obraz), kdy je nutné využívat informaci o prostorovém uspořádání [20,21].



Vstupní vrstva Skrytá vrstva 1 Skrytá vrstva 2 Výstupní vrstva

Obrázek 13: Plně propojená dopředná neuronová síť (převzato z [20] a upraveno)

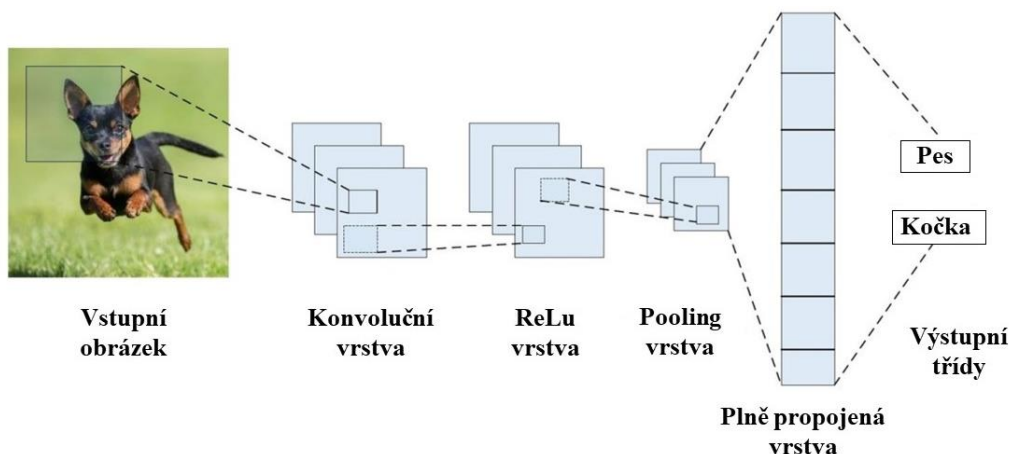


Obrázek 14: Konvoluce a parametry konvoluce v CNN (převzato z [17] a upraveno)

Konvoluční vrstva obsahuje jednotlivé konvoluční neurony, které se taky označují jako filtry. Na jednotlivých filtrech v konvoluční vrstvě dochází ke konvoluci vstupu s konvolučním jádrem (filtrem). Výstupem je pak obraz, ve kterém jsou zvýrazněny požadované rysy (příznaky). Při procesu učení (popsán v následující podkapitole) se tyto filtry přizpůsobují tak, aby bylo v datech zvýrazněno to, co je pro danou úlohu podstatné [20,21]. Konvoluce je znázorněna na obrázku 14.

Konvoluční jádra (filtry) jsou typicky čtvercová a jsou často velikosti 3×3 nebo 5×5 . Při vícerozměrném vstupu, což může být například barevný obraz, který má tři barevné kanály (typicky RGB), může vstupní konvoluční vrstva obsahovat filtry, které jsou ve tvaru $5 \times 5 \times 3$, případně $3 \times 3 \times 3$. Pro tyto filtry jsou definovány parametry počet pixelů posunutí konvolučního jádra (*stride*) a přidání nul kolem hranic obrazu (*padding*), respektive do matice, jež ho reprezentuje. Tyto parametry umožňují upravovat velikost výstupu z konvoluce [20,21]. Na obrázku 14 jsou tyto parametry uvedeny a znázorněny, kde je navíc parametr *size*, jež představuje velikost konvolučního jádra (v tomto případě 3×3). Na obrázku je posunutí konvolučního jádra (*stride*) znázorněno zeleným rámečkem. Parametr *padding* je uveden jako nula, protože konvoluční jádro nepřesahuje za hranice vstupní matice.

Pooling vrstva ve většině případů následuje konvoluční vrstvu. Využívá se pro zredukování rozměru dat. Často používanou poolingovou metodou je max-pooling, který slučuje 2×2 pole pixelů do jednoho pixelu, jehož hodnota je maximum z původních čtyř hodnot pixelů. Dalšími možnými poolingovými metodami jsou například min-pooling, kdy se



Obrázek 15: Zjednodušené schéma CNN (převzato z [21] a upraveno)

ze slučované oblasti bere nejmenší hodnota pixelu nebo average-pooling, kdy se zprůměrují hodnoty pixelu ze slučované oblasti a výsledná hodnota se zaokrouhlí na celé číslo [20,21].

Jednotlivé konvoluční vrstvy, respektive jejich filtry, obecně nemusí být spojeny se všemi filtry na vrstvě předešlé. Jinými slovy, konvoluční neuronová síť nemusí být plně propojena, a to z toho důvodu, že vstupy do sítě mohou být velké, respektive datově objemné, což může znamenat extrémní výpočetní náročnost. Místo toho se často používá lokální konektivita, kdy určité neurony (filtry) jsou spojeny s určitou oblastí vstupního obrazu [20,21]. Nicméně topologií CNN je mnoho a často se liší dle konkrétních řešených úloh.

Na konci konvoluční neuronové sítě je pomocí filtrů extrahována příznaková mapa („*feature map*“), což je struktura obsahující vyfiltrované rysy vstupních dat, která typicky vstupuje do plně propojené vrstvené neuronové sítě na jejíž konci je aktivační funkce podle typu úlohy [20,21]. Zjednodušené schéma konvoluční neuronové sítě je znázorněno na obrázku 15.

2.3.8 Učení neuronových sítí

Stěžejním procesem při práci s neuronovými sítěmi je jejich učení. Procesu učení se v kontextu neuronových sítí typicky říká trénování. Z matematického pohledu je neuronová síť funkce, u níž jsou jednotlivé váhy a prahy neuronů množina trénovatelných (nastavitelných) parametrů. U konvolučních neuronových sítí jsou váhy i hodnoty konvolučního jádra (filtru). Proces učení je pak minimalizace chybové funkce („*loss function*“), jenž určuje míru odlišnosti výstupu ze sítě od správného výstupu [20,21]. Často používané chybové funkce jsou uvedeny v tabulce 2.

Tabulka 2: Chybové funkce (převzaty z [21], IoU z [65])

Chybová funkce	Předpis	Charakteristika
Střední kvadratická chyba (MSE)	$E_{loss} = \frac{1}{2N} \sum_{j=1}^N (o_j - y_j)^2$	Vhodná pro regresní úlohy. (13)
Křížová entropie	$E_{loss} = - \sum_{j=1}^M y_j \ln(o_j)$	Vhodná pro klasifikační úlohy. (14)
IoU chybová funkce pro jeden vzor	$E_{loss} = 1 - IoU$	Vhodná pro detekci objektů v obraze. (15)

Jelikož chybová funkce určuje míru odlišnosti výstupu od správného výstupu na výstupní vrstvě, je nutné určit míru odlišnosti i na jednotlivých neuronech na vrstvách skrytých. Tomuto procesu se říká zpětné šíření chyby („*backpropagation*“) [20,21]. Algoritmus zpětného šíření chyby lze rozdělit do tří částí. V první části proběhne průchod vstupu, respektive vstupních dat, neuronovou sítí a je určena hodnota chybové funkce na výstupní vrstvě. V druhém kroku je chyba, respektive hodnota gradientu chybové funkce, zpětně šířena přes váhy po vrstvách zpět směrem ke vstupu. Ve třetím kroku dojde k upravení vah podle optimalizačního algoritmu. Tímto způsobem jsou upraveny trénovatelné parametry celé sítě tak, aby minimalizovali hodnotu chybové funkce [20,21]. Tento proces se obvykle opakuje mnohokrát, tedy trénovací

množina je předkládána trénovacímu algoritmu vícenásobně. Jednomu kroku průchodu trénovací množiny dat se obvykle říká epocha [20,21].

Optimalizačních algoritmus pro úpravu parametrů sítě má mnoho variant. Základním algoritmem je **gradientní sestup** („*gradient descent*“). Gradientní sestup funguje na principu určení gradientu chybové funkce pro celou trénovací množinu dat, jeho zprůměrování a následnou úpravou trénovatelných parametrů sítě proti směru určeného gradientu za účelem posunutí se ve směru lokálního minima chybové funkce [20,22]. Úprava parametrů sítě je následující:

$$w = w - \eta \cdot \frac{\partial E_{loss}}{\partial w} \quad (16)$$

kde w je parametr sítě, η je krok učení („*learning rate*“) a E_{loss} je hodnota chybové funkce na trénovací množině. Výhodou gradientního sestupu je, že je poměrně snadno implementovatelný. Nicméně velkou nevýhodou je velká výpočetní náročnost při procesu minimalizace chybové funkce, zvláště při velkém množství trénovacích dat. Možným vylepšením této metody při úpravě trénovatelných parametrů je použit člen, který zohledňuje setrvačnost. Úprava vah není pak závislá pouze na aktuálním gradientu, ale i na gradientu v předchozích krocích (epochách) [20,22]. Úpravou metody gradientního sestupu je **stochastický gradientní sestup** (SGD). Na rozdíl od gradientního sestupu SGD provádí změnu parametrů sítě po každém vzoru nebo podmnožině vzorů z množiny trénovacích dat. Pro upřesnění SGD provádí úpravu parametrů sítě po každém vzoru, ale reálně nejčastější varianta je úprava parametrů po jednotlivých podmnožinách vzorů (dávkách, „*batch*“), což je označeno jako MGD („*Mini-batch gradient descent*“), nicméně v praxi a literatuře se nejčastěji objevuje označení SGD pro úpravu po každém vzoru a i pro úpravu po dávce vzorů [20,22]. Výhodou toho přístupu je nižší výpočetní náročnost a rychlejší konvergence kvůli častější úpravě parametrů sítě během jedné epochy tréninku. Nevýhodou SGD může být určitá oscilace gradientu, která je u klasického gradientního sestupu potlačena zprůměrováním hodnoty chybové funkce celé epochy [22]. Vylepšením SGD je pak, podobně jako u gradientního sestupu, zahrnutí setrvačnosti („*momentum*“) při úpravě parametrů sítě. Zohlednění setrvačnosti vypadá následovně:

$$v_t = \alpha \cdot v_{t-1} + \eta \cdot \frac{\partial E_{loss}}{\partial w} \quad (17)$$

parametry sítě jsou pak upraveny následovně:

$$w = w - v_t \quad (18)$$

kde člen w je parametr sítě, α je setrvačnost („*momentum*“), η je krok učení a E_{loss} je hodnota chybové funkce [20,22]. Velice často používaným optimalizačním algoritmem, a to především v kontextu hlubokého učení, je algoritmus **Adam** („*Adaptive moment estimation*“) [20]. Tento algoritmus využívá princip setrvačnosti v kombinaci s adaptivním krokem učení pro jednotlivé parametry úpravou gradientu [23]. Adam se řídí následujícími vztahy:

$$g_t = \frac{\partial E_{loss}}{\partial w_{t-1}} + \lambda \cdot w_{t-1} \quad (19)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \cdot g_t \quad (20)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (21)$$

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1} \quad (22)$$

$$\widehat{v}_t = \frac{m_t}{1 - \beta_2} \quad (23)$$

$$w_t = w_{t-1} - \eta \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} \quad (24)$$

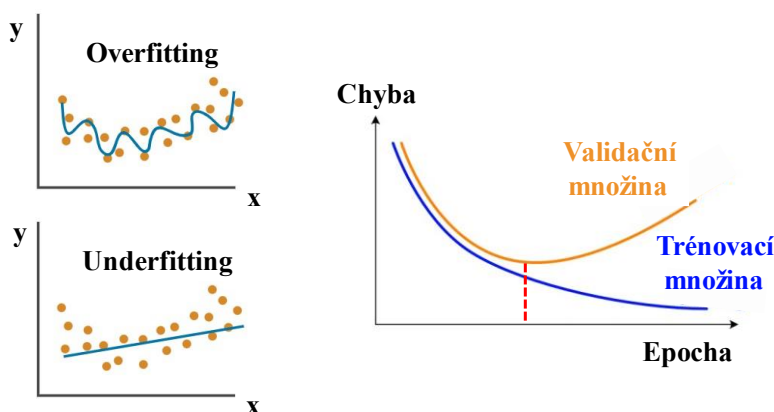
kde w je parametr sítě, η je krok učení \widehat{v}_t a \widehat{m}_t jsou členy zohledňující setrvačnost prvního a druhého řádu, λ je parametr tuhnutí vah (může být nulový), β_1 a β_2 jsou parametry řídicí pokles setrvačnosti, g_t je gradient chybové funkce a ϵ je číslo, které zabraňuje dělení nulou při inicializaci algoritmu [23].

SGD a Adam jsou reálně nejčastěji používané optimalizační algoritmy, jenž mají mnoho úprav [20]. Pro účely této práce jsou významné jen některé, jenž jsou popsány v praktické části práce.

Proces tréninku jako takový upravuje trénovatelné parametry neuronových sítí. Tento proces je však řízen mnoha dalšími parametry, jejichž hodnoty se procesem tréninku nemění. Tyto parametry se nazývají hyperparametry a patří mezi ně například počet neuronů na skryté vrstvě, počet skrytých vrstev, krok učení, typ optimalizačního algoritmu a jeho parametry apod. [20].

S procesem tréninku neuronových sítí (a i dalších algoritmů strojového učení) souvisí pojmy přeučení („*overfitting*“) a nedoučení modelu („*underfitting*“). **Nedoučení modelu** nastává v okamžiku, kdy model není dostatečně natrénovaný a nedosahuje dobrých výsledků na tréninkových, a především na dalších předložených datech [20,21]. **Přeučení** naopak nastává ve chvíli, kdy má model velmi dobré výsledky tréninkových datech, nicméně na datech, jež nebyly použity v trénovací množině nedosahuje dobrých výsledků a obecně má velmi malé zobecňovací schopnosti [20,21]. Přeučení a nedoučení modelu jsou znázorněny na obrázku 16 vlevo.

K předejití obou zmíněných jevů se často používá rozdělení trénovací množiny na dvě části. Jedna část je použita na trénink modelu, zatímco druhá je pak použita na průběžné kontrolování chybové funkce na datech, které model při tréninku nemá k dispozici. Těto množiny se říká validační. V okamžiku, kdy hodnota chybové funkce validační množiny



Obrázek 16: Přeučení a nedoučení (vlevo, převzato z [24] a upraveno), průběh chyby během tréninku (vpravo, převzato z [25] a upraveno)

přestane klesat, případně začne růst, model začíná být přetrénovaný (přeučeny). Tento fenomén je znázorněn na obrázku 16 vpravo, kde je červenou čarou naznačena oblast minima chybové funkce na validační množině. V praxi se často používá technika, kdy se dostupný dataset rozdělí do tří množin. První množina je trénovací, druhá validační a třetí testovací. Trénovací a validační množiny jsou použity stejně jako bylo popsáno, tedy na trénink a kontrolu chybové funkce. Testovací množina je pak použita na závěrečné otestování modelu na další množině, jejíž data nebyla v průběhu tréninku k dispozici, aby se předešlo výskytu možné náklonosti vůči kombinaci trénovací a validační množiny [20]. Mezi další techniky rozdělení dat patří například křížová validace („*cross validation*“), která se typicky používá ve chvíli, kdy není k dispozici dostatečné množství tréninkových dat [20].

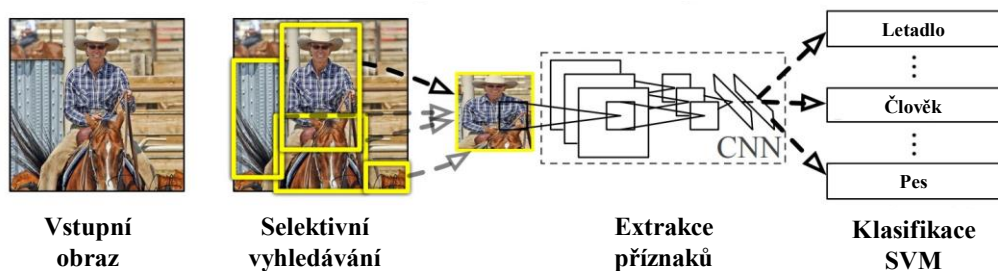
2.3.9 Architektury konvolučních neuronových sítí pro rozpoznávání objektů

Tradiční metody rozpoznávání objektů, které jsou popsány v kapitole 2.2.3, jsou v dnešní době zastíněny metodami na bázi hlubokého učení, a to konkrétně hlubokými konvolučními neuronovými sítěmi (DCNN, v následujícím textu je zkratkou CNN míněna DCNN) [8]. Na rozdíl od CNN, většina tradičních metod rozpoznávání objektů v obraze spoléhá na ručně navrhované extraktory podstatných rysů. CNN se tyto extraktory učí, a to pomocí trénování filtrů konvolučních vrstev. CNN jsou pak schopny extrahovat komplexní sémantické informace, jenž umožní vykonávat složité úlohy rozpoznávání objektů v obraze [8].

Jeden z nejjednodušších přístupů pro lokalizaci a klasifikaci objektů je metoda posouvajícího se okna. Tato metoda vychází z velice podobného principu jako je popsáný v kapitole o tradičních metodách rozpoznávání objektů v obraze s tím rozdílem, že druhý a třetí krok zmíněné metody je nahrazen CNN. Výhodou této metody je jednoduchá implementace, nicméně obrovskou nevýhodou je časová náročnost výpočtu [8].

V dnešní době se dají přístupy pro rozpoznávání objektů v obraze pomocí hlubokého učení rámcově rozdělit do dvou skupin. První skupinou jsou dvoufázové detektory („*Two-stage detectors*“) do níž patří R-CNN a jeho varianty a druhou skupinou jsou jednofázové detektory („*One-stage detectors*“) do níž patří algoritmus YOLO a jeho varianty [8]. Obecně se dá říct, že dvoufázové detektory dosahují vyšších úspěšností při testování na veřejných porovnávacích datasetech („*Public benchmarks datasets*“), o kterých je pojednáno později, nicméně toho je dosaženo na úkor časové náročnosti výpočtu. Jednofázové detektory obvykle dosahují horší úspěšnosti na zmíněných porovnávacích datasetech, ale jsou mnohem efektivnější v kontextu real-time úloh [8].

Před samotným popisem jednotlivých architektur je nutné uvést, že je celá tato kapitola směřována vůči detekci objektů v obraze, což souvisí i s analýzou problému a volbami metod jeho řešení (kapitola 3), nicméně u většiny uvedených architektur existují modifikace i pro ostatní přístupy rozpoznávání objektů v obraze (tedy pro samotnou klasifikaci, sémantickou segmentaci nebo segmentaci instancí).



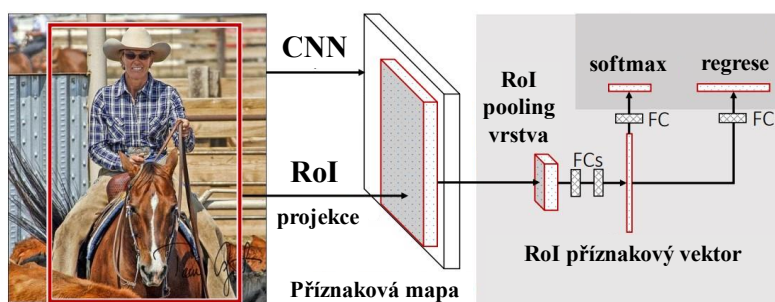
Obrázek 17: R-CNN (převzato z [26] a upraveno)

Dvoufázové detektory

Dvoufázové detektory rozdělují detekci do dvou fází. V první fázi jsou navrženy kandidátské oblasti (oblasti zájmu, případně ROI) a v druhém kroku pak dojde k určení třídy příslušnosti daného objektu (klasifikace) a lokalizaci hraničního objektu (regrese) [8]. Mezi nejtypičtější dvoufázové detektory patří R-CNN a jeho úpravy.

R-CNN (zkratka pro „Regions with CNN features“) je pionýrem skupiny dvoufázových detektorů na bázi CNN [8]. Algoritmus má ve své podstatě tři kroky. Prvním krokem je návrh oblastí, kde se mohou objekty vyskytovat („proposal generation“). Toho je docíleno pomocí selektivního vyhledávání („selective search“). Selektivní vyhledávání vybírá ROI pomocí porovnávání podobnostních metrik na základě barvy, textury, velikosti a tvaru a je navrženo tak, aby přímo nezavrhovalo oblasti, které se dají snadno označit jako pozadí v obrazu. Podobné oblasti jsou iterativně zkombinovány, čímž dojde k získání objektů, kterých bývá obvykle okolo 2000. Selektivní vyhledávání je naznačeno na obrázku 17. V druhém kroku jsou pomocí CNN extrahovány podstatné rysy a příznaky použitím VGG-16, což je CNN s 13 konvolučními vrstvami a 3 plně propojenými vrstvami na konci. Extrahované příznaky vstupují do třetího kroku, což je klasifikátor na bázi SVM typu jedna na ostatní (obrázek 9 vpravo) [8,26]. Extrahované rysy vstupují ještě do regrese za účelem zpřesnění hraničního obdélníku. Architektura R-CNN je znázorněna na obrázku 17. Velkým nedostatkem R-CNN je to, že algoritmus selektivního vyhledávání spoléhá na nízko-úrovňové sémantické znaky (např. hrany, textury apod.), které mohou být v kontextu komplexních objektů zavádějící. Dalším nedostatkem R-CNN je obrovská časová náročnost výpočtu, protože každá ROI vstupuje do CNN pro extrahování příznakové mapy [8].

Fast R-CNN na rozdíl od R-CNN neprovádí průchod CNN pro každou kandidátskou oblast určenou pomocí selektivního vyhledávání, ale do CNN vstupuje celý vstupní obraz. Použitá CNN je VGG-16 stejně jako u R-CNN. Pomocí CNN jsou extrahovány podstatné rysy a je získána příznaková mapa, která je zkombinována s kandidátskými oblastmi, stejně jako u

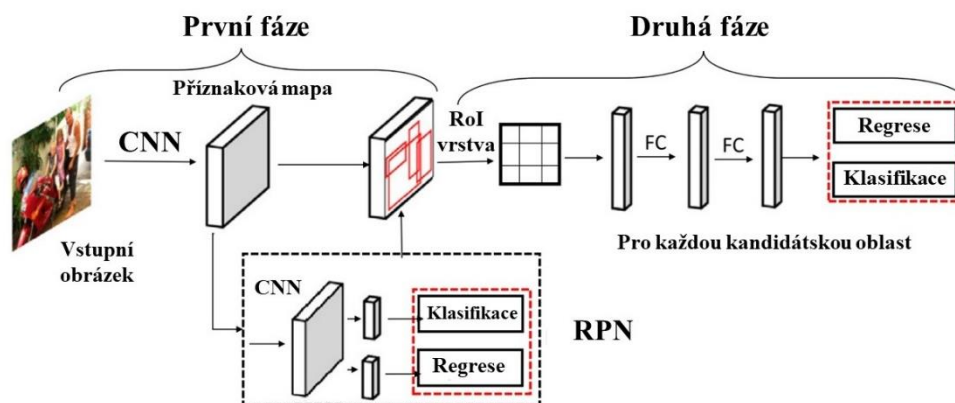


Obrázek 18: Fast R-CNN (převzato z [27] a upraveno)

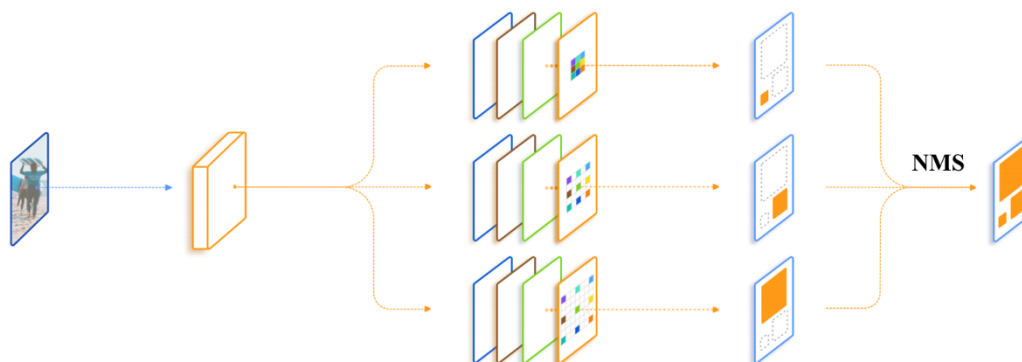
R-CNN, určenými pomocí selektivního vyhledávání. Tato kombinace je pak pomocí RoI poolingové vrstvy zmenšena na malou příznakovou oblast o velikost $N \times N$ (v článku [27] uvedena 7×7), která vstupuje do dvou plně propojených NN, jedna zodpovídající za klasifikaci objektu a druhá za regresi souřadnic hraničního objektu [8,27]. Architektura Fast R-CNN je uvedena na obrázku 18. Fast R-CNN je oproti R-CNN rychlejší, nicméně stále používá algoritmus selektivního vyhledávání, který proces značně zpomaluje [8].

Faster R-CNN je detektor, v němž již není použit algoritmus selektivního vyhledávání. Selektivní vyhledávání je nahrazeno konvoluční neuronovou sítí, jenž navrhuje oblasti výskytu objektů (RPN, zkratka pro „Region Proposal Network“) [8,28]. Do RPN vstupuje obrázek jakékoliv velikosti a na výstupu je pak množina čtvercových kandidátských oblastí. V praxi RPN prochází příznakovou mapu ze vstupní CNN (VGG-16) pomocí posouvajícího se okna o velikosti $N \times N$, což vytváří vektor příznaků pro každou oblast příznakové mapy. V každé pozici posouvajícího se okna je v jednu chvíli predikováno několik podoblastí. Predikované podoblasti jsou podřízeny referenčním hraničním obdélníkům, kterým se také říká kotvy („anchors“). Každá kotva má střed ve středu posouvajícího se okna, je dána měřítkem a poměrem stran a slouží k definování tvaru kandidátského objektu [28]. Kotvy jsou často ručně a heuristicky navrhovány, což může vnášet do algoritmu určitá omezení [8]. Na výstupní vrstvě RPN je dvojice plně propojených vrstev. Jedna je vrstva klasifikační určující příslušnost navržené oblasti mezi objekty a pozadí a druhá je regresní vrstva, která zpřesňuje hraniční objekt. Výstup z RPN se pak kombinuje s příznakovou mapou určenou pomocí vstupní CNN a prochází RoI pooling vrstvou do další plně propojené NN na jejímž výstupu je již predikovaná třída objektu s příslušným hraničním obdélníkem [8,28]. Architektura Faster R-CNN je uvedena na obrázku 19. Velkou výhodou Faster R-CNN oproti předchozím uvedeným variantám R-CNN je to, že nevychází z algoritmu selektivního vyhledávání a také to, že RPN sdílí konvoluční vrstvy s CNN detektorem a je díky tomu možné celou takto vzniklou strukturu optimalizovat jako celek, což je při použití selektivního vyhledávání obtížné [8].

TridentNet je dvoufázový detektor, jenž přímo nespadá do modifikací R-CNN jako předchozí uvedené a je popsán v článku [32]. Hlavní myšlenkou jeho autorů bylo vytvořit detektor invariantní vůči různým velikostem objektů. Toho je docíleno pomocí rozdělení CNN (použitá CNN je ResNet-101) do tří paralelních větví, jenž spolu sdílejí váhy, ale filtry jednotlivých konvolučních vrstev mají odlišný parametr dilatace (*dilatation*). Rozdělení do paralelních větví je znázorněno na obrázku 20, kde je v jednotlivých větvích naznačen barevně parametr dilatace, při jehož zvětšení dochází k rozšíření receptivního pole, což znamená, že



Obrázek 19: Faster R-CNN (převzato z [8] a upraveno)



Obrázek 20: Schéma principu detektoru TridentNet a znázornění parametru dilatace (převzato z [32] a upraveno)

filtry zohledňují při konvoluci větší oblast pixelů snímku. Každá z paralelních větví se zaměřuje na detekci objektů o jiné velikosti a výstupy z nich jsou na konci zkombinovány pomocí techniky Non-max suppression (NMS) [32]. NMS vybere z množiny hraničních obdélníků, které se navzájem překrývají o více než je stanovený práh parametru IoU, pouze ten, jenž má největší konfidenci [8].

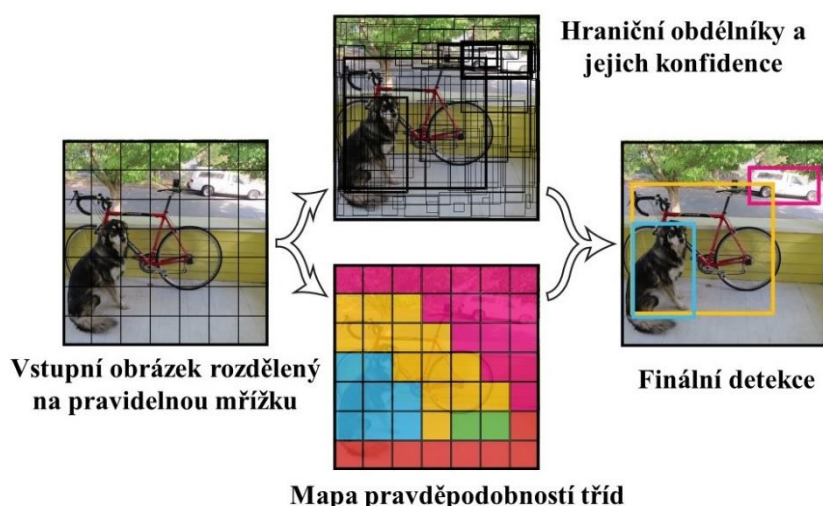
Jednofázové detektory

Na rozdíl od dvoufázových detektorů, jednofázové detektory obvykle určují všechny oblasti v obraze jako potencionální objekty a snaží se je rozdělit mezi skutečné objekty a pozadí. Velmi významnou rodinou jednofázových detektorů je rodina algoritmu YOLO [8]. V následujícím textu je tento algoritmus důkladně popsán a dále jsou uvedeny a popsány jeho stěžejní varianty.

YOLO („*You only look once*“) je algoritmus, který provádí detekci pouze v jednom průchodu vstupního obrazu algoritmem s použitím jedné CNN [29]. K predikci hraničních obdélníků používá příznaky extrahované z celého obrázku a provádí detekci všech hraničních obdélníků všech tříd současně. Díky tomu je možné celý algoritmus (model) trénovat a optimalizovat jako celek [29].

Princip spočívá v tom, že vstupní obrázek je rozdělen na $S \times S$ mřížku. Za předpokladu, že střed objektu spadá do určité buňky této mřížky, je tato buňka zodpovědná za jeho detekci. Každá buňka pak detekuje B hraničních obdélníků a jejich konfidenční skóre, které vyjadřuje míru jistoty modelu v tom, že se v buňce vyskytuje objekt a dále s jakou přesností je určen jeho hraniční obdélník. Tato konfidence je definovaná jako $\Pr(\text{Object}) \cdot \text{IoU}$. Každý hraniční obdélník obsahuje 5 predikcí. První dvě jsou x a y , jenž popisují střed hraničního obdélníku vůči buňce mřížky, další dvě jsou w a h , jenž reprezentují šířku a výšku hraničního obdélníku vůči rozměrům celého obrázku a nakonec konfidenci. Každá buňka ještě obsahuje C podmíněných pravděpodobností vyjadřujících pravděpodobnost příslušnosti objektu do dané třídy za předpokladu, že se jedná o objekt, čímž vznikne pravděpodobnostní mapa celé mřížky. Výstupem ze sítě je pak tenzor o velikosti $S \times S \times (B \cdot 5 + C)$ (V původním článku $S=7$, $B=2$ a $C=20$) [29]. Rozdělení obrázku na $S \times S$ mřížku a princip detekce je znázorněn na obrázku 21.

YOLO používá architekturu inspirovanou GoogLeNet CNN. Konkrétněji, CNN s 24 konvolučními vrstvami následovanými 2 plně propojenými vrstvami [29]. Tato architektura je znázorněna v [29].

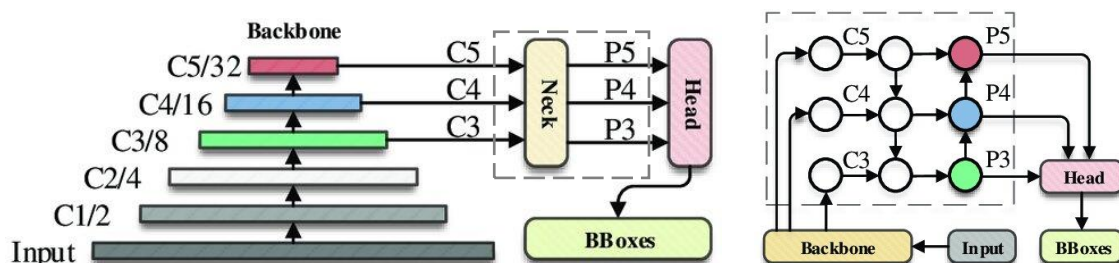


Obrázek 21: Princip metody YOLO (převzato z [29] a upraveno)

Mezi hlavní omezení algoritmu YOLO patří skutečnost, že každá buňka mřížky může predikovat pouze jednu třídu objektu, což může být nevýhodné v kontextu malých objektů vyskytujících se blízko u sebe. Dalším omezením modelu, jenž popisují autoři v původním článku je detekování objektů s netypickými poměry výšky a šířky [29].

YOLOv5 (5. verze algoritmu) vychází ze stejného základu jako YOLOv1 (YOLO), do nějž integruje efektivní techniky použité v jeho předchozích verzích (verze 2 až verze 4) a tím pádem je oproti YOLOv1 výrazně komplexnější a také efektivnější [31].

Architektura modelu YOLOv5 by se dala rozdělit do tří částí a její zjednodušené schéma je znázorněno na obrázku 22. Úplné schéma architektury je dostupné v [30]. První částí je tzv. páteř („backbone“), což je CNN, jenž zodpovídá za extrakci příznaků a příznakové mapy ze vstupních snímků. Jako páteř algoritmu je použita CNN Darknet53, respektive CSP-Darknet53. CSP („Cross-stage-Partial-connections“, v úplné architektuře je označen jako modul C3¹) je označení pro modul integrovaný do architektury Darknet53, jenž slouží k extrakci příznaků a k potlačení výskytu mizejícího nebo explodujícího gradientu, jenž se typicky může vyskytovat u velmi hlubokých neuronových sítí. Na konci CNN je umístěn modul SPPF („Spatial pyramid pooling fusion“) pro zvýraznění podstatných příznaků [33]. Druhou částí je krk („neck“), jenž zodpovídá za propojení páteře, tedy extraktoru příznaků, a hlavy modelu („head“), jenž zodpovídá umístění hraničních obdélníků. Pomocí krku algoritmu jsou pro detekci využity víceúrovňové příznaky extrahované v různých hloubkách CNN, ze kterých jsou před vstupem



Obrázek 22: Zjednodušené schéma architektury YOLOv5 (vlevo) a detail krku algoritmu YOLOv5 (vpravo). Oboje převzato z [33].

¹ Na obrázku 22 se vyskytuje označení C3, avšak to značí příznakovou mapu na výstupu z vrstvy CNN, ne modul C3 v úplné architektuře algoritmu YOLOv5.

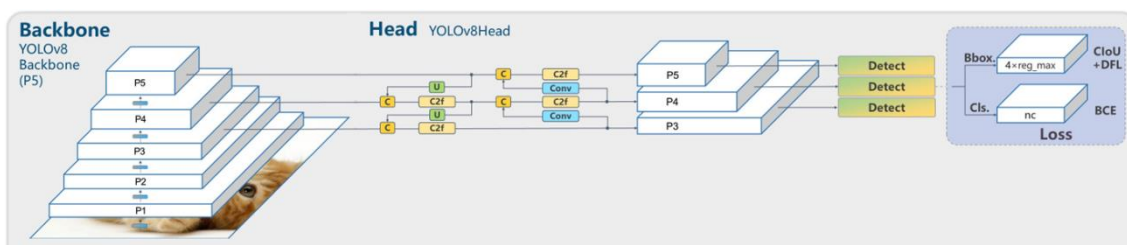
do hlavy algoritmu vytvořeny tři příznakové mapy různých velikostí (v původní implementaci 80×80 , 40×40 a 20×20) pro detekci malých, středních a velkých objektů, a tedy pro vytvoření určité invariance vůči různým velikostem objektů. Pro kombinaci víceúrovňových příznaků používá krk moduly FPN („*Feature pyramid network*“, na obrázku 22 vlevo prostřední blok, do nějž vstupují příznakové mapy C5, C4 a C3, který je propojen ze shora dolů, tedy v proti směru redukce dimenze dat konvolučními a poolingovými vrstvami) a PAN („*Pyramid attention network*“, na obrázku 22 vlevo blok označený barvami z nějž vystupují příznakové mapy P5 až P3). Oba bloky slouží ke komplexnějšímu zkombinování příznaků do výstupních příznakových map před vstupem do hlavy modelu. V samotné hlavě algoritmu pak dojde k určení souřadnic hraničních obdélníků, tříd objektů a konfidence detekcí [33].

YOLOv5 používá na rozdíl od YOLOv1 k predikci hraničních obdélníků kotvy. V každé buňce mřížky je definováno několik kotev, které slouží jako prototypy hraničních obdélníků detekovaných objektů. Jedná se o velice podobný princip, který byl popsán v algoritmu Faster R-CNN. V YOLOv5 je navíc použit AutoAnchor algoritmus, který upravuje před tréninkem tvar kotev, tak aby dobře vystihovaly typické tvary hraničních obdélníků v trénovací množině [32]. Pro potlačení vícero hraničních obdélníků na jednom objektu, které se od sebe téměř neliší používají algoritmy z rodiny YOLO metodu Non-max suppression (NMS) [33].

YOLOv8 je v době vypracování této práce považován za SOTA („*state-of-the-art*“) algoritmus mezi jednofázovými detektory [34]. Koncepce jeho architektury je velice podobná jako u algoritmu YOLOv5 s určitými rozdíly [31]. Zjednodušené schéma architektury YOLOv8 je uvedeno na obrázku 23. Úplné schéma jeho architektury je uvedeno v přílohách práce a je převzato z [35].

V páteři algoritmu, tedy v CNN pro extrakci příznaků jsou na vstupní vrstvě změněny velikosti konvolučních jader (filtrů) z 6×6 u YOLOv5 na 3×3 u YOLOv8 a modul C3 je nahrazen modulem C2f, jenž slouží k podobnému účelu, ale je efektivnější v procesu tréninku [31,35]. V krku algoritmu jsou odebrány dvě konvoluční vrstvy a v YOLOv8 je oproti YOLOv5 méně vzájemných propojení, což tvoří krk algoritmu kompaktnější, nicméně používá stejné moduly jako YOLOv5 [31,35]. Největší změnou oproti YOLOv5 prošla hlava algoritmu. YOLOv8 podobně jako YOLOv1 nepoužívá při detekci kotvy, a tedy predikuje přímo souřadnice hraničních obdélníku, což určitým způsobem zjednodušuje detekci a snižuje komplexnost algoritmu [31].

Obecně lze říct, že architektura YOLOv8 je do jisté míry jednodušší než architektura YOLOv5, což může vést k efektivnější optimalizace algoritmu jako celku.



Obrázek 23: Zjednodušené schéma architektury YOLOv8 (převzato z [35] a upraveno)

Porovnání popsaných architektur

V tabulce 3 je uvedeno porovnání algoritmů popsaných v této podkapitole. Algoritmy jsou srovnány na dvou veřejných testovacích datasetech, což jsou velice rozsáhlé a robustní datasety obsahující širokou škálu objektů různých velikostí [8]. Mezi používané veřejné testovací datasety patří dataset Pascal VOC a MSCOCO (dále jen COCO) [8]. Pascal VOC 2007 a jeho pozdější varianta VOC 2012 je dataset obsahující 2 až 3 objekty na jednom obrázku a celkově 20 tříd rozpoznávaných objektů. Objekty v tomto datasetu jsou obvykle střední velikosti relativně vůči velikosti obrázku [8]. Dalším datasetem je COCO (zkratka pro „*Common objects in context*“). COCO je velice obecný dataset od společnosti Microsoft obsahující přibližně 80 tříd objektů a až 10 objektů v jednom obrázku. Jeho obrázky často obsahují malé objekty, což bývá typicky pro detektory obtížné [8].

Hlavní porovnávací metrika algoritmů je mAP („*Mean average precision*“). Tato metrika představuje souhrnné ohodnocení jejich výkonosti a vyjadřuje průměrnou hodnotu metriky AP („*Average precision*“) napříč detekovanými třídami objektů [8]. Metrika AP je vypočtena pro každou třídu zvlášť a to tak, že je seřazeno N nejlepších detekcí algoritmu podle jejich konfidence v sestupném pořadí, v tomto pořadí jsou určeny přesnosti (P) a citlivosti (R) a jsou vyneseny do závislosti P-R (P-R křivka, „*Precision-recall curve*“). Plocha pod P-R křivkou je pak hodnota AP [8]. Obě metriky (AP, mAP) nabývají hodnot z intervalu $\langle 0; 1 \rangle$ (0 % až 100 %). P-R křivky jsou znázorněny v praktické části práce např. na obrázku 45 vpravo.

V tabulce 3 jsou uvedeny dva parametry mAP. Prvním je mAP50, u kterého je pro vyhodnocení, zda je objekt TP nebo FP použita hodnota prahu IoU jako 0.5 (50 %). Druhým parametrem je mAP50-95, jenž vznikne posunováním prahu IoU z 0.5 na 0.95. Parametr mAP50-95 vyjadřuje konzervativnější přístup k vyhodnocení hraničního obdélníku, protože zpříšňuje kritérium pro jeho detekování, tedy minimální práh parametru IoU. V literatuře se lze setkat i s vyjádřením prahové hodnoty IoU uvedené v dolním indexu metriky mAP, nicméně rámci této práce je použitý zápis bez indexu (viz. předchozí text).

Tabulka 3: Porovnání modelů na základě mAP

Model	Dataset	Metrika	Výsledek [%]	Zdroj
R-CNN	VOC 2007	mAP50	66.0 %	[8]
	VOC 2012	mAP50	62.4 %	[8]
Fast R-CNN	VOC 2007	mAP50	70.0 %	[8]
	VOC 2012	mAP50	68.4 %	[8]
	COCO	mAP50	35.9 %	[8]
	COCO	mAP50-95	19.7 %	[8]
Faster R-CNN	VOC 2007	mAP50	73.4 %	[8]
	VOC 2012	mAP50	70.4 %	[8]
	COCO	mAP50	42.7 %	[8]
	COCO	mAP50-95	21.9 %	[8]
TridentNet	COCO	mAP50-95	48.4 %	[32]
YOLOv1	VOC 2007	mAP50	63.4 %	[8]
	VOC 2012	mAP50	57.9 %	[8]
YOLOv5	COCO	mAP50-95	49.0 %	[36]
YOLOv8	COCO	mAP50-95	52.9 %	[37]

Z tabulky 3 je patrné, že v rámci testovacího datasetu VOC 2012 si nejlépe vedl algoritmus Faster R-CNN, nicméně toto porovnání není z hlediska všech uvedených algoritmů, jelikož ostatní algoritmy jsou z pohledu doby vydání novější a jsou porovnány pouze na datasetu COCO. Z porovnání na datasetu COCO je patrné, že si nejlépe vedl algoritmus YOLOv8. K tomuto algoritmu, a i k algoritmu YOLOv5, je nutné doplnit, že se u obou jedná o druhou největší dostupnou variantu algoritmu (varianta *Large*). O vlivu velikosti algoritmů je pojednáno dále v práci, nicméně pro uvedené porovnání jsou tyto varianty dostačující.

Z hlediska porovnání rychlosti detekce modelu jsou jednofázové detektory (rodina YOLO) mnohem efektivnější a často se uplatňují v aplikacích, kde je požadována real-time odezva modelu na vstupní snímky [8]. V tabulce 4 jsou uvedeny příklady rychlosti odezvy pro model YOLOv5 a YOLOv8, respektive jejich nejmenší a druhé nejmenší varianty. Parametr odezva v tabulce 4 je průměrná rychlost průchodu snímku z datasetu COCO.

Tabulka 4: Rychlost odezvy modelu YOLOv5 a YOLOv8

Model	Varianta modelu	Odezva [ms]	Výpočetní prostředí	Zdroj
YOLOv5	<i>Nano</i>	45	Amazon EC2 P3 Instances	[36]
	<i>Large</i>	430		
YOLOv8	<i>Nano</i>	80	Amazon EC2 P4 Instances	[37]
	<i>large</i>	375		

3 Analýza problému a volba metod pro jeho řešení

Z teoretické části práce je patrné, že rozpoznávání objektů v obraze se dělí na několik přístupů (viz. kapitola 2.2.2). Jelikož cílem práce není pouze rozpoznat typ měřeného vzorku (typická klasifikační úloha), ale i lokalizovat jeho polohu v obraze, vhodnými přístupy takto stanovaných parametrů úlohy se jeví detekce objektů a segmentace objektů (segmentace instancí). Nevýhodou detekce objektů by mohla být skutečnost, že predikovaný hraniční objekt je obdélník, který přímo nenese informaci o skutečném tvaru objektu. Nicméně s ohledem na charakter detekovaných těles, jenž je popsán v následující kapitole, a s ohledem na typické nastavení měřicí soustavy při testování materiálů s využitím DIC, a to především orientace kamery vůči vzorku, kdy je vzorek v poli kamery zarovnán horizontálně nebo vertikálně, je vyhodnocen přístup detekce jako dostatečný [67]. Navíc, při použití segmentace instancí, kdy je predikovaným hraničním objektem pixelová maska, je proces sestavování vhodného datasetu velmi pracný a dataset musí být velice obsáhlý, což by mohl být problém s ohledem na možnosti dostupné databáze, které jsou popsány v následující kapitole. V rámci této práce je tedy použit přístup detekce objektů v obraze, jenž umožňuje dostatečnou přesnost lokalizace a klasifikaci typů objektů.

Strojové učení nabízí mnoho algoritmů pro detekci objektů v obraze. Jak již bylo zmíněno, v dnešní době se za tímto účelem často používají hluboké konvoluční neuronové sítě. V teoretické části práce je pojednáno o několika architekturách těchto sítí a jsou porovnány na veřejných testovacích datasetech. Za účelem splnění cílů této práce je jako nejvhodnější algoritmus zvolen algoritmus YOLOv8. Jeden z hlavních důvodů této volby je, že YOLOv8 je v době vypracování této práce považován za SOTA („state-of-the-art“) algoritmus v oblasti detekce objektů [34]. Dále, jelikož se jedná o jednofázový detektor, je vhodný pro real-time použití, jenž je specifikováno v zadání práce a z uvedených algoritmů dosahoval nejlepších výsledků na datasetu COCO. Další velkou výhodou algoritmu YOLOv8 je také to, že je implementován v jazyce Python společností Ultralytics ve stejnojmenné knihovně, která je veřejně dostupná a nabízí mnoho funkcionalit potřebných pro strojové učení.

3.1 Nástroje použité pro implementaci

K anotování snímků byl použit anotační nástroj Roboflow, jenž poskytuje mnoho funkcionalit pro tento úkon. Proces anotace je popsán v podkapitole 4.9.

Veškerá implementační činnost byla provedena v programovacím jazyce Python. Použité knihovny byly: *os* pro manipulaci s daty (snímky), *PIL* pro úpravu snímků, již zmíněná knihovna *ultralytics*, knihovna *PyTorch* poskytující další funkcionality potřebné při strojovém učení a dále knihovna *RayTune*, jenž poskytuje funkcionality potřebné pro optimalizaci algoritmů strojového učení.

Implementační činnost probíhala na osobním počítači Lenovo typ IdeaPad 330. Trénování probíhalo primárně na počítači s grafickou kartou RTX A4000, jenž byl poskytnut vedoucím práce, a sekundárně v prostředí Google Colaboratory.

4 Tvorba datasetu

Tato kapitola se zabývá tvorbou datasetu pro implementaci zvoleného algoritmu. Je v ní popsána práce s databází společnosti X-Sight, rozšíření datasetu o vlastní záznamy a princip tvorby datasetu jako takový.

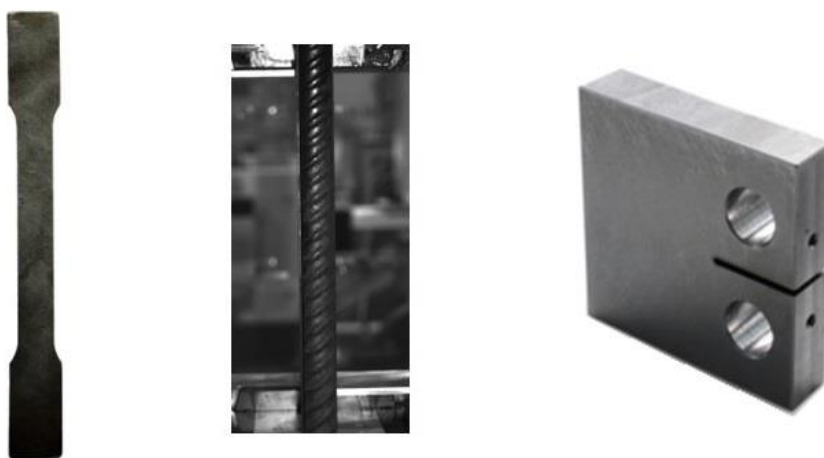
V kontextu strojového učení, konkrétně učení neuronových sítí, je práce s daty a vytvoření vhodného datasetu stěžejním krokem. Tato skutečnost je v práci mnohokrát naznačena, nicméně je vhodné ji uvést i úvodem této kapitoly.

4.1 Databáze X-Sight

Za účelem vypracování této práce byl společností X-Sight poskytnut přístup do jejich databáze snímků pořízených ze standartních DIC měření. Společnost X-Sight se zabývá vývojem optických extenzometrů a pokročilých DIC systému na měření deformací [66].

V databázi společnosti X-Sight (v textu často označována jako dostupná databáze) bylo v době vypracování této práce obsaženo několik set různorodých měření s různými zkušebními tělesy (vzorky). Jak již bylo zmíněno, při implementaci algoritmů na bázi CNN je obecně nutno disponovat velkým množstvím dat k vytvoření modelu s dostatečnými zobecňovacími schopnostmi. Na základě tohoto předpokladu byla poskytnutá databáze důkladně prozkoumána a bylo vyhodnoceno, že obsahuje dostatečné množství dat pro vytvoření dvou tříd objektů. První třídou objektů byly tahové vzorky (obrázek 24 vlevo). Druhou třídou objektů byla betonářská výztuž, respektive její jednotlivé pruty (někdy též označovány jako roxory). Prut betonářské výztuže je znázorněn na obrázku 24 uprostřed. Třetí třídou objektů, jež byla vytvořena, byla třída objektů označující vrubovaná tělesa používaná při měření lomových vlastností. Mezi tato tělesa patří například těleso CT (obrázek 24 vpravo). V době vypracování této práce databáze X-Sight neobsahovala dostatečné množství dat pro vytvoření této třídy, nicméně tento problém byl vyřešen pomocí rozšíření dostupné databáze (popsáno v podkapitole 4.5).

V tabulce 5 jsou uvedeny počty jednotlivých měření, respektive kamerových záznamů, provedených u jednotlivých typů zkušebních těles, z nichž jsou vytvořeny třídy detekovaných objektů.



Obrázek 24: Ukázka objektů definovaných tříd (vlevo tahový vzorek, uprostřed prut betonářské výztuže a vpravo CT těleso převzato z [39] a upraveno)

Tabulka 5: Počet záznamů a snímků definovaných tříd v dostupné databázi

Třída objekt	Počet záznamů	Celkový počet snímků všech záznamů
Tahový vzorek	130	119 193
Betonářská výztuž	28	54 996
Vrubovaná tělesa	15	98 211

4.2 Doporučení autorů YOLOv8

Dle doporučení autorů algoritmu YOLOv5 je vhodný počet snímků pro sestavení jedné třídy alespoň 1500, kde každý snímek obsahuje alespoň jednu instanci objektu třídy [38]. Toto doporučení lze brát v potaz i při použití algoritmu YOLOv8, jenž je implementován stejnými autory a jeho dokumentace se odkazuje na dokumentaci pro algoritmus YOLOv5.

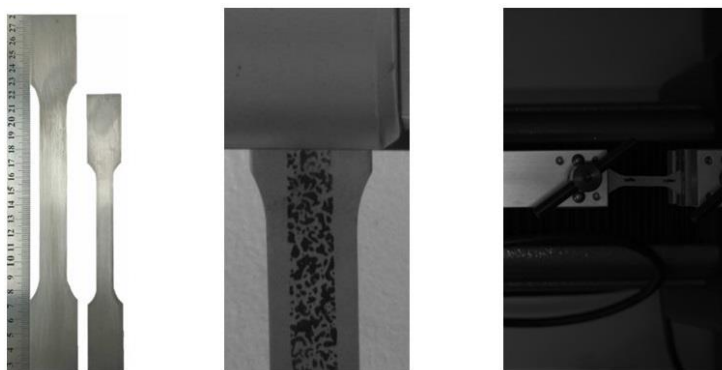
Jelikož YOLOv8 je univerzální algoritmus, který lze použít pro detekci široké škály objektů, dá se předpokládat, že autoři uvádí daná doporučení konzervativně. Předmětem této práce je detekovat zkušební tělesa, o kterých lze konstatovat, že nejsou ze sémantického hlediska tak složitá jako jiné objekty, na něž lze algoritmus YOLOv8 použít. Z toho důvodu nejsou brány doporučení autorů naprosto striktně.

4.3 Tvorba třídy tahových vzorků

První vytvořenou třídou objektů byla třída tahových vzorků, jenž byla v rámci implementace označena DogBone na základě typického tvaru zkušební tělesa připomínajícího psí kost („Dog bone“).

Základní předpoklady

Základní předpoklad pro tuto třídu byl stanoven tak, že se na snímku vstupujícím do detekčního algoritmu vyskytuje pouze jeden objekt (instance) této třídy. Tento předpoklad se opírá o charakter DIC měření, při kterém je cílem vyhodnotit deformaci na povrchu tělesa (viz. podkapitola 2.1) a to se typicky v oblasti testování materiálů provádí pouze pro jedno zkušební těleso v poli kamery pořizující záznam. Na obrázku 25 je znázorněna trojice snímků, kde prostřední a pravý snímek splňují vznesený předpoklad a levý snímek vznesený předpoklad nespĺňuje. Další předpoklady stanovené pro tuto třídu a obecně pro všechny třídy v této práci přímo vychází z předpokladů použití DIC, jenž jsou popsány v podkapitole 2.1.



Obrázek 25: Stanovené předpoklady na snímky třídy tahových vzorků a naznačení kritéria viditelnosti

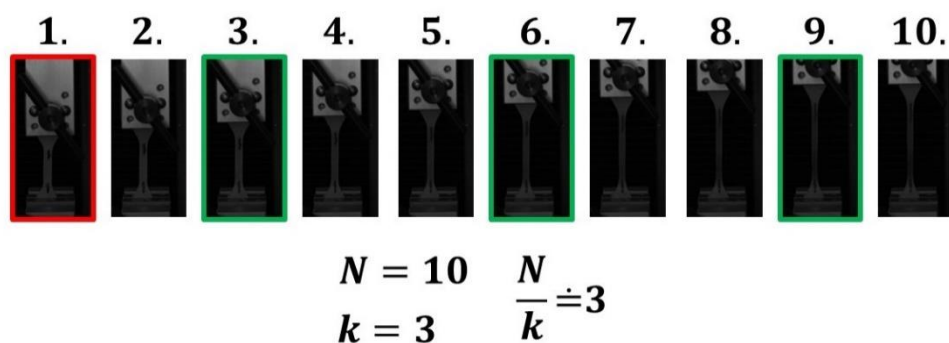
Snímky z databáze X-Sight

V tabulce 5 je uvedeno množství jednotlivých měření, respektive kamerových záznamů, z dostupné databáze. V rámci vytvoření dostatečně obecného základu této třídy byly jednotlivé záznamy z databáze ohodnoceny na základě dvou kritérií. První kritérium spočívalo v charakteru záznamu z hlediska, zda je vzorek v poli kamery malý nebo velký. Druhé kritérium spočívalo ve viditelnosti vzorku v poli kamery. Na základě těchto dvou kritérií bylo vyhodnoceno množství snímků promítnutých z jednotlivých záznamů do datasetu. V tabulce 6 jsou uvedena obě kritéria a množství snímků promítnutých v rámci jednoho záznamu do datasetu.

Tabulka 6: Kritéria pro třídu tahových vzorků

Malý vzorek	Viditelnost	Snímků na záznam
Ano	Dobrá	12
Ano	Dostačující	11
Ano	Špatná	9
Ne	Dobrá	8
Ne	Dostačující	7
Ne	Špatná	6

Smysl prvního stanoveného kritéria vycházel z toho, že většina detekčních algoritmů má problém rozpoznávat malé objekty [8]. Tento problém lze řešit tím, že se do detekčního algoritmu vstupuje se snímky o velkém rozlišení, nicméně velkou nevýhodou tohoto přístupu je, že značně roste výpočetní náročnost a tím pádem i celkový trénovací čas. Z tohoto důvodu byly jednotlivé záznamy ohodnoceny z hlediska velikosti vzorku v záznamu a záznamy, kde byl vzorek vyhodnocen jako malý, byly promítnuty do datasetu s větší váhou. Druhé kritérium souviselo se skutečností, že v mnoha záznamech vzorek přesahoval zorné pole kamery. Smyslem tohoto kritéria bylo do datasetu promítnout s větší váhou záznamy, kde se vyskytoval vzorek celý z hlediska viditelnosti. Nicméně k docílení co nejlepších zobecňovacích schopností algoritmu byly v rámci datasetu promítnuty i snímky, kde vzorky nebyly vidět celé. Příklad snímku, kde byl vzorek vyhodnocen jako malý je na obrázku 25 vpravo. Na stejném obrázku uprostřed je znázorněn snímek, kde je viditelnost vzorku vyhodnocena jako špatná, protože je vidět pouze jeho část.



Obrázek 26: Princip promítnutí snímků do datasetu (červeně a zeleně jsou označeny promítnuté snímky)

Princip promítnutí snímků ze záznamů do datasetu spočíval v tom, že záznam, jenž tvořilo N snímků a mělo z něj být promítnuto k snímků na základě stanovených kritérií byl rozdělen rovnoměrně na N/k násobky. Nultý (první snímek záznamu) a každý snímek na pozici N/k násobku byl pak promítnut do datasetu. Například ze záznamu o 800 snímcích z nějž mělo být promítnuto snímků 8, byl promítnut snímek na pozici první, sté, dvousté apod, až na pozici sedmistou. Takto navržený princip nepromítá do datasetu poslední snímek u jednotlivých záznamů z toho důvodu, že vzorek na něm byl často velmi deformovaný nebo porušený. Popsaný princip je znázorněn na obrázku 26. V případě, že by byl zvolen jednodušší princip promítnutí prvních n snímků ze záznamu na základě stanovených kritérií, hrozilo by riziko, že snímky budou z vizuálního hlediska téměř totožné, což platí především pro záznamy s velkou snímkovou frekvencí nebo pomalým průběhem zkoušky. Nicméně je stále nutné brát v potaz, že snímky se i přes zvolený princip výběru obvykle lišily pouze v pohybu pohyblivých částí zkušebního zařízení a deformací vzorku v průběhu zkoušky, která nemusí být výrazná (platí především pro kovy).

Tímto procesem bylo z dostupné databáze promítnuto do výsledného datasetu celkem 945 snímků třídy tahových vzorků.

Rozšíření třídy tahových vzorků o vlastní záznamy

Dataset byl dále rozšířen o snímky tahových vzorků, které byly za účelem vypracování této práce pořízeny. Celkem bylo pořízeno 67 snímků z 11 různých tahových vzorků v laboratoři Ústavu mechaniky těles FSI VUT. Jednotlivé snímky stejného vzorku se od sebe lišily expozičním časem a vzdáleností kamery od vzorku upnutém v čelistech zkušebního stroje. Různé snímky jednoho vzorku jsou znázorněny na obrázku 27. Pro pořízení těchto snímků byla použita kamera FLIR Backfly S o rozlišení 8.9 Mpx a objektiv Nikon 60 mm F2.8 AF MICRO-NIKKOR.



Obrázek 27: Rozšíření třídy tahových vzorků

4.4 Tvorba třídy vzorků betonářské výztuže

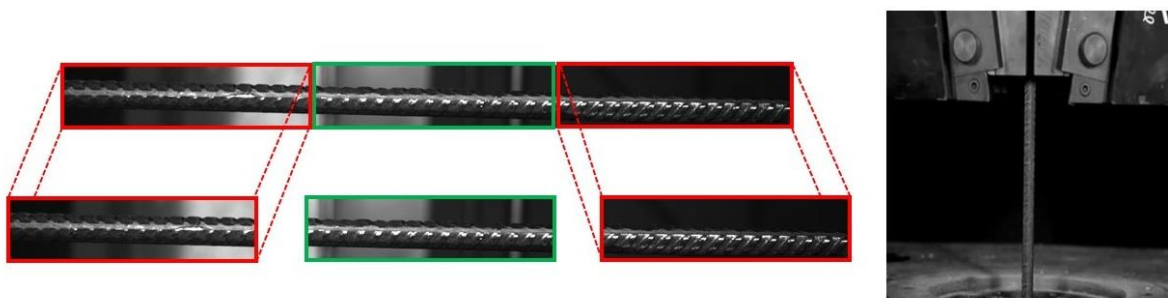
Druhou vytvořenou třídou objektů byla třída vzorků betonářské výztuže, respektive prutů betonářské výztuže, v rámci implementace pojmenovaná jako Roxor dle jejího slangového označení.

Základní předpoklady

Pro tuto třídu objektů byly vneseny stejné předpoklady jako pro třídu tahových vzorků.

Snímky z databáze X-Sight

Podobně jako u třídy tahových vzorků, byly jednotlivé záznamy ohodnoceny na základě toho, zda je vzorek malý nebo velký relativně vůči zornému poli kamery z důvodu uvedených



Obrázek 28: Princip oříznutí prutů betonářské výztuže (vlevo) a malý prut bet. v. (vpravo)

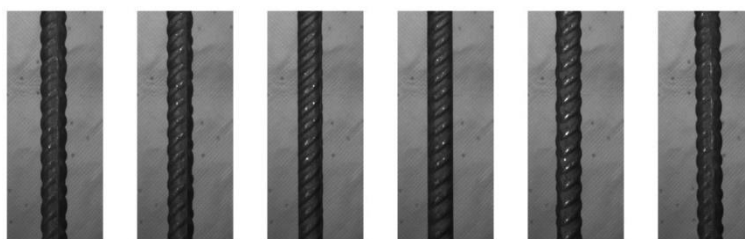
v předchozí podkapitole. Díky charakteru objektů v této třídě však nebylo použito kritérium viditelnosti z toho důvodu, že pruty betonářské výztuže jsou z hlediska tvaru podél jejich osy obvykle neměnné.

Druhé kritérium, jenž bylo v rámci této třídy použito, vyhodnocovalo, zda se jedná o záznam s velkým nepoměrem stran. Jelikož bylo v poskytnuté databázi mnoho těchto záznamů, jednotlivé snímky promítnuté do datasetu byly rozděleny na třetiny dle většího rozměru tak, aby z každého snímku vznikly tři snímky nové. Tento postup je znázorněn na obrázku 28 vlevo. Smyslem tohoto kritéria bylo omezit vliv velkého počtu záznamů s významným nepoměrem stran. Na obrázku 28 je dále znázorněno porovnání velkého oproti malému vzorku, kde velký vzorek je na uvedeném obrázku vlevo a malý vzorek vpravo. V tabulce 7 jsou obě kritéria uvedena a k nim příslušné počty snímků promítnutých do datasetu.

Tabulka 7: Kritéria pro třídu betonářské výztuže

Malý vzorek	Větší rozměr / menší rozměr ≥ 5	Snímky na záznam
Ano	Platí	V databázi nebylo obsaženo
Ne	Platí	20
Ano	Neplatí	100
Ne	Neplatí	40

Princip promítnutí snímků ze záznamů do datasetu byl stejný jako u třídy tahových vzorků popsány v předchozí podkapitole. Na rozdíl od třídy tahových vzorků bylo v dostupné databázi mnohem méně jednotlivých záznamů (viz. tabulka 5) a z toho důvodu bylo množství promítnutých snímků na záznam pro tuto třídu v porovnání s třídou tahových vzorků podstatně větší (tabulka 6 v porovnání s tabulkou 7). Dále, z tabulky 7 je patrné, že počet snímků ze záznamů malých vzorků byl značný. Tyto záznamy byly v dostupné databázi pouze tři. Celkový počet snímků promítnutých do výsledného datasetu pro třídu betonářské výztuže byl z dostupné databáze 800. Lze odvodit, že tři záznamy obsahující malé vzorky tvořily 3/8 snímků části



Obrázek 29: Otáčení prutu betonářské výztuže

datasetu pro tuto třídu z dostupné databáze. Tato skutečnost by mohla vyvolat u modelu tendenci učit se výhradně z této části datasetu, jinými slovy, model by mohl ztratit vůči této třídě zobecňovací schopnosti a schopnost pracovat nad rámec vytvořeného datasetu. Nicméně betonářská výztuž, respektive její pruty, jsou sémanticky jednoduchá tělesa a lze říct, že se liší pouze při rotaci (otáčení) vzorku kolem osy (obrázek 29). Podél osy je prut betonářské výztuže obvykle neměnný. Z toho důvodu byl tak velký počet malých vzorků této třídy do datasetu promítnut jako základ pro augmentace před trénováním a během trénování (proces popsán v podkapitolách 4.6 a 5.2). Dále, pro potlačení vzájemné podobnosti snímků z jednotlivých měření, bylo provedeno výraznější rozšíření této třídy oproti rozšíření třídy tahových vzorků o vlastní záznamy.

Rozšíření třídy betonářské výztuže o vlastní záznamy

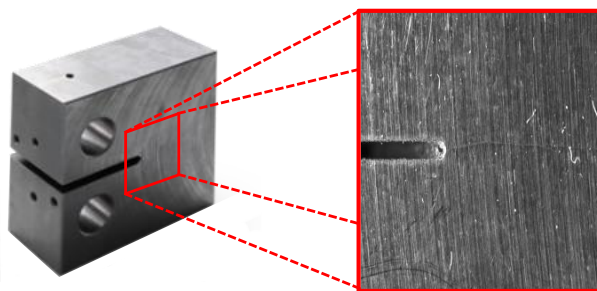
Podobně jako u třídy tahových vzorků byl dataset rozšířen o vlastní záznamy. Hardware pro jejich pořízení byl stejný jako je popsán v podkapitole 4.3. Jako zkušební vzorky byly použity pruty betonářské výztuže o různých průměrech. Celkem bylo použito 5 prutů. Takto bylo pořízeno celkově 60 nových záznamů. Jednotlivé záznamy stejných vzorků se od sebe lišily expozičním časem kamery, vzdáleností kamery od vzorku umístěného ve zkušebním stroji a pozadím. Dále, v rámci každého záznamu bylo provedeno otáčení vzorku (prutu) kolem jeho osy, a to z toho důvodu, že na prototypch trénovaných modelů pouze z dostupné databáze se ukázalo, že modely nebyly invariantní vůči natočení vzorků kolem jejich osy. Několik snímků z jednoho záznamu, na němž byl vzorek otáčen kolem své osy, je znázorněno na obrázku 29. Celkem bylo pořízeno 400 snímků prutů betonářské oceli, jenž byly promítnuty do datasetu.

4.5 Tvorba třídy vrubovaných těles

Třetí vytvořenou třídou objektů byla třída vrubovaných těles, respektive kořenů vrubů vrubovaných těles. Tato třída byla v rámci implementace pojmenována CT-MT dle typických zástupců těchto těles.

Základní předpoklady

Základní předpoklad této třídy vychází z charakteru měření lomových vlastností pomocí DIC. Měřená oblast, respektive oblast zájmu, obvykle bývá u kořene vrubu a dále podél osy vrubu předpokládaným směrem šíření trhliny (do tělesa). Na základě tohoto poznatku je předpoklad pro snímky této třídy stanoven tak, že snímek na vstupu do detekčního algoritmu obsahuje pouze kořen vrubu tělesa, jenž může být např. těleso CT, a dále oblast za kořenem vrubu. Na snímku pak není vidět těleso celé. Tento předpoklad může být omezující z hlediska použitelnosti detekčního algoritmu pro snímky, které neobsahují pouze zmíněné oblasti, nicméně umožňuje určitou invarianci vůči použitému vzorku (tělesu) pro měření lomových vlastností, protože v oblasti kořene vrubu se od sebe vzorky téměř neliší. Na obrázku 30 je znázorněna oblast tělesa splňující stanovený předpoklad pro třídu vrubovaných těles. Oproti předpokladům předchozích tříd může snímek obsahovat až dvě instance objektů této třídy. Toto je stanoveno na základě možné přítomnosti dvou vrubů (kořenů vrubů) v obraze např. u tělesa MT („*Middle-crack tension specimen*“).



Obrázek 30: Snímek splňující předpoklad pro třídu vrubovaných těles (vpravo část)

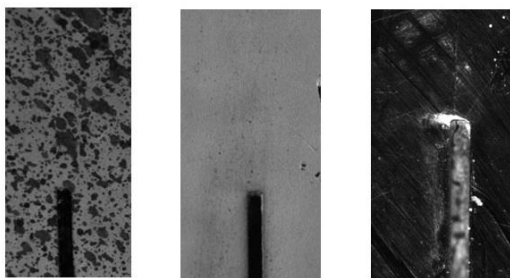
Snímky z databáze X-Sight

Z tabulky 5 je patrné, že jednotlivých záznamů pro tvorbu této třídy bylo v dostupné databázi ze všech tříd nejméně. Dalším problémem byla skutečnost, že jednotlivé záznamy jsou v průběhu zkoušky z vizuálního hlediska téměř totožné. Z toho důvodu, byla velká část této třídy vytvořena pomocí vlastních záznamů (popsáno dále). V rámci dostupné databáze bylo manuálně vybráno celkem 41 snímků, jenž byly promítnuty do datasetu. Tento počet byl následně uměle navýšen pomocí augmentace snímků, o které je pojednáno v podkapitole 4.6. Na obrázku 31 jsou znázorněny tři snímky z třech různých záznamů z dostupné databáze.

Rozšíření třídy vrubovaných těles o vlastní záznamy

Jak již bylo uvedeno, dostupná databáze neobsahovala dostatečné množství vizuálně odlišných snímků pro vytvoření třídy vrubovaných těles. Z toho důvodu bylo provedeno rozsáhlé doplnění datasetu o vlastní záznamy. Snímací aparatura obsahovala kameru Blackfly S o rozlišení 20 Mpx a telecentrický objektiv TC3MHR016-C a je znázorněna na obrázku 32 vpravo.

Celkem byla použity čtyři tělesa z čeho byly dvě typu CT a dvě typu MT. Pomocí těchto čtyř těles bylo pořízeno celkově 20 záznamů, jenž byly pořízeny tak, aby splňovaly stanovené předpoklady na tuto třídu, a aby co nejvíce odpovídaly charakteru záznamů, které byly k dispozici v dostupné databázi. Jednotlivé záznamy se od sebe lišily osvětlením a vertikální polohou vrubu v obraze. Každý záznam trval přibližně jednu minutu a snímková frekvence byla nastavena na 1 fps. Během záznamu bylo horizontálně posouváno se vzorkem tak, aby se pozice kořene vrubu měnila podél horizontální osy. Rychlost horizontálního pohybu byla dostatečně nízká, aby nedošlo k rozmazání snímků. Na obrázku 32 vlevo jsou znázorněny tři polohy kořene vrubu v rámci jednoho tělesa. Dále je na obrázku 32 vlevo znázorněna změna horizontální polohy kořene vrubu v rámci jednoho záznamu. Z každého záznamu takto vzniklo přibližně 60 snímků s různými horizontálními polohami kořene vrubu. Jak již bylo zmíněno, na základě



Obrázek 31: Snímky vrubovaných těles z databáze



Obrázek 32: Změna polohy kořene vrubu (vlevo) a snímací aparatura (vpravo)

charakteru měření lomových vlastností pomocí DIC je významná oblast za kořenem vrubu směrem do tělesa. Z toho důvodu, bylo z části záznamů vytvořených pomocí CT těles promítnuto do datasetu vždy prvních 20 snímků, kde byla poloha kořene vrubu horizontálně blíže ke kraji zorného pole kamery, což tvořilo celkem 200 snímků. Dále ze zbylých 40 snímků v každém záznamu bylo náhodně vybráno dalších 10, kde kořen vrubu zasahoval blíže ke středu záběru. To bylo provedeno z důvodu umožnění detekčnímu algoritmu přizpůsobit rozměry hraničního obdélníku ke kořeni vrubu. Celkově bylo promítnuto 300 snímků ze záznamů vytvořených pomocí CT těles. Ze záznamů vytvořených pomocí MT těles bylo promítnuto vždy 40 náhodných snímků z jednoho záznamu, tedy celkově 400 snímků. Z vlastních záznamů pro třídu vrubovaných těles bylo tedy celkově do datasetu promítnuto 700 snímků.

4.6 Augmentace snímků

Před samotným rozdělením snímků do trénovací množiny, validační množiny a testovací množiny bylo provedeno několik úprav datasetu pomocí augmentace snímků. Augmentace snímků je proces, při kterém jsou snímky upraveny pomocí různých transformací, jenž mají za cíl zvětšit robustnost datasetu. Mezi typické techniky augmentace snímků patří geometrické transformace (rotace, translace, zvětšení, zmenšení, zrcadlení, oříznutí atd.), úpravy hodnot barevného prostoru (úprava jasu, kontrastu, saturace a poměru barev) nebo přidání šumu do snímku [40]. Metod augmentace snímků existuje mnoho, nicméně je vhodné použít pouze takové, jenž mají v kontextu s řešeným problémem význam [61].

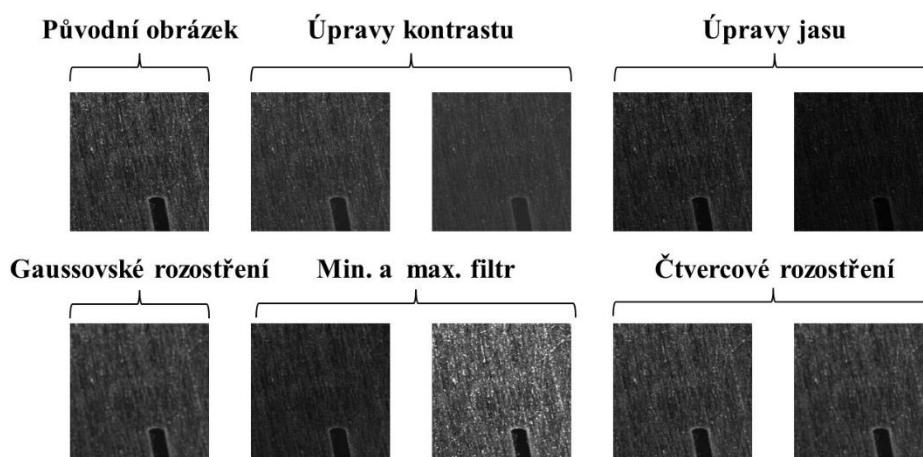
V rámci této práce byla augmentace snímků provedena ve dvou fázích. První fáze augmentace proběhla před procesem trénování a je popsána v této podkapitole. Druhá fáze probíhala během tréninku a je podrobně popsána v podkapitole 5.2.

První fáze augmentace snímků vytvořeného datasetu spočívala převážně v zvětšení robustnosti vytvořeného datasetu, potlačení vizuální podobnosti jednotlivých snímků (převážně pro třídu betonářské výztuže) a zvětšení počtu snímků třídy vrubovaných těles z dostupné databáze. V tabulce 8 jsou uvedeny jednotlivé části datasetu tříd tahových vzorků a betonářské výztuže spolu s počtem snímků před procesem augmentace, po procesu augmentace a celý proces je pro každou část popsán.

Tabulka 8: Augmentace třídy tahových vzorků a prutů betonářské výztuže

Třída	Zdroj	Počet snímků před augmentací	Počet snímků po augmentaci	Popis augmentace
Tahových vzorků	X-Sight databáze	945	945	Z každého záznamu byla náhodně vybraná polovina snímků otočena o 90°
	Vlastní záznamy	67	134	Všechny snímky byly otočeny o 90° a přidány do datasetu
Prutů betonářské výztuže	X-Sight databáze	800	800	Z každého záznamu byla náhodně vybraná polovina snímků otočena o 90°
	Vlastní záznamy	400	400	Z každého záznamu byla náhodně vybraná polovina snímků otočena o 90°

V rámci vlastních záznamů třídy vrubovaných těles bylo použito otočení o 90° náhodně vybrané poloviny snímků z každého záznamu, podobně jako u předchozích tříd (viz tabulka 9). Nicméně v rámci snímků z dostupné databáze bylo použito několik augmentačních metod uvedených v tabulce 9. Na obrázku 33 je znázorněn snímek z dostupné databáze a všechny jeho varianty po augmentaci. Na základě předpokladů stanovených pro tuto třídu, lze snímek této třídy rozdělit na dvě oblasti. První oblast je vrub, respektive kořen vrubu, a jeho blízké okolí, jenž má určitý tvar, který je z hlediska geometrie téměř neměnný. Druhou oblastí je povrch vzorku, jenž tvoří zbytek snímku. Použité metody augmentace byly zvoleny převážně z důvodů změny vizuální podoby povrchu vzorků k docílení invariance modelu vůči různým povrchům. Na základě toho byly i mimo jiné zvoleny metody augmentace částečného Gaussovského rozostření a částečného box rozostření, které ze své podstaty poněkud porušují požadavky stanovené metodou DIC, nicméně míra rozostření byla zvolena tak, aby významně nezměnila zaostření kořene vrubu a celkovou kvalitu snímků. Takto vzniklo z původních 41 snímků 410 snímků, které vstupovaly do druhé etapy augmentace (viz. tabulka 9).



Obrázek 33: Ukázka augmentovaných snímků vrubovaných těles z dostupné databáze

Tabulka 9: Augmentace třídy vrubovaných těles

Třída	Zdroj	Počet snímků před augmentací	Počet snímků po augmentaci	Popis augmentace
Vrubované těleso	X-Sight databáze	41	410	Úprava jasu, kontrastu, částečné Gaussovské rozostření, částečné box rozostření a morfologické filtry
		410	410	Náhodně vybrána polovina snímků a otočena o 90°
	Vlastní záznamy	700	700	Náhodně vybrána polovina snímků a otočena o 90°

Dále bylo v každé třídě vybráno náhodně 7.5 % snímků, na které bylo aplikováno částečné Gaussovské rozostření, což jak již bylo zmíněno, do jisté míry porušuje předpoklady použití metody DIC. Nicméně na základě některých záznamů obsažených v dostupné databázi, jež nebyly dokonale zaostřeny, byl tento postup navržen k vytvoření určité invariance algoritmu vůči takovým snímkům, aby mu bylo umožněno se částečně učit za hranici stanovených předpokladů.

4.7 Výsledný dataset

Ze zdrojů dostupné databáze a vlastních záznamů byl popsáním principem v předchozím textu vytvořen výsledný dataset. Tento dataset byl rozdělen na trénovací, testovací a validační množinu v poměru 70:15:15. Samotné rozdělení proběhlo tak, že z celkového počtu snímků bylo náhodně vybráno z každé třídy 70 % do trénovací množiny, 15 % do validační množiny a 15 % do testovací množiny. Počet snímků a instancí pro jednotlivé třídy a jednotlivé množiny datasetu je uveden v tabulce 10.

Tabulka 10: Počet snímků a instancí jednotlivých tříd a následně množin

Třída	Počet snímků	Počet instancí
Tahové vzorky	1079	1079
Betonářská výztuž	1200	1200
Vrubovaná tělesa	1110	1510
Množina datasetu	Počet snímků	Počet instancí
Trénovací	2373	2652
Validační v1	508	570
Testovací v1	508	567

Na základě popsaného principu tvorby datasetu, je zohledněna skutečnost, že se snímky ze stejných záznamů mohly promítnout zároveň do trénovací, validační a testovací množiny. Tyto snímky sice nebyly totožné, ale existuje mezi nimi určitá vizuální závislost, která by i přes proces dvoufázové augmentace mohla v procesu validace a testování modelu zkreslovat výsledky a jeho výkonnostní metriky. Na základě této skutečnosti byly vytvořeny další dva

datasety pouze za účelem validace a testování modelu a jsou popsány v podkapitole 4.8. Validační a testovací množiny vytvořené z dostupné databáze a vlastních záznamů uvedené v tabulce 10 jsou pro přehlednost označeny jako v1 (první varianty těchto množin).

4.7.1 Doplnění datasetu o snímky bez objektů

S ohledem na doporučení v dokumentaci algoritmu YOLOv8 bylo do datasetu přidáno přibližně 400 snímků, které neobsahovaly žádné objekty. Tento úkon napomáhá modelu redukovat počet možných falešných detekcí (FP) [38]. Dokumentace metody YOLOv8 uvádí přidání 0 % až 10 % takových snímků. Z celkového datasetu 400 snímků odpovídá přibližně 11 %, což je u horní hranice uvedeného doporučení. Tento přístup byl zvolen na základě charakteru detekovaných objektů. Jelikož se jedná o tvarově a sémanticky poměrně jednoduchá tělesa, hrozilo riziko častého zaměňování pozadí snímků za detekované objekty.

Snímky bez objektů byly použity převážně z dostupné databáze, která poskytuje relevantní snímky vůči snímkům detekované objekty obsahující, jinými slovy obsahuje kontextuálně příbuzné snímky. Dále v rámci třídy vrubovaných těles snímky bez objektů obsahovaly část snímků, na nichž se vyskytoval pouze povrch vzorků, aby bylo algoritmu umožněno se učit, že ne v každém snímku splňujícím předpoklad pro třídu vrubovaných těles musí být kořen vrubu, jenž je snadno zaměnitelný např. za větší strukturu náhodného vzoru. Část snímků bez objektů byla také pořízena během rozšiřování datasetu o vlastní záznamy pro třídu tahových vzorků a třídu betonářské výztuže.

4.8 Validační a testovací množiny

Jak již bylo popsáno, na základě principu tvorby výchozího datasetu hrozilo riziko, že dojde ke zkrácení procesu validace a testování modelu. Z toho důvodu byly vytvořeny další dva datasety, přičemž první z nich byl použit k validaci a testování modelu a druhý byl použit pouze k testování. Tyto datasety jsou pro přehlednost označeny v2 a v3 (druhé a třetí varianty).

První zmíněný dataset byl vytvořen v laboratoři Ústavu mechaniky těles FSI VUT. Pro jeho vytvoření bylo použito celkem 15 tahových vzorků, 6 prutů betonářské výztuže, 4 tělesa typu CT, 3 tělesa typu MT a jedno těleso typu SENB. Všechny pořízené snímky splňovaly předpoklady stanovené na jednotlivé třídy. Některé vzorky použité při vytváření tohoto datasetu byly použity i při vytváření, respektive rozšiřování, původního datasetu, což je popsáno v předchozím textu, nicméně při vytváření tohoto datasetu byl kladen důraz na vytvoření co nejroznorodějších záznamů vzorků za použití různých pozadí, různých kompozic, různých světelných podmínek a různých vzdáleností kamery od vzorků. Celkově bylo pořízeno 452 různých snímků, což bylo navýšeno augmentací na dvojnásobek pomocí otočení každého snímku o 90°. Dataset byl následně rozdělen náhodně na novou validační a novou testovací množinu v poměru 1:1. Nově vzniklá dvojice validační a testovací množiny spolu s původním trénovací množinou tvořily poměr přibližně 72:14:14 (trénovací ku validační v2 ku testovací v2). V tabulce 11 jsou uvedeny počty snímků a instancí všech tříd a nově vzniklých množin.

Tabulka 11: Počet snímků a instancí jednotlivých tříd a následně množin v2

Třída	Počet snímků	Počet instancí
Tahové vzorky	238	238
Betonářská výztuž	408	408
Vrubovaná tělesa	258	378
Množina datasetu	Počet snímků	Počet instancí
Validační v2	452	509
Testovací v2	452	515

Druhý zmíněný dataset byl vytvořen pouze za účelem testování, a to pro třídu tahových vzorků a třídu betonářské výztuže. Snímky, které obsahoval, byly z části veřejně dostupné na internetu a z části snímky z dostupné databáze, jenž nebyly použity v předchozích množinách (např. snímky co přibyly do databáze po vytvoření výsledného datasetu). Tyto snímky mnohdy nespĺňovaly předpoklady kladené na obě třídy, protože často obsahovaly víc než jednu instanci objektu a ve většině případů neměly dostatečné rozlišení z pohledu použití DIC (platí pro snímky mimo dostupnou databázi). Nicméně jelikož byl tento dataset vytvořen pouze za účelem testování, lze na něm otestovat algoritmus mimo specifickou oblast použití a odvodit díky němu určité přednosti a omezení algoritmu. Toto platí pouze pro zmíněné třídy v tomto datasetu obsažené, protože z charakteru předpokladů stanovených na třídu vrubovaných těles bylo velice obtížné vytvořit dostatečný počet dostupných snímků z internetu pro testování modelu, proto nebyla tato třída v tomto testovacím datasetu obsažena. Označení testovací množiny vzniklé z tohoto testovacího datasetu je v3 a v tabulce 12 jsou uvedeny počty snímků a instancí jednotlivých tříd a výsledné množiny.

Tabulka 12: Počet snímků a instancí jednotlivých tříd a následně množiny v3

Třída	Počet snímků	Počet instancí
Tahové vzorky	100	150
Prut betonářské výztuže	68	80
Množina datasetu	Počet snímků	Počet instancí
Testovací v3	168	230

4.9 Anotace snímků

Nedílnou součástí vytváření detekčního algoritmu pomocí strojového učení, a to na bázi učení s učitelem, je anotování objektů. Anotování (označování) objektů je proces, při kterém jsou u snímků vytvářeny hraniční obdélníky (anotace) obsahující hledané objekty. Pomocí těchto anotací se pak algoritmus učí detekovat hledané objekty v obraze [21].

K anotovanému snímku je typicky připojen externí soubor obsahující souřadnice hraničních obdélníků. V kontextu algoritmu YOLO, respektive YOLOv8, jsou hraniční obdélník popsán souřadnicemi středu a jeho šířkou a výškou. Tyto souřadnice a rozměry jsou normalizovány vůči rozměrům snímku na vstupu do algoritmu.

K anotování patří dodržování určitých zásad. První zásadou je, že hraniční obdélníky by měly co nejtěsněji opisovat hledaný objekt. Druhou zásadou je, že anotované objekty, případně

oblasti, by měly zůstat konzistentní během celého anotačního procesu [41]. Anotované oblasti jsou u jednotlivých tříd znázorněny na obrázku 34.

Třída tahových vzorků a třída betonářské výztuže byly v rámci této práce anotovány pomocí polygonů, které byly následně převedeny na hraniční obdélníky vyžadované algoritmem YOLOv8. U třídy tahových vzorků byla anotována oblast nejmenšího průřezu vzorku (znázorněno na obrázku 34 vlevo) a u třídy betonářské výztuže byl anotován celý prut jenž byl viditelný v rámci snímku, případně jenž byl dostatečně viditelný v čelistech zkušebního zařízení (znázorněno na obrázku 34 na druhém místě zleva).

Polygony byly použity z toho důvodu, že je pomocí nich možné lépe vystihnout hranice objektu, ze kterých lze ve většině případů určit hraniční obdélník přesněji než při anotaci pomocí hraničního obdélníku přímo. Polygony byly převedeny na hraniční obdélníky jednoduchými vztahy:

$$x_s = \frac{\max(X) + \min(X)}{2} \quad (25)$$

$$y_s = \frac{\max(Y) + \min(Y)}{2} \quad (26)$$

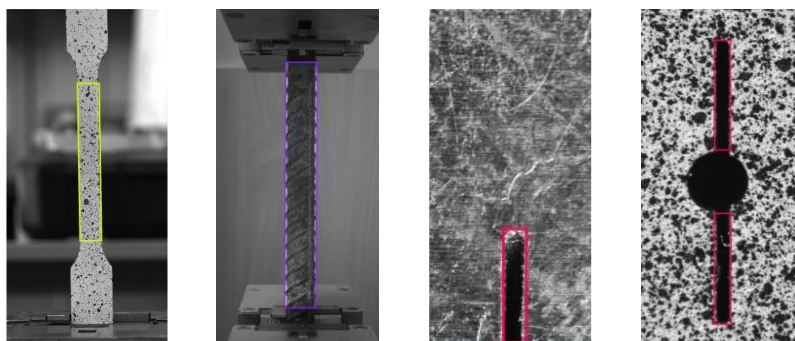
$$w = \frac{\max(X) - \min(X)}{2} \quad (27)$$

$$h = \frac{\max(Y) - \min(Y)}{2} \quad (28)$$

kde X představuje množinu x -ových souřadnic vrcholů polygonu a Y představuje množinu y -ových souřadnic vrcholů polygonu, x_s a y_s představují souřadnice středu hraničního obdélníku a w a h představují šířku a výšku hraničního obdélníku.

Další výhodou použití polygonů je skutečnost, že je lze snadno převést do různých anotačních formátů pro různé detekční algoritmy a dataset tak není vázán pouze na algoritmus YOLOv8 a algoritmy užívající stejný anotační formát.

U třídy vrubovaných těles byl anotován kořen vrubu a dále část oblasti vrubu směrem od kořene do tělesa, což se vyskytuje v případě CT těles (obrázek 34 druhé místo zprava). V případě MT těles, kde se vyskytují dva kořeny vrubu, byly anotovány oba kořeny směrem k díře (obrázek 34 vpravo).



Obrázek 34: Anotované oblasti vzorků

5 Implementace modelu

Tato kapitola se zabývá implementací algoritmu YOLOv8 (v kontextu vlastní práce s algoritmem je často označován jako model). V rámci celé implementace bylo trénováno mnoho modelů, jenž se lišily optimalizačním algoritmem a hodnotami hyperparametrů. Tyto modely byly následně porovnány na testovacích množinách.

Úvodem této kapitoly je nutné zmínit, že pro práci se všemi modely byla stanovena velikost snímků na 960x960 pixelů. Rozměry všech snímků byly převedeny na tuto velikost a všechny snímky byly dále převedeny do stupně šedi s 8-bitovou hloubkou. Nepoužití barev, respektive jejich matematické reprezentace, může být omezující faktor, nicméně bylo předpokládáno, že barvy nemají ze sémantického hlediska v kontextu rozpoznávaných tříd objektů velký význam. Navíc digitální korelace obrazu, jenž je svázána s předmětem práce, používá černobílé fotky.

5.1 Volba velikosti modelu

Algoritmus YOLOv8 implementovaný společností Ultralytics má několik variant. Jednotlivé varianty se od sebe liší specifikacemi architektury a velikostí. V tabulce 13 je uvedeno srovnání variant dle počtu trénovatelných parametrů, mAP50-95 na testovacím datasetu COCO a rychlosti průběhu snímku ze zmíněného datasetu na jednotném výpočetním prostředí.

Tabulka 13: Srovnání variant algoritmu YOLOv8 [37]

Varianta	Počet parametrů v miliónech	mAP50-95	Rychlost [ms]
YOLOv8n (nano)	3.2	33.7	80
YOLOv8s (small)	11.2	44.9	128
YOLOv8m (medium)	25.9	50.2	234
YOLOv8l (large)	43.7	52.9	375
YOLOv8x (extra large)	68.2	53.9	479

Z tabulky je patrné, že větší varianty algoritmu dosahují větších hodnot mAP50-95 na úkor rychlosti. Jelikož, jak již bylo zmíněno, předmětem práce je detekovat sémanticky jednoduchá tělesa, proto byla zvolena varianta YOLOv8s, která oproti větším modelům dosahuje nižší hodnoty mAP50-95 na uvedeném testovacím datasetu, nicméně po modelu YOLOv8n dosahuje největší rychlosti detekce. Dalším důvodem pro výběr varianty YOLOv8s byla skutečnost, že při optimalizaci hyperparametrů, která je popsána v dále, je nutné trénovat v součtu až tisíce epoch, což je časově velmi náročné a tato časová náročnost se zvětšuje s počtem trénovatelných parametrů algoritmu.

Dále je nutné doplnit, že v rámci celé implementace byly použity pouze předtrénované modely. Předtrénované modely jsou v kontextu CNN efektivní v tom, že nemají náhodně inicializované trénovatelné parametry sítě, což jsou například filtry na konvolučních vrstvách. Filtry v prvních konvolučních vrstvách typicky zodpovídají za extrakci základních příznaků a jsou v rámci detekce různorodých těles podobné, proto je výhodné použít modely, jenž mají tyto filtry předtrénované a během tréninku na vlastních datech dochází pouze k úpravě vah vůči

řešené úloze [42]. Tato technika může výrazně zrychlit proces učení. Dostupné modely jsou předtrénované na datasetu COCO [43].

5.2 Trénink základního modelu

Proces tréninku je řízen sadou hyperparametrů, které jsou stanoveny před jeho začátkem. Hyperparametry se dají obecně rozdělit do dvou skupin. Tyto skupiny jsou hyperparametry související s architekturou algoritmu a hyperparametry související s procesem trénování. V kontextu použití algoritmu YOLOv8 implementovaného společností Ultralytics přibývá další skupina hyperparametrů související s augmentacemi dat během tréninku. Tato varianta augmentace je v práci označena jako druhá fáze augmentace s ohledem na první fázi augmentace, jež proběhla v procesu tvorby datasetu popsáném v podkapitole 4.6.

Hyperparametry související s architekturou modelu byly přímo dány použitím zvolené varianty YOLOv8s. Úplná architektura algoritmu YOLOv8 je znázorněna v přílohách k práci. Varianta YOLOv8s má 225 vrstev a 11 166 560 parametrů.

V rámci tréninku základního modelu byla zvolena výchozí sada hyperparametrů související s procesem tréninku (předdefinovaná autory) a jako optimalizační algoritmus byl zvolen algoritmus SGD se zohledněním setrvačnosti. Tyto hyperparametry jsou uvedeny v tabulkách 14 a 15 a pro každý z nich je uveden popis. K výchozím hyperparametrům souvisejícím s procesem tréninku, respektive jejich hodnotám, je nutné uvést, že se jedná o takové hyperparametry, jenž byly optimalizovány na veřejných testovacích datasetech (např. dataset COCO) [43]. Veřejné testovací datasety, jenž jsou popsány v podkapitole 2.3.9, jsou komplexní a rozsáhlé a lze tedy očekávat, že základní model v kontextu s řešeným problémem dosáhne ucházejících výsledků.

Tabulka 14: Výchozí hyperparametry související s procesem tréninku převzato z [46]

Označení hyperparametru	Název hyperparametru	Hodnota	Popis
<i>lr0</i>	Počáteční krok učení	0.01	Udává míru úpravy vah a prahů.
<i>lrf</i>	Činitel konečného kroku učení	0.01	Součin tohoto hyperparametru spolu s počátečním krokem učení dává konečný krok učení ($lr0 \cdot lrf$).
<i>momentum</i>	Setrvačnost	0.937	Hodnota setrvačného členu v optimalizačním algoritmu.
<i>weight_decay</i>	Tuhnutí vah	0.0005	Penalizace velkých vah proti přetrénování.
<i>warmup_epochs</i>	Počet rozehrívacích epoch	3	Rozehrívací fáze na začátku tréninku, při kterém je povolna navyšování krok učení na jeho počáteční hodnotu ke stabilizaci tréninku.

Tabulka 15: Výchozí hyperparametry související s procesem tréninku (pokračování)

Označení hyperparametru	Název hyperparametru	Hodnota	Popis
<i>warmup_momentum</i>	Rozehřívací setrvačnost	0.8	Hodnota setrvačnosti optimalizačního algoritmu při rozehřívací fázi ke stabilizaci tréninku.
<i>warmup_bias</i>	Rozehřívací bias	0.1	Hodnota kroku učení pro prahy neuronů v rozehřívací fázi ke stabilizaci tréninku.
<i>Box</i>	Váhy komponent chybové funkce	7.5	Doplněno v textu.
<i>Cls</i>		0.5	
<i>Dfl</i>		1.5	

Chybová funkce algoritmu YOLOv8 se skládá ze součtu tří komponent a každá komponenta je ve výsledném vyjádření chybové funkce vynásobena příslušnou váhou (naznačeno v pravé části obrázku 23). Váhy ve výchozím nastavení algoritmu jsou uvedeny v tabulce 15. První komponentou chybové funkce je CIOU („*Complete Intersection over Union*“), která vychází z podstaty IoU chybové funkce, ale na rozdíl od ní zohledňuje i vzdálenost středů predikovaného a skutečného hraničního obdélníku a jejich rozměry [44]. Příslušné označení její váhy je *Box* a její minimalizace v procesu učení zodpovídá za zpřesňování lokalizace hraničních obdélníků. Druhou komponentou chybové funkce je binární křížová entropie (BCE), což je speciální případ křížové entropie uvedené v teoretické části práce (tabulka 2). BCE je vhodná pro binární úlohy. V implementaci YOLOv8 je předpokládáno, že detekované třídy jsou vůči sobě nezávislé a BCE je vyhodnocena pro každou zvlášť. Výsledná hodnota této komponenty chybové funkce je pak suma jednotlivých entropií všech tříd. Příslušné označení její váhy je *Cls*, určuje chybu klasifikace objektu do definovaných tříd a její minimalizací se model učí rozlišovat jednotlivé třídy mezi sebou [31]. Třetí komponenta chybové funkce je DFL (DFL je zkratkou pro „*Distribution Focal Loss*“), což část chybové funkce, která podobně jako CIOU zodpovídá za lokalizaci hraničních obdélníků, nicméně na rozdíl od ní neurčuje přímo chybu souřadnic hraničního obdélníku, ale určuje chybu rozdělení možných poloh hraničních obdélníků, respektive možných ofsetů hraničních obdélníků, od skutečných hraničních obdélníku (její váha je označena *Dfl*). Princip, jímž se autoři YOLOv8 inspirovali je popsán v článku [45] a umožňuje modelu se lépe učit ze složitých snímků, které jsou často hustě pokryté různými objekty. Zjednodušená verze vztahu pro určení celkové chybové funkce se všemi jejími komponentami je následující:

$$E_{loss} = Box \cdot CIOU + Cls \cdot BCE + Dfl \cdot DFL \quad (29)$$

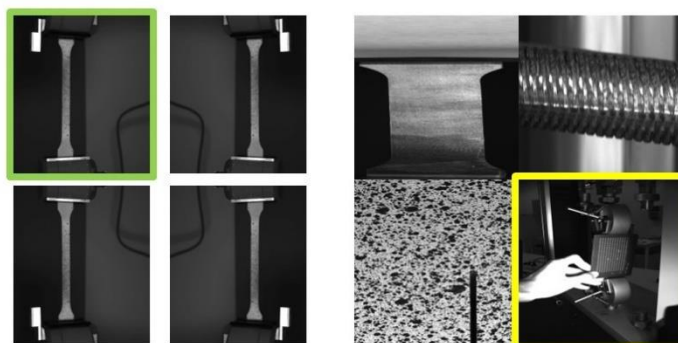
Sada hyperparametrů související s augmentací snímků během tréninku je uvedena a popsána v tabulce 16. Ze zmíněné tabulky je patrné, že druhá fáze augmentace zanesla do datasetu, respektive jeho trénovací množiny, značnou variabilitu snímků. Na obrázku 35 vlevo je uveden příklad jednoho snímku z trénovací množiny (zvýrazněn zeleně) a všechny konfigurace, ve kterých se mohl během tréninku vyskytovat díky hyperparametrům horizontálního a vertikálního zrcadlení. Dále je z tabulky patrné, že augmentace pomocí mozaiky se vyskytovala vždy, jelikož je považována za velmi silnou augmentační techniku,

která při zkombinování s ostatními augmentačními technikami vytváří ze snímků nové kompozice a zanáší do trénovací množiny větší robustnost [46]. Příklad mozaiky snímků je uveden na obrázku 35 vpravo. V uvedené mozaice se na žlutě zvýrazněném snímku nevyskytuje žádný objekt.

Tabulka 16: Hyperparametry související s procesem druhé fáze augmentace

Označení hyperparametru	Název hyperparametru	Hodnota	Popis
<i>hsv_v</i>	Úprava jasu	0.75	Provede změnu jasu o náhodnou hodnotu v intervalu $\langle 0,0.75 \rangle$.
<i>translate</i>	Translace snímku	0.2	Posune snímek v náhodném směru o náhodnou hodnotu do 20 % jeho rozměru.
<i>scale</i>	Úprava měřítka	0.5	Provede změnu měřítka o náhodnou hodnotu v intervalu $\langle -0.5,0.5 \rangle$.
<i>flipud</i>	Horizontální zrcadlení	0.5	Provede horizontální zrcadlení snímku s pravděpodobností 50 %.
<i>flipur</i>	Vertikální zrcadlení	0.5	Provede vertikální zrcadlení snímku s pravděpodobností 50 %.
<i>mosaic</i>	Mozaika	1.0	Vytvoří náhodnou kombinaci ze čtveřic snímků z trénovací množiny do jednoho snímku s pravděpodobností 100 %.

Implementace algoritmu YOLOv8 umožňuje mnoho dalších augmentačních technik jako například rotaci, zkosení nebo různé úpravy v barevném prostoru. Vyjmenované augmentační techniky nebyly použity z toho důvodu, že z hlediska charakteru řešené úlohy neměly význam a pomocí nich vzniklé snímky by modelu nepřinesly užitečné informace. Rotace sice určitý význam má, nicméně v oblasti testování materiálů s použitím DIC se vzorky zarovnávají horizontálně či vertikálně [67]. K tomu je nutno doplnit, že některé vzorky v záznamech v dostupné databázi nebyly z hlediska zarovnání naprosto horizontální, případně vertikální, a tím pádem měl model tuto informaci k dispozici v rámci vytvořeného datasetu do takové míry,



Obrázek 35: Příklady augmentací snímků během tréninku

kteřá se u záznamů pořízených při standartních DIC měření vyskytuje. Např. na obrázku 35 v uvedené mozaice snímek v pravém horním rohu, u kterého je vhodné doplnit, že míra natočení je částečně zkreslená vlivem změny velikosti a v původním snímku nebylo natočení (nezarovnání) tak znatelné. Zkosení snímků nemělo z hlediska řešené úlohy fyzikální význam a úpravy barevného prostoru, kromě použité úpravy jasů, jsou z hlediska černobílých snímků nepoužitelné.

Další parametry související s tréninkem jsou vedeny v tabulce 17. Jedním z uvedených parametrů je parametr *batch* neboli dávka snímků. V rámci implementace byla použita funkce Auto-batch, což je funkce, jenž umožní vyhodnocení optimální velikosti dávky na základě možností výpočetního prostředí. S hardwarem, jenž byl použit k výpočtu byla optimální dávka vyhodnocena jako 16 snímků.

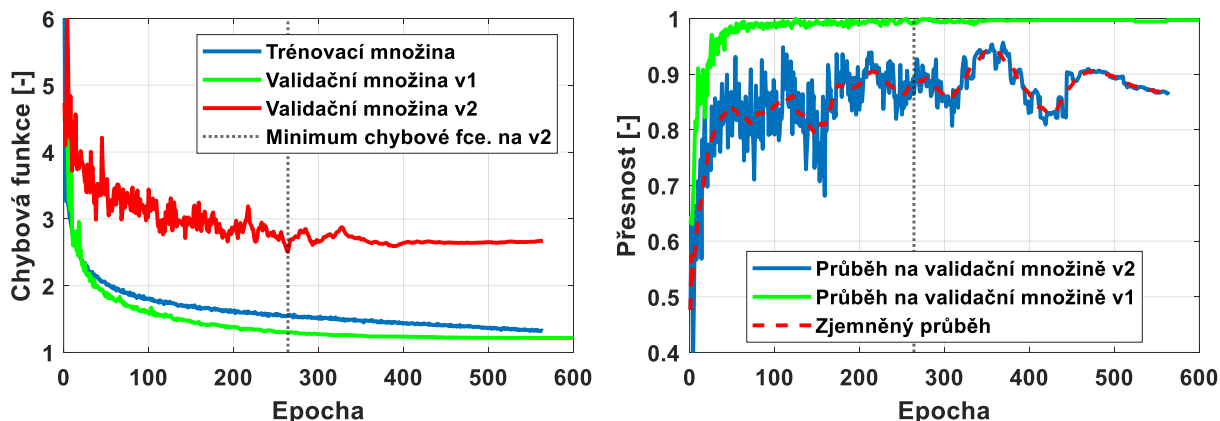
Tabulka 17: Další parametry související s tréninkem

Označení hyperparametru	Název hyperparametru	Hodnota	Popis
<i>batch</i>	Dávka	16	Doplněno v textu.
<i>epochs</i>	Počet epoch	700	Celkový počet epoch trénování.
<i>patience</i>	Trpělivost	300	Počet epoch, po kterém je trénink přerušen, pokud nedojde ke zlepšení ve validačních metrikách.
<i>save_period</i>	Perioda ukládání vah	20	Perioda, ve které dochází k ukládání vah modelu.

Kombinace celkového počtu epoch tréninku a hodnoty trpělivosti (*patience*) modelu byla zvolena tak, aby se ukázalo, jak se model chová za hranicí minima validačních metrik v oblasti, kde hrozí k přetrénování. Nicméně implementace YOLOv8 umožňuje periodicky ukládat hodnoty vah v definovaných epochách (uvedeno v tabulce 17) a navíc na konci každého tréninku dojde k uložení nejlepšího stavu modelu v průběhu tréninku z hlediska validačních metrik.

Průběh tréninku a výkonnostní metriky

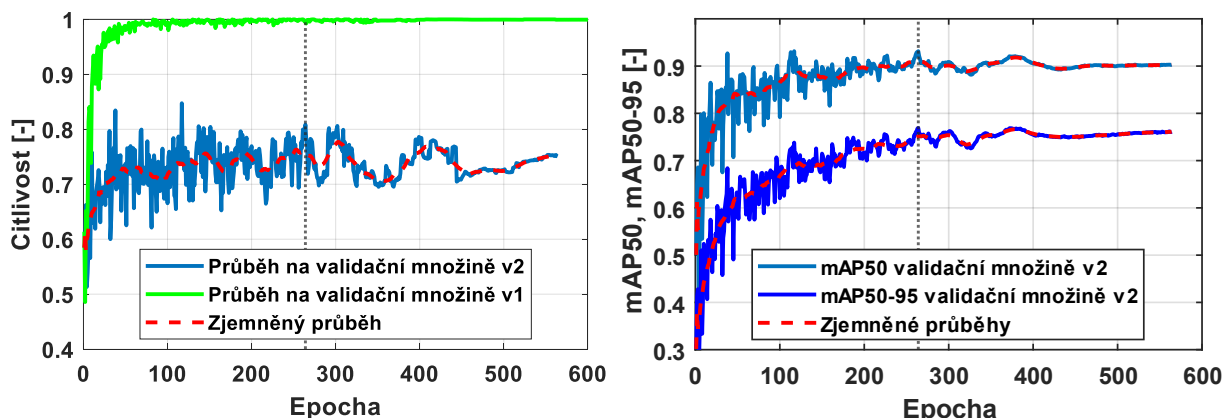
Na obrázcích 36 a 37 jsou znázorněny průběhy chybové funkce, přesnosti a citlivosti na trénovací a obou validačních množinách. Mezi jednotlivými průběhy je velmi zřetelný rozdíl. V průběhu tréninku jsou hodnoty chybové funkce u množiny v1 oproti množině v2 menší a celý průběh je ustálenější. Všechny výkonnostní metriky jsou u množiny v1 oproti množině v2 výrazně větší a podstatně rychleji saturují k jejich maximu. Ke zmíněné saturaci dochází v oblasti kolem sté epochy tréninku, ve které je na grafu průběhu chybové funkce na validační množině v1 (obrázek 36 vlevo) patrný konec prudkého a silně nelineárního průběhu. Tyto skutečnosti napovídají faktu, že vytvořená validační množina v1 popsáním principem v kapitole 4 do jisté a poměrně značné míry zkresluje výsledky modelu. Z toho důvodu jsou modely dále vyhodnoceny pouze na validační množině v2 a testovacích množinách v2 a v3. Dále jsou na obrázku 37 vpravo uvedeny průběhy metrik mAP50 a mAP50-95 pouze na validační množině v2.



Obrázek 36: Průběh chybové funkce (vlevo) a průběh přesnosti (vpravo) základního modelu

Z obrázku 36 vlevo je patrné, že nejnižší hodnota chybové funkce se na validační množině v2 vyskytla v epoše 264 (naznačeno vertikální přerušovanou čarou). Samotný trénink byl přerušen až na epoše 564 s ohledem na hyperparametr *patience* stanovený na hodnotu 300. Hodnota chybové funkce kolem jejího globálního minima osciluje a dále se postupně zvětšuje. Lze očekávat, že při pokračování tréninku za poslední epochu, by došlo k prudkému nárůstu chybové funkce na validační množině v2 a tím pádem i přetrénování modelu.

Optimální hodnoty parametrů sítě jsou z hlediska odlišných výkonnostních metrik různé. Z hlediska minimální hodnoty chybové funkce je optimální již zmíněná epocha 264. Z hlediska nejlepší přesnosti epocha 369, z hlediska nejlepší citlivosti epocha 117, z hlediska metrik mAP50 a mAP50-95 epocha 264. Nicméně je nutné zohlednit silnou oscilaci ve všech průbězích, a to především v oblasti před globálním minimem chybové funkce (tedy před 264 epochou). Konkrétně u přesnosti a citlivosti, jejichž průběhy mají obvykle opačné trendy, tedy při zvětšení přesnosti dochází ke snížení citlivosti a obráceně, což se silně projevuje ve velkých lokálních minimech a maximech těchto metrik. Popsaný jev je dobře pozorovatelný z průběhu zmíněných metrik mezi epochami 300 a 400. Reprezentativní stav modelu byl určen na minimum chybové funkce, tedy v epoše 264, a vstupoval do porovnání s ostatními modely (viz podkapitola 5.4).



Obrázek 37: Průběhy citlivosti (vlevo) a metrik mAP (vpravo) základního modelu

5.2.1 Porovnání optimalizačních algoritmů

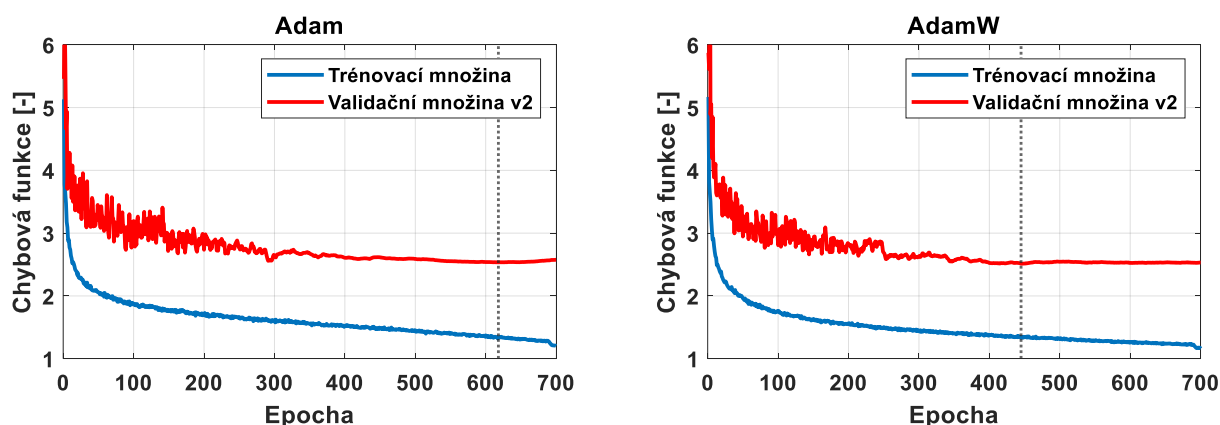
Jak již bylo zmíněno, ve výchozím nastavení YOLOv8s je použit opt. algoritmus SGD se zohledněním setrvačnosti. V rámci porovnání vlivu optimalizačního algoritmu na průběh tréninku a výsledky modelu byly provedeny další dva tréninky. U jednoho byl použit jako optimalizační algoritmus Adam a u druhého AdamW. Algoritmus Adam je popsán v teoretické části práce. AdamW je stejně jako Adam z rodiny optimalizačních algoritmů s adaptivním gradientem, nicméně od algoritmu Adam se liší tím, že penalizaci velkých vah, za kterou zodpovídá parametr tuhnutí vah (*weight_decay*), neaplikuje na gradienty (rovnice 19), ale přímo na váhy [47].

Hyperparametry těchto tréninků se lišily od základního modelu pouze v hodnotě počátečním kroku učení. Počáteční krok učení pro algoritmy Adam a AdamW byl 0.001. Dále, Adam i AdamW obsahují dva parametry zohledňující setrvačnost (rovnice 20 až 23). Prvním je parametr β_1 jehož hodnota byla totožná jako pro parametr setrvačnosti u základního modelu ($\beta_1 = 0.937$). Druhým je parametr β_2 , jehož hodnota byla $\beta_2 = 0.999$ [50]. Další hyperparametry procesu tréninku a procesu augmentací snímků během tréninku zůstaly totožné jako v tabulkách 14 až 17.

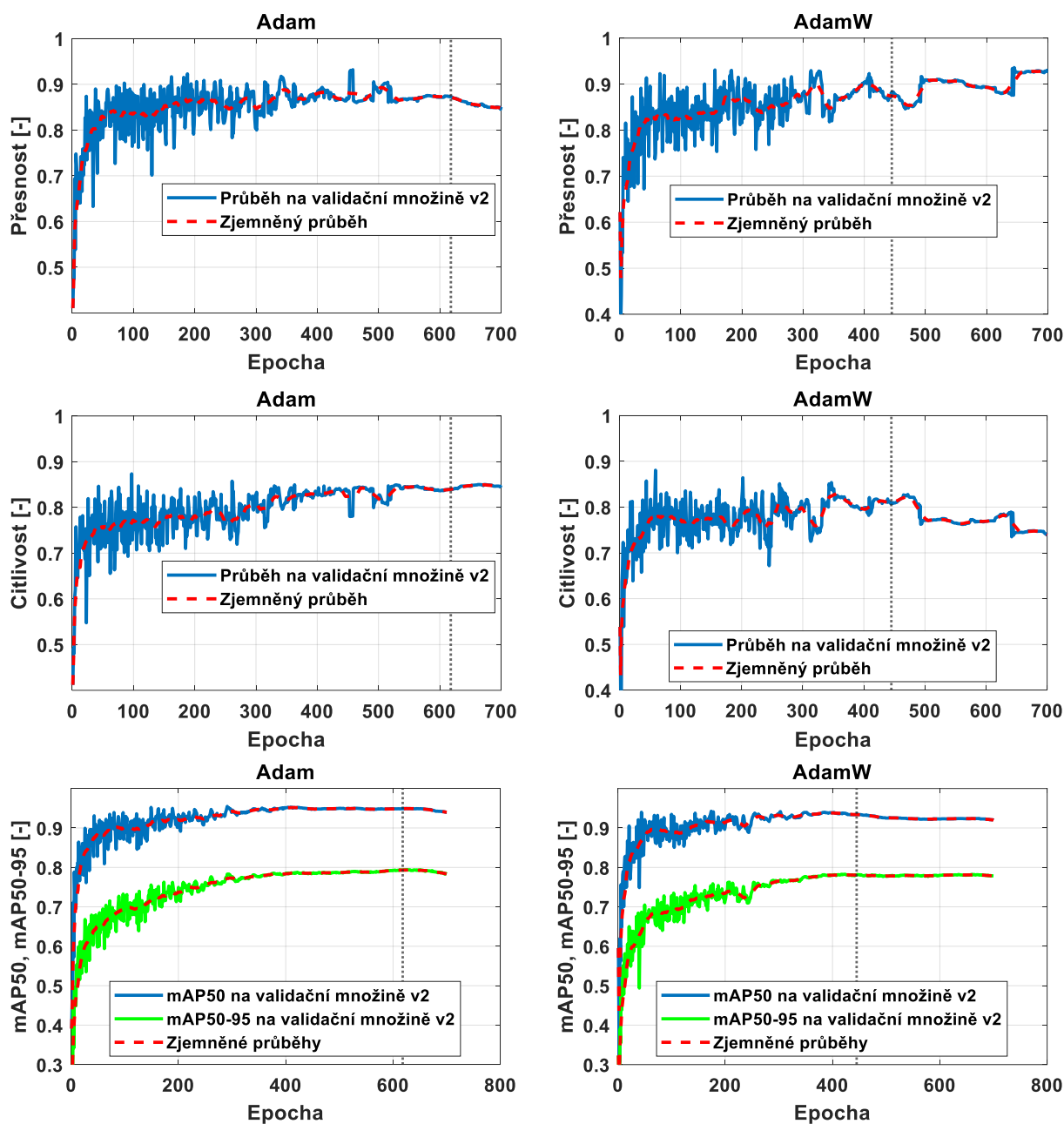
Průběhy tréninků a výkonnostní metriky

Na obrázcích 38 a 39 jsou znázorněny průběhy chybové funkce a výkonnostních metrik modelů s optimalizačním algoritmem Adam (vlevo) a AdamW (vpravo). Z průběhů je patrné, že došlo k dotrénování obou modelů až na stanovený maximální počet epoch (700). Minimální hodnota chybové funkce se u modelu vlevo vyskytla na epoše 618 a u modelu vpravo na epoše 445, což znamená, že s ohledem na hodnotu *patience* nedošlo k přerušení tréninku. Dále lze z průběhů výkonnostních metrik pozorovat, že u obou modelů nedocházelo oproti základnímu modelu k tak velkým oscilacím a je patrné, že oba optimalizační algoritmy dosahovaly v ustáleném stavu oproti základnímu modelu menší přesnosti, ale větší citlivosti a větších hodnot metrik mAP50 a mAP50-95.

Pro model s optimalizačním algoritmem AdamW byl jako reprezentativní stav zvoleno minimum chybové funkce (445 epocha) a pro model s opt. algoritmem Adam byl zvolen reprezentativní stav v epoše 400, což je sice 218 epoch od minima chybové funkce, avšak z průběhu chybové funkce, a i validačních metrik je patrné, že během dalšího tréninku již nedošlo k významnému zlepšení a výsledky modelu se od minima chybové funkce lišily do 3 %.



Obrázek 38: Průběhy chybové funkce opt. algoritmu Adam (vlevo) a AdamW (vpravo)



Obrázek 39: Průběhy chybové funkce a všech výkonnostních metrik základních modelů s opt. algoritmy Adam (levý sloupec) a AdamW (pravý sloupec)

5.3 Optimalizace hyperparametrů

Optimalizace hyperparametrů je proces, při kterém jsou pomocí optimalizačních metod hledány takové hodnoty hyperparametrů, jenž efektivně minimalizují chybovou funkci a dosahují optimálních výkonnostních metrik [48].

Optimalizačních metod existuje mnoho, nicméně pro účely této práce byly použity dvě. První metodou byl genetický algoritmus, který je implementován v knihovně *ultralytics* pro optimalizaci poskytovaných modelů. Druhou metodou bylo náhodné prohledávání prostoru hyperparametrů.

V rámci optimalizačního procesu byl zvolen suboptimální počet epoch na hodnotu 100. Suboptimální počet epoch, je počet epoch, po kterém dojde k přerušení tréninku a vyhodnocení

aktuálního stavu modelu. V ideálním případě by měly jednotlivé tréninky proběhnout na optimální počet epoch, jenž byl u výchozích modelů pro různé optimalizační algoritmy odlišný, nicméně ve všech případech výrazně větší než uvedený suboptimální počet. Suboptimální počet byl stanovený na základě toho, že při procesu optimalizace hyperparametrů je typicky nutné trénovat až tisíce epoch, což je z hlediska výpočetního času extrémně náročné. Dále, z průběhu uvedených chybových funkcí je patrné, že na sté epoše byly všechny dosud uvedené modely za oblastí silné nelinearity a velkého poklesu v jejích hodnotách, což je oblast, která je pro vyhodnocení potenciálního výkonu modelu neobjektivní.

V rámci optimalizace byl u všech modelů použit algoritmus Adam nebo AdamW a to z toho důvodu, že na rozdíl od SGD se u těchto algoritmů ve výchozích trénincích neprojevovala tak silná oscilace v metrikách a to především v přesnosti a citlivost, což může značně zkreslit výsledky modelu na suboptimální epoše, nicméně výskyt oscilace je nutné při vyhodnocení modelů zohlednit. Dále oba zmíněné algoritmy dosahovaly oproti SGD v ustáleném stavu větších hodnot citlivosti, mAP50 a mAP50-95 na validační množině v2 a u obou zmíněných algoritmů docházelo k ustálenější saturaci těchto metrik.

5.3.1 Prostor hyperparametrů

V tabulce 18 jsou uvedeny hyperparametry, které byly podrobeny procesu optimalizace. Dále jsou ve zmíněné tabulce uvedeny intervaly hodnot těchto hyperparametrů, jenž byly převzaty z výchozího stavového prostoru pro optimalizaci hyperparametrů od autorů YOLOv8 [49]. Obecně lze optimalizované hyperparametry rozdělit do tří skupin. První skupina (*lr0*, *lrf* a *momentum*, *weight_decay*) souvisí s nastavením optimalizačního algoritmu. Druhá skupina (*warmup_epochs* a *warmup_momentum*) souvisí se stabilizací průběhu tréninku na jeho začátku a třetí skupina (*Box*, *Cls* a *Dfl*) souvisí s chybovou funkcí, respektive s nastavením vah jednotlivých komponent této funkce.

Tabulka 18: Intervaly hodnot optimalizovaných hyperparametrů

Hyperparametr	Interval hodnot
<i>lr0</i>	$\langle 1 \cdot 10^{-5}, 1 \cdot 10^{-1} \rangle$
<i>lrf</i>	$\langle 1 \cdot 10^{-5}, 1 \cdot 10^{-1} \rangle$
<i>momentum</i>	$\langle 0.70, 0.98 \rangle$
<i>weight_decay</i>	$\langle 0, 1 \cdot 10^{-3} \rangle$
<i>warmup_epochs</i>	$\langle 0, 5 \rangle$
<i>warmup_momentum</i>	$\langle 0.00, 0.95 \rangle$
<i>Box</i>	$\langle 1.0, 20.0 \rangle$
<i>Cls</i>	$\langle 0.2, 4.0 \rangle$
<i>Dfl</i>	$\langle 0.4, 6.0 \rangle$

5.3.2 Genetický algoritmus

Genetický algoritmu (GA) je optimalizační metoda na heuristické bázi, jenž se řadí mezi evoluční algoritmy a je inspirována přirozeným výběrem v přírodě [51].

GA implementovaný v knihovně *ultralytics* v podobě třídy *Tuner* probíhá následujícím způsobem. V první iteraci je vytvořen jeden jedinec, což je model s danými hyperparametry. Tento jedinec je podroben ohodnocení, což je proces, při kterém mu je přidělena hodnota

hodnotící funkce (Fitness) na základě které jsou v průběhu GA jedinci porovnáváni [52]. Hodnotící funkce je pak definována takto:

$$Fitness = 0.1 \cdot mAP50 + 0.9 \cdot mAP50_90 \quad (30)$$

Ze vztahu hodnotící funkce je patrné, že klade podstatně větší důraz hodnotu metriky mAP50-95, což je konzervativnější metrika oproti mAP50, jenž se v uvedeném vztahu také vyskytuje.

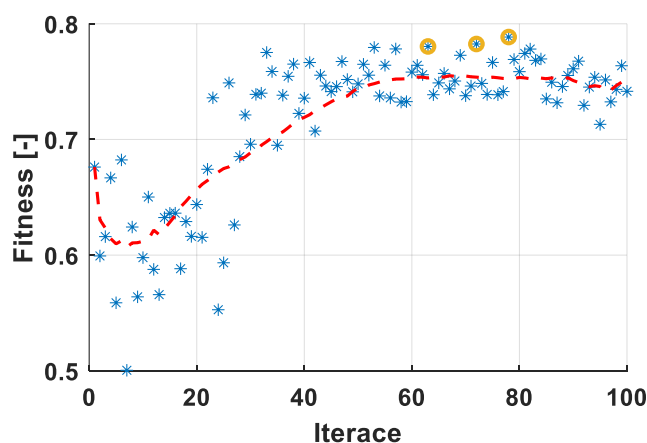
V druhé iteraci GA jsou hyperparametry jedince zmutovány, čímž vznikne nový jedinec, který je podroben ohodnocení. Mutace je proces, při kterém dochází k úpravě hyperparametrů o náhodné číslo, jehož vytvoření se řídí normálním rozdělením se střední hodnotou v nule a rozptylem, který je stanoven pro každý hyperparametr zvlášť [52].

V třetí iteraci GA již vznikla populace dvou jedinců a před mutací dojde k selekci, což je proces, při kterém je populace seřazena dle ohodnocení jednotlivých jedinců a z K nejlepších jedinců je náhodně vybrán jeden, jehož hyperparametry jsou zmutovány a je ohodnocen. V popisovaném GA je hodnota $K=5$, což znamená, že v prvních 5 iteracích vstupuje do selekce celá populace neohledně na pořadí jedinců dle ohodnocení. Princip náhodné selekce z K jedinců je nastaven tak, že jedinci s lepším ohodnocením mají větší pravděpodobnost, že budou vybráni [49].

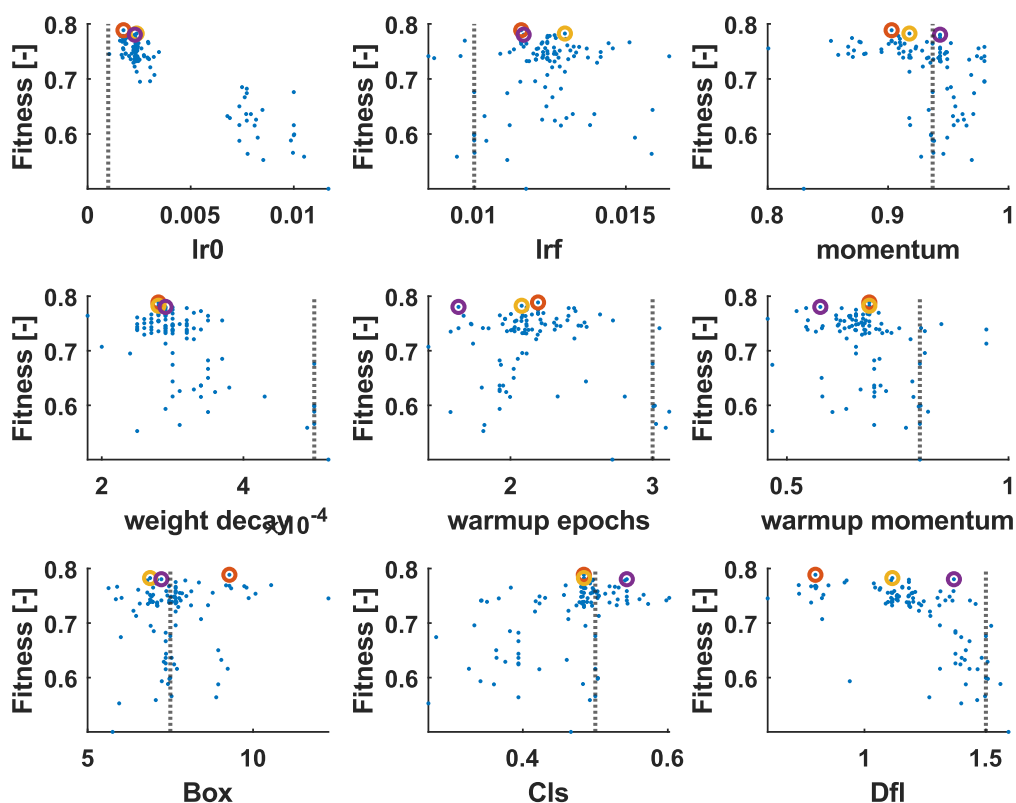
V každé další iteraci GA se tento postup opakuje, dokud není dosaženo daného počtu iterací. Počet iterací odpovídá i konečné velikosti populace, a tedy i výslednému počtu trénovaných modelů.

Genetické algoritmy jsou obvykle implementovány tak, že používají kromě mutace navíc křížení, což je proces, při kterém jsou nové jedince vytvářeny pomocí více než jednoho jedince v původní populaci [51,52]. Z popsaného principu GA použitého v knihovně *ultralytics* je patrné, že algoritmus nepoužívá princip křížení. Tato skutečnost do jisté míry omezuje použitelnost uvedeného GA, nicméně záminka autorů knihovny *ultralytics*, respektive autorů algoritmu YOLOv8, pravděpodobně byla vytvořit variantu GA, jenž prohledává oblast lokálního minima uvedené chybové funkce s přiměřenou výpočetní náročností.

V rámci práce s genetickým algoritmem byl zvolen počet iterací na 100 se suboptimálním počtem 100 epoch pro každý trénink. Jako optimalizační algoritmus byl zvolen algoritmus Adam. Na obrázku 40 je znázorněn průběh hodnotící funkce v závislosti na iteracích. Na zmíněném obrázku jsou žlutě zvýrazněny tři modely, jejichž hyperparametry dosáhly na suboptimální epoše tréninku nejlepších hodnot hodnotící funkce. Tyto modely byly dále



Obrázek 40: Průběh fitness v závislosti na iteraci



Obrázek 41: Vývoj hodnotící funkce v závislosti na jednotlivých hyperparametrech

podrobeny tréninku na 400 epoch k získání průběhu chybové funkce a výkonnostních metrik za účelem, podobně jako u základního modelu, vyhodnotit reprezentativní stav parametrů sítě, který následně vstupoval do porovnání modelů (popsáno v podkapitole 5.4). U všech třech modelů byl jako reprezentativní stav určena epocha, na které se vyskytlo minimum chybové funkce.

Na obrázku 41 jsou znázorněny hodnoty hodnotící funkce v závislosti na hodnotách jednotlivých hyperparametrů. Z uvedených závislostí je patrné, že u většiny hyperparametrů došlo ke shlukování kolem určité hodnoty. Tato hodnota pravděpodobně odpovídá lokálnímu minimu hodnotící funkce, ke kterému měl GA tendenci konvergovat. V každé závislosti je naznačena černá přerušovaná čára, jenž reprezentuje výchozí hodnotu jednotlivých hyperparametrů. Dále jsou v každé závislosti zvýrazněny barevně tři body, jenž reprezentují sady hyperparametrů z třech nejlepších modelů.

5.3.3 Náhodné prohledávání prostoru hyperparametrů

Na základě popsaných omezení použitého GA byla zvolena další poměrně efektivní a jednoduchá metoda prohledávání prostoru hyperparametrů, a to metoda náhodného prohledávání (zkratka RS z anglického „*Random search*“) s cílem prozkoumat větší oblast tohoto prostoru s ohledem na časové a výpočetní možnosti. Tato metoda spočívá ve vytvoření náhodných kombinací hyperparametrů, což je z praktického hlediska výhodné díky snadné implementaci [48]. Jinou jednoduchou a intuitivní metodou prohledávání prostoru hyperparametrů je pravidelné prohledávání (zkratka GS z anglického „*Grid search*“), při níž jsou jednotlivé intervaly hodnot hyperparametrů pravidelně rozděleny a jsou testovány všechny vzniklé možnosti, nicméně v [53] autoři uvádí, že RS je efektivnější oproti GS kvůli výskytu

tzv. kombinatorické exploze. Konkrétně v případě této práce by pravidelné rozdělení intervalů jednotlivých hyperparametrů např. třemi body znamenalo vznik 3^9 možností, což je z výpočetního hlediska extrémně zdlouhavé.

V rámci RS bylo trénováno 60 iterací, kde iterace v tomto případě představuje, stejně jako u GA, jednu sadu hyperparametrů, ale je nutné zdůraznit, že jednotlivé iterace jsou vůči sobě nezávislé na rozdíl od iterací v GA. Daný počet iterací byl zvolen na základě skutečnosti, že při 60 iteracích je pravděpodobnost 95 %, že alespoň jedna sada hyperparametrů náleží do množiny 5 % nejlepších sad hyperparametrů [54]. Pravděpodobnostní rozdělení, jímž se řídilo vytvoření náhodných hodnot hyperparametrů bylo zvoleno jako rovnoměrné. V případě hyperparametrů $lr0$ a lrf bylo použito log-rovnoměrné rozdělení, jelikož se jejich hodnoty typicky liší řádově.

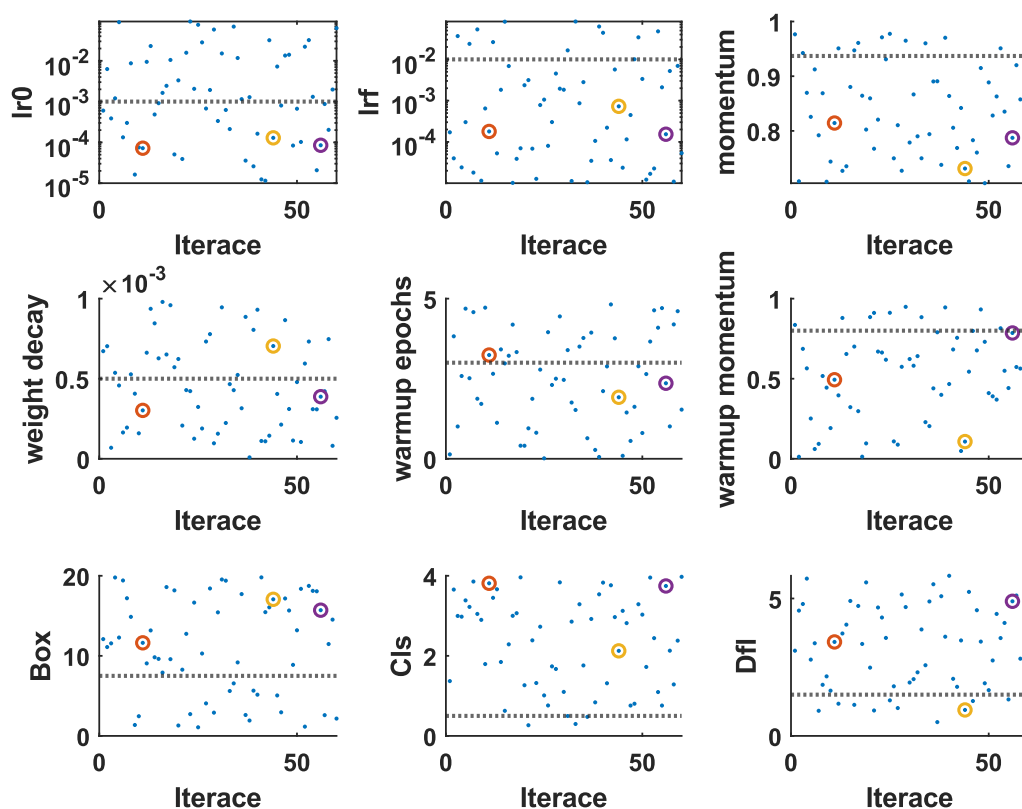
Všech 60 iterací bylo trénováno na suboptimální počet epoch (100) a bylo určeno 15 nejlepších modelů, dle následující hodnotící funkce:

$$Fitness = 0,5 \cdot F_1 + 0,5 \cdot mAP50_{95} \quad (31)$$

kde parametr F_1 (někdy také F-skóre) představuje harmonický průměr přesnosti a citlivosti [21]. Tato forma hodnotící funkce byla navržena tak, aby zohlednila přesnost i citlivost modelů, a nejen metriky na bázi mAP, na rozdíl od hodnotící funkce použité u GA (rovnice 30).

Uvedených 15 nejlepších modelů bylo dále trénováno na 250 epoch pro určení jejich průběhů chybové funkce a výkonnostních metrik. Z těchto 15 modelů byly určeny 3 nejlepší na základě navržené hodnotící funkce v rovnici 31. Je patrné, že bylo použito menší množství epoch pro dotrénování modelů než u GA (400), nicméně během tréninku 15 nejlepších modelů u mnoha z nich nedocházelo k výraznému zlepšení oproti stavu na suboptimální epoše, což napovídá skutečnosti, že tréninky s jejich odpovídajícími hyperparametry konvergovaly k určitým lokálním minimům chybové funkce, jež nebyly z hlediska globálních výsledků na validační množině $v2$ tak úspěšné. Dále některé kombinace hyperparametrů vůbec neminimalizovaly chybovou funkci, což je pochopitelné z hlediska podstaty jejich vytvoření, kdy mohlo dojít k vytvoření kombinace koeficientů optimalizačního algoritmu, jež nebyla schopna minimalizovat chybu a tím pádem se model nebyl schopen učit. Nejlepší 3 modely byly vyhodnoceny z těch modelů, u nichž došlo k výraznému zlepšení mezi suboptimální epochou a 250. epochou tréninku. U těchto tří modelů nedošlo k dalšímu tréninku, jelikož se u žádného z nich na posledních 50 epochách neprojevovalo zlepšení v rámci validačních metrik, což je v kontrastu s hyperparametrem *patience* u základního modelu, jenž byl zvolen na 300 epoch, výrazně kratší interval, nicméně *patience* u základního modelu byla nastavena na tak velkou hodnotu z již uvedených důvodů.

Na obrázku 42 jsou znázorněny jednotlivé hyperparametry v jednotlivých iteracích. Jedná se o podobný obrázek jako obrázek 41 (GA), nicméně u tohoto obrázku nelze mluvit o závislostech, jelikož byly jednotlivé sady hyperparametrů vytvořeny náhodně, což bylo popsáno v předchozím textu. U každého hyperparametru je horizontální přerušovanou čarou znázorněna jeho výchozí hodnota a barevně jsou zvýrazněny 3 nejlepší modely. V jednotlivých grafech lze na hodnotách hyperparametrů nejlepších modelů pozorovat, že některé hyperparametry nabývaly podobných hodnot, respektive jejich hodnoty se pohybovaly v podobné oblasti. Toto lze pozorovat například u $lr0$ (s ohledem na logaritmické měřítko osy y). Na základě toho by bylo možno stanovit vliv jednotlivých hyperparametrů na trénink a výsledky modelu, avšak pro vyvozování závěrů ohledně této problematiky by bylo nutno uskutečnit podrobnější analýzu, jenž nebyla cílem práce, a proto nebyla provedena.



Obrázek 42: Prostor hyperparametrů se zvýrazněnými hyperparametry nejlepších modelů

5.4 Porovnání natrénovaných modelů

Před samotným porovnáním nejlepších modelů je nutné uvést, že na výstupu snímku z detekčního algoritmu byly uvažovány vždy maximálně dvě detekce s největší konfidencí. Toto kritérium vychází z předpokladů stanovených pro třídy rozpoznávaných těles v kapitole 4, kdy se v rámci všech těles mohou na snímku při vstupu do detekčního algoritmu vyskytovat maximálně dvě instance objektů. Toto stanovisko by mohlo mít pozitivní dopad na přesnost modelu, kdy je očekáváno, že model určuje s velkou konfidencí skutečné detekce (TP) a malou, ideálně téměř nulovou konfidencí falešné detekce (FP), které při hojnějším výskytu mohou snížit celkovou přesnost. Na druhou stranu existuje možnost, že zmíněné stanovisko negativně ovlivní citlivost modelu, v případě, že by model určoval s menší konfidencí skutečné detekce oproti falešným.

5.4.1 Porovnání na testovací množině v2

V tabulkách 19 a 20 jsou uvedeny výkonnostní metriky všech modelů vstupujících do srovnání na testovací množině v2. Pro rekapitulaci se jedná o základní model (v tabulce označený jako SGD, dle jeho optimalizačního algoritmu), dva modely z procesu srovnání vlivu optimalizačního algoritmu (označení podobně jako u základního modelu dle příslušných opt. algoritmů tedy Adam a AdamW), tři nejlepší modely z optimalizace pomocí genetického algoritmu (v tabulce označeny jako Tuner s příslušným číslem iterace v průběhu GA) a tři nejlepší modely z optimalizace pomocí náhodného prohledávání prostoru hyperparametrů (v tabulce označeny pomocí zkratky RS a příslušným číslem iterace při tomto prohledávání), což tvoří dohromady devět modelů. Uvedené metriky odpovídají prahu konfidence, ve kterém se

vyskytuje maximum parametru F1 (harmonický průměr přesnosti a citlivosti) v závislosti na nastavené minimální konfidenci detekcí. Uvedený práh konfidence reprezentuje minimální hodnotu konfidence, s níž může model uskutečňovat detekce. Při vysokém prahu konfidence lze očekávat, že model dosáhne vysoké přesnosti při nízké citlivosti, jinými slovy predikuje pouze hraniční obdélníky, se kterými si je jistý. Při nízkém prahu lze očekávat opačný fenomén. Údaj o konfidenci maxima harmonického průměru mezi citlivostí a přesností (F1) je důležitý z hlediska chování modelu a metriky určené na prahu konfidence reprezentují model v jeho nejlepším stavu. Tento práh je pro každý model uveden v tabulce 19 v posledním sloupci a je označen Cf.

Tabulka 19: Výkonnostní metriky z testovací množiny v2

Model	P	P	P	P	R	R	R	R	F1	Cf
	CT-MT	Dogbone	Roxor		CT-MT	Dogbone	Roxor			
AdamW	0.964	0.784	0.968	0.905	0.936	0.975	0.459	0.790	0.844	0.423
Adam	0.958	0.672	0.972	0.868	0.960	0.983	0.535	0.826	0.846	0.486
SGD	0.956	0.751	0.950	0.886	0.888	0.95	0.582	0.807	0.844	0.349
Tuner63	0.937	0.618	0.990	0.849	0.944	0.959	0.508	0.804	0.826	0.265
Tuner72	0.932	0.657	0.957	0.849	0.955	0.967	0.563	0.828	0.838	0.168
Tuner78	0.940	0.688	0.983	0.870	0.965	0.950	0.574	0.830	0.849	0.539
RS11	0.953	0.796	0.928	0.892	0.912	0.992	0.852	0.919	0.905	0.356
RS44	0.978	0.831	0.978	0.929	0.897	0.983	0.786	0.889	0.909	0.643
RS56	0.926	0.870	0.943	0.913	0.919	1.000	0.757	0.892	0.902	0.439

Tabulka 20: Výkonnostní metriky z testovací množiny v2 (pokračování)

Model	AP50	AP50	AP50	mAP50	AP50-95	AP50-95	AP50-95	mAP50-95
	CT-MT	Dogbone	Roxor		CT-MT	Dogbone	Roxor	
AdamW	0.966	0.986	0.805	0.919	0.734	0.930	0.661	0.775
Adam	0.981	0.973	0.848	0.934	0.762	0.909	0.655	0.775
SGD	0.973	0.962	0.797	0.911	0.774	0.908	0.637	0.773
Tuner63	0.956	0.959	0.832	0.916	0.707	0.900	0.648	0.752
Tuner72	0.970	0.971	0.820	0.920	0.746	0.892	0.600	0.746
Tuner78	0.974	0.964	0.865	0.934	0.759	0.903	0.663	0.775
RS11	0.970	0.994	0.951	0.972	0.735	0.932	0.747	0.805
RS44	0.977	0.988	0.940	0.968	0.718	0.928	0.726	0.791
RS56	0.963	0.992	0.937	0.964	0.736	0.940	0.743	0.806

Z uvedených tabulek je patrné, že všechny modely, jenž vstupovaly do porovnání, dosahovaly dobrých výsledků ve všech posuzovaných metrikách. I neoptimalizované modely (označené SGD, Adam a AdamW) dosahovaly obecně slušných výsledků, což souvisí se skutečností o optimalizaci jejich výchozích hodnot hyperparametrů na robustních datasetech, jenž byla naznačena v úvodu této kapitoly. Dále si lze všimnout, že model označený SGD (základní model) si na testovací množině v2 vedl podobně jako modely označené Adam a AdamW. Pro optimalizaci hyperparametrů byly použity právě poslední dva zmíněné modely kvůli jejich průběhům a výsledkům na validační množině v2.

V rámci optimalizovaných modelů pomocí GA nedocházelo k téměř žádnému zlepšení metrik oproti neoptimalizovaným. Jediné výrazné zlepšení spočívalo v průběhu tréninku, respektive v počtu epoch k dosažení minima chybové funkce. Průběhy chybové funkce a metrik těchto modelů nejsou v práci přímo uvedeny, nicméně model označený Tuner63 dosáhl minima chybové funkce v epoše 290, model označený Tuner72 v epoše 226 a model označený Tuner78 v epoše 276. V porovnání s modely označenými Adam a AdamW, jenž dosáhly minima chybové funkce na epochách 618 a 445, je patrné, že došlo k výraznému zrychlení nalezení tohoto minima, což může být zapříčiněno formou hodnotící funkce v kombinaci se suboptimální epochou pro trénink modelů do suboptimálního stavu, jenž byla stanovena na 100. Hodnotící funkce uvedená v rovnici 30 klade největší váhu metrice mAP50-95, jejíž maximum se typicky vyskytuje v oblasti minima chybové funkce, což lze pozorovat na všech doposud uvedených průbězích tréninků a tento fenomén se obecně vyskytoval u všech modelů, jejichž průběhy nejsou přímo uvedeny. Na základě zmíněných okolností je možné, že GA konvergoval k takovým hyperparametrům, jenž byly schopny minimalizovat chybovou funkci v kratším intervalu tréninku, ale z globálního hlediska ke zlepšení metrik nedocházelo.

U nejlepších modelů získaných pomocí RS došlo ke zlepšení všech výkonnostních metrik a zrychlení tréninku, respektive snížení epochy, ve které se objevilo minimum chybové funkce. U modelu s označením RS11 se minimum chybové funkce vyskytlo na epoše 96, u modelu s označením RS44 se vyskytlo na epoše 113 a u modelu s označením RS56 se vyskytlo na epoše 151. Z tabulky 19 je patrné, že k nejvýraznějšímu zlepšení došlo v rámci citlivosti vůči třídě betonářské výztuže, jenž u neoptimalizovaných modelů a modelů optimalizovaných pomocí GA dosahovala znatelně nižších hodnot. Dále je patrné, že se toto zlepšení v citlivosti neprojevalo ve významném zhoršení hodnot přesnosti vůči této třídě a tím pádem celkově došlo ke zvětšení hodnot parametru F1 a obou pokročilých metrik na bázi mAP.

5.4.2 Porovnání na testovací množině v3

V tabulkách 21 a 22 jsou uvedeny výkonnostní metriky všech modelů na testovací množině v3 a stejně jako u porovnání na validační množině v2 jsou uvedené metriky pro práh konfidence s maximální hodnotou parametru F1 (model v nejlepším stavu). Je vhodné připomenout, že testovací množina v3 byla vytvořena pro otestování modelů za hranice stanovených předpokladů na detekované třídy a není do ní zahrnuta třída vrubovaných těles z důvodů popsaných v podkapitole 4.8.

Tabulka 21: Výkonnostní metriky z testovací množiny v3

Model	P	P	P	R	R	R	F1	Cf
	Dogbone	Roxor		Dogbone	Roxor			
AdamW	0.719	0.949	0.834	0.547	0.700	0.623	0.713	0.029
Adam	0.659	0.881	0.770	0.593	0.650	0.621	0.688	0.031
SGD	0.722	0.892	0.807	0.580	0.723	0.652	0.721	0.062
Tuner63	0.774	0.966	0.870	0.587	0.700	0.643	0.740	0.342
Tuner72	0.723	0.889	0.806	0.680	0.588	0.634	0.710	0.071
Tuner78	0.692	0.981	0.836	0.567	0.700	0.633	0.721	0.025
RS11	0.657	0.959	0.808	0.460	0.650	0.555	0.658	0.156
RS44	0.763	0.820	0.791	0.607	0.681	0.644	0.710	0.155
RS56	0.734	0.823	0.779	0.534	0.650	0.592	0.672	0.133

Tabulka 22: Výkonnostní metriky z testovací množiny v3 (pokračování)

Model	AP50	AP50	mAP50	AP50-95	AP50-95	mAP50-95
	Dogbone	Roxor		Dogbone	Roxor	
AdamW	0.661	0.861	0.761	0.517	0.613	0.565
Adam	0.666	0.820	0.743	0.522	0.570	0.546
SGD	0.677	0.848	0.763	0.518	0.588	0.553
Tuner63	0.687	0.836	0.762	0.535	0.595	0.565
Tuner72	0.721	0.785	0.753	0.571	0.523	0.547
Tuner78	0.621	0.861	0.741	0.497	0.609	0.553
RS11	0.548	0.787	0.667	0.428	0.561	0.494
RS44	0.703	0.807	0.755	0.531	0.546	0.539
RS56	0.656	0.757	0.707	0.517	0.527	0.522

Z uvedených tabulek je patrné, že celkově došlo k poklesu všech výkonnostních metrik u všech modelů a na této testovací množině si nejlépe vedl model označený Tuner63, jenž dosáhl největší hodnoty přesnosti, F1 a mAP50-95, nicméně výsledky většiny ostatních modelů nejsou znatelně horší. K znatelnému poklesu ve výkonnostních metrikách došlo u modelů RS11 a RS56, které si naopak na testovací množině v2 vedly průměrně lépe než ostatní modely. Z toho lze usuzovat, že při optimalizaci pomocí RS tyto dvě iterace našly takové hyperparametry, jenž model dostaly do stavu, ve kterém pracuje dobře se snímky splňující stanovené předpoklady použití DIC, avšak není schopen pracovat za stanovená omezení tak dobře jako ostatní modely.

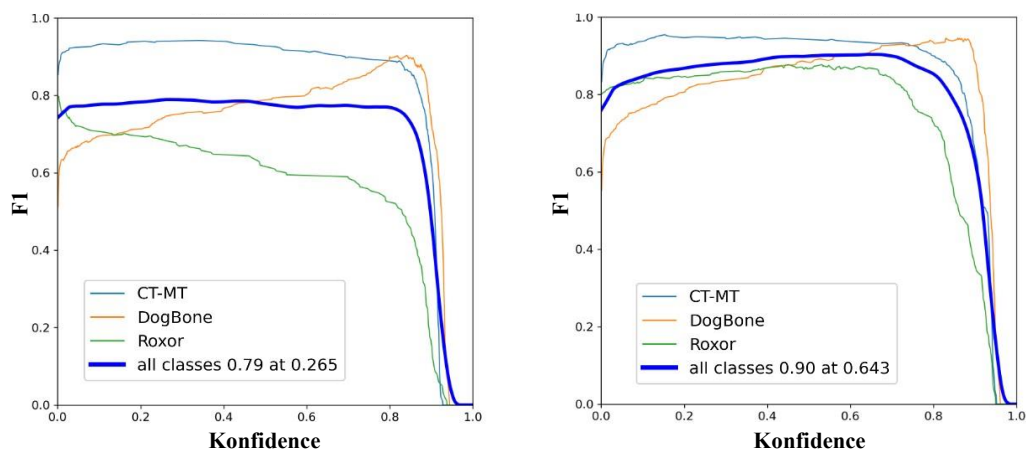
5.5 Volba nejlepšího modelu

Před samotnou volbou nejlepšího modelu je nezbytné podotknout, že je nutno hledět na srovnání modelů nejen z hlediska nejlepšího stavu modelu, respektive modelu se stanoveným prahem konfidence, jenž poskytuje maximum F1 na testovacích datech, ale i z hlediska celkového výkonu modelu při jiném prahu konfidence, protože v praktické aplikaci je možné, že tento práh bude stanoven na nižší nebo na vyšší hodnotu vůči naladění k reálným

vstupům. Na obrázku 43 jsou uvedeny závislosti parametru F1 na minimální hodnotě konfidence pro model Tuner63 (vlevo) a model RS44 (vpravo) na testovací množině v2. U průběhu vlevo je patrné, že celková hodnota parametru F1 (průměr dílčích hodnot tohoto parametru pro všechny třídy) byla téměř konstantní až do konfidence přibližně 0.8, avšak dílčí průběhy tříd se lišily. Pro třídu tahových vzorků (DogBone) F1 roste a maxima nabývá na konfidenzi mezi 0.8 a 0.9, pro třídu betonářské výztuže (Roxor) klesá a maximální hodnota je na extrémně nízkých konfidencích, což může být způsobeno nízkou hodnotou citlivosti tohoto modelu vůči této třídě celkově i v nejlepšímu stavu modelu, jenž je patrná v tabulce 19. Třída vrubovaných těles má průběh nejpodobnější celkovému zprůměrovanému průběhu F1 a dosahuje větších hodnot oproti ostatním třídám. Na průbězích vpravo, jenž odpovídají modelu RS44, je patrné výrazné zlepšení v rámci třídy tahových vzorků a třídy betonářské výztuže, což se projevilo i na výrazném posunutí parametru F1 k větším hodnotám. Dále je patrné, že zprůměrovaný průběh F1 je mírně rostoucí směrem k maximu na prahu konfidence 0.643, nicméně i na nízkých konfidencích dosahuje přes hodnotu 0.8 a na extrémně nízkých konfidencích (do 0.05) padá mírně pod hodnotu 0.8. Průběhy na obrázku 43 jsou uvedeny jako znázornění podrobnějšího porovnání chování modelů na testovací množině v2.

Na základě výsledků modelů na testovacích množinách v2 a v3 (tabulky 19 až 22) a podrobnějšího porovnání chování modelů vůči různým nastavením prahu konfidence naznačeném v předchozím odstavci, byl jako nejlepší model zvolen model RS44. Dalším kandidátem na nejlepší model byl Tuner63, jenž si vedl průměrně nejlépe na testovací množině v3, nicméně s ohledem na skutečnost, že se jedná množinu z velké části nesplňující předpoklady na detekované třídy těles, nebylo k této skutečnosti přihlíženo s velkou vahou.

V tabulce 23 jsou uvedeny hodnoty hyperparametrů modelu RS44 (nejlepšího modelu), jenž byly podrobeny procesu optimalizace. V této tabulce jsou navíc uvedeny pro porovnání výchozí hodnoty hyperparametrů modelu označeného jako AdamW dle jeho opt. algoritmu, jenž byl pro RS44 stejný. Z tabulky je patrné, že se jednotlivé hodnoty zřetelně lišily. Hyperparametry související s opt. algoritmem jsou výrazně menší, především krok učení, jenž je u RS44 oproti AdamW téměř osmkrát menší, a činitel konečného kroku učení (*lrf*), jenž je téměř čtrnáctkrát menší. Dále setrvačnost opt. algoritmu (*momentum*, v tabulce uvedeno jako *mom.*) je podstatně menší a hyperparametr související s penalizací velkých vah (*weight decay*) je nepatrně větší. Z uvedeného lze usuzovat, že nastavení opt. algoritmu u RS44 bylo oproti



Obrázek 43: Porovnání průběhu parametru F1 v závislosti prahu konfidence modelu Tuner63 (vlevo) a modelu RS44 (vpravo)

modelu AdamW jemnější ve smyslu menší úpravy trénovatelných parametrů sítě a celkově menší setrvačnosti algoritmu. Co se týče rozehrívací fáze spojené s hyperparametry *warmup epochs* a *warmup momentum* (v tabulce slovo *warmup* zkráceno na *w.*) je patrné, že oba hyperparametry dosahovaly menších hodnot, a to především setrvačnost v rozehrívací fázi. Z toho lze usuzovat, že jsou tyto hyperparametry z hlediska hledání globálního minima nepodstatné nebo rozehrívací fáze byla u modelu AdamW příliš pozvolná, nicméně na vyvození podrobnějších závěrů by bylo nutno podrobit tyto vlivy podrobnější analýze. Co se týče nastavení vah chybové funkce (*Box*, *Cls* a *Dfl*), tak je zřejmé, že byl kladen větší důraz na člen související s ClIoU chybovou funkcí (*Box*), jež zodpovídá za určení chyby lokalizace hraničního obdélníku, a na člen související s penalizací klasifikace jednotlivých tříd (*Cls*). Naopak byl kladen menší důraz na člen související s penalizací predikovaného rozdělení offsetů skutečných od predikovaných hraničních obdélníků, což je pochopitelné v souladu s významností této části chybové funkce v kontextu hustě pokrytých snímků objekty, jež nejsou typické z pohledu vytvořeného datasetu a obecně z pohledu řešené úlohy. Z porovnání poměru *Box:Cls*, jež byl u modelu AdamW 30:2 a u modelu RS44 přibližně 17:2, je patrné, že byl zvětšen důraz na chybu klasifikace v rámci detekce.

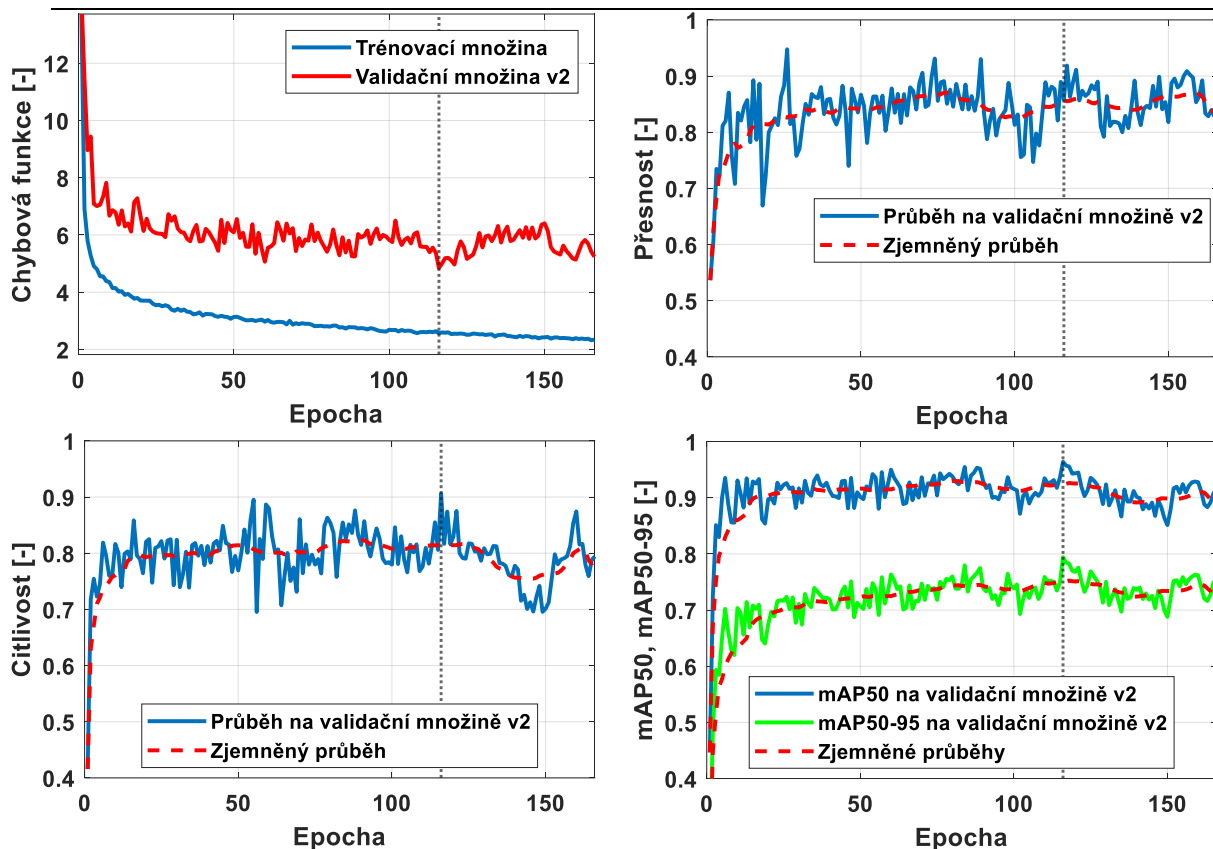
Tabulka 23: Porovnání hyperparametrů neoptimalizovaného a optimalizovaného modelu

Model	<i>lr0</i>	<i>lrf</i>	<i>mom.</i>	<i>weight decay</i>	<i>w. epochs</i>	<i>w. mom.</i>	<i>Box</i>	<i>Cls</i>	<i>Dfl</i>
AdamW	0.00100	0.0100	0.937	0.0005	3.0	0.800	7.5	0.5	1.5
RS44	0.00013	0.0007	0.731	0.0007	1.9	0.107	17.0	2.1	0.9

Na obrázku 44 jsou uvedeny průběhy chybové funkce, přesnosti, citlivosti a metrik na bázi mAP modelu RS44 (nejlepšího modelu) na validační množině v2. V každém průběhu je přerušovanou čarou naznačena epocha, v níž se vyskytlo minimum chybové funkce, a která zároveň odpovídá reprezentativnímu stavu modelu. Z uvedených průběhů je patrné, že se v nich vyskytovala oscilace podobně jako u předchozích modelů (obrázky 36 až 39) a lze očekávat, že při pokračování tréninku by také došlo k její vyhlazení podobně jako u předchozích modelů, avšak z uvedených průběhů, a to především průběhy metrik mAP50 a mAP50-95, je patrná výrazně rychlejší saturace, což napovídá skutečnosti, že by model mohl dosáhnout stavu přetrénování blízko od přerušování tréninku.

5.6 Vyhodnocení nejlepšího modelu

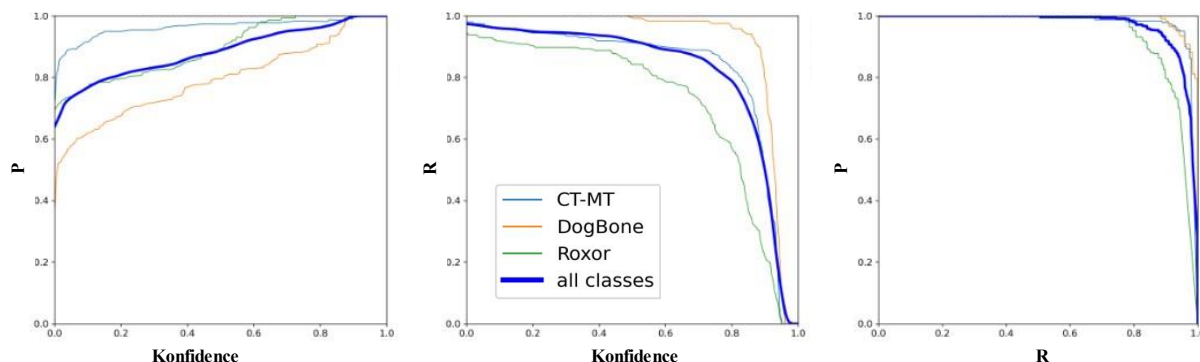
Na obrázku 45 jsou uvedeny závislosti přesnosti na prahu konfidence (vlevo), citlivosti na prahu konfidence (uprostřed) a P-R křivka (přesnost v závislosti na citlivosti modelu, jejíž plocha pod křivkou reprezentuje metriku mAP50 a je uvedena na obrázku vpravo) odvozené z testovací množiny v2 pro RS44 (nejlepší model). Ze závislosti přesnosti na prahu konfidence je patrné, že při naladění modelu na nízkou hodnotu tohoto prahu lze očekávat pokles přesnosti modelu vůči všem třídám, a primárně vůči třídě tahových vzorků, nicméně stále je nutné zohlednit skutečnost, že pro tuto třídu a třídu betonářské výztuže se pro splnění stanovených předpokladů může na vstupu do detekčního algoritmu objevit pouze jedna instance těchto tříd, která má na výstupu v ideálním případě největší konfidence ze všech možných detekcí i včetně



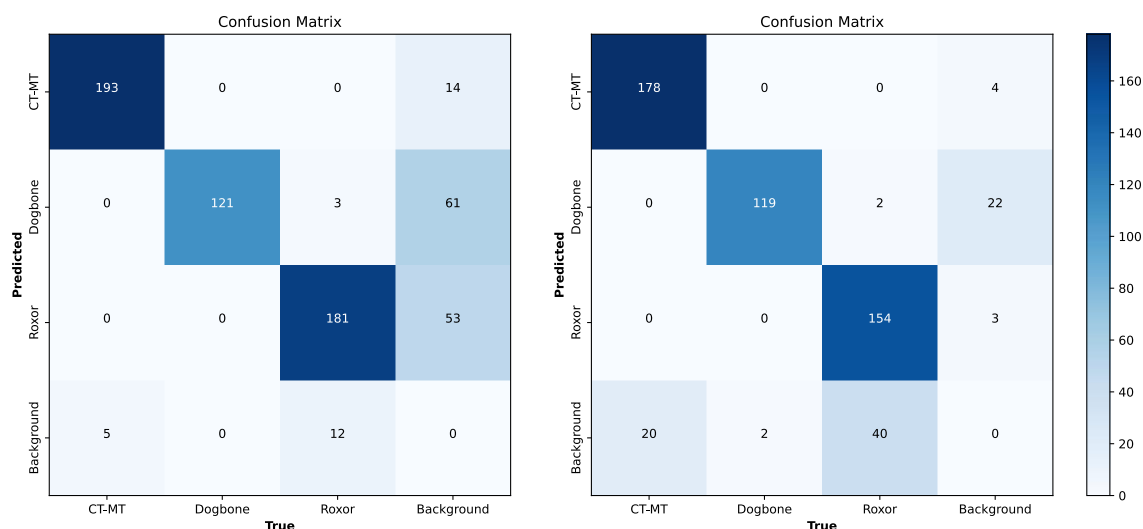
Obrázek 44: Průběh chybové funkce a všech metrik na validační množině v2 modelu RS44

falešných (FP). Navíc falešná detekce může být při nastavení na maximální počet dvou detekcí na výstupu pouze jedna za předpokladu, že druhá detekce bude skutečná (TP). Tento fakt je částečně podložen i vývojem citlivosti modelu, kdy je patrné, že citlivost dosahuje na nízkých konfidencích velmi vysokých hodnot, avšak je stále nutné brát v potaz, že se jedná pouze o skutečnosti odvozené na základě testovací množiny v2. V praxi může být chování modelu vůči vstupům odlišné, nicméně je předpokládáno, že při dodržení stanovených předpokladů bude podobné. Dále lze na průbězích přesnosti a citlivosti dobře pozorovat „trade-off“ mezi těmito metrikami, jenž byl naznačen již v předchozím textu a lze ho volně interpretovat jako nemožnost dosažení maximální citlivosti spolu s dosažením maximální přesnosti a jenž není tak dobře pozorovatelný ze samotného průběhu parametru F1 (obrázek 43 vpravo) [55].

Na obrázku 46 jsou uvedeny dvě matice záměn („Confusion matrix“) modelu RS44 na testovací množině v2. Vlevo je tato matice uvedena pro práh konfidence 0.100 a vpravo pro



Obrázek 45: Závislost přesnosti na prahu konfidence (vlevo), citlivosti na prahu konfidence (uprostřed) a P-R křivka (vpravo)



Obrázek 46: Matice záměn (vlevo práh konfidence 0.100, vpravo práh konfidence 0.643)

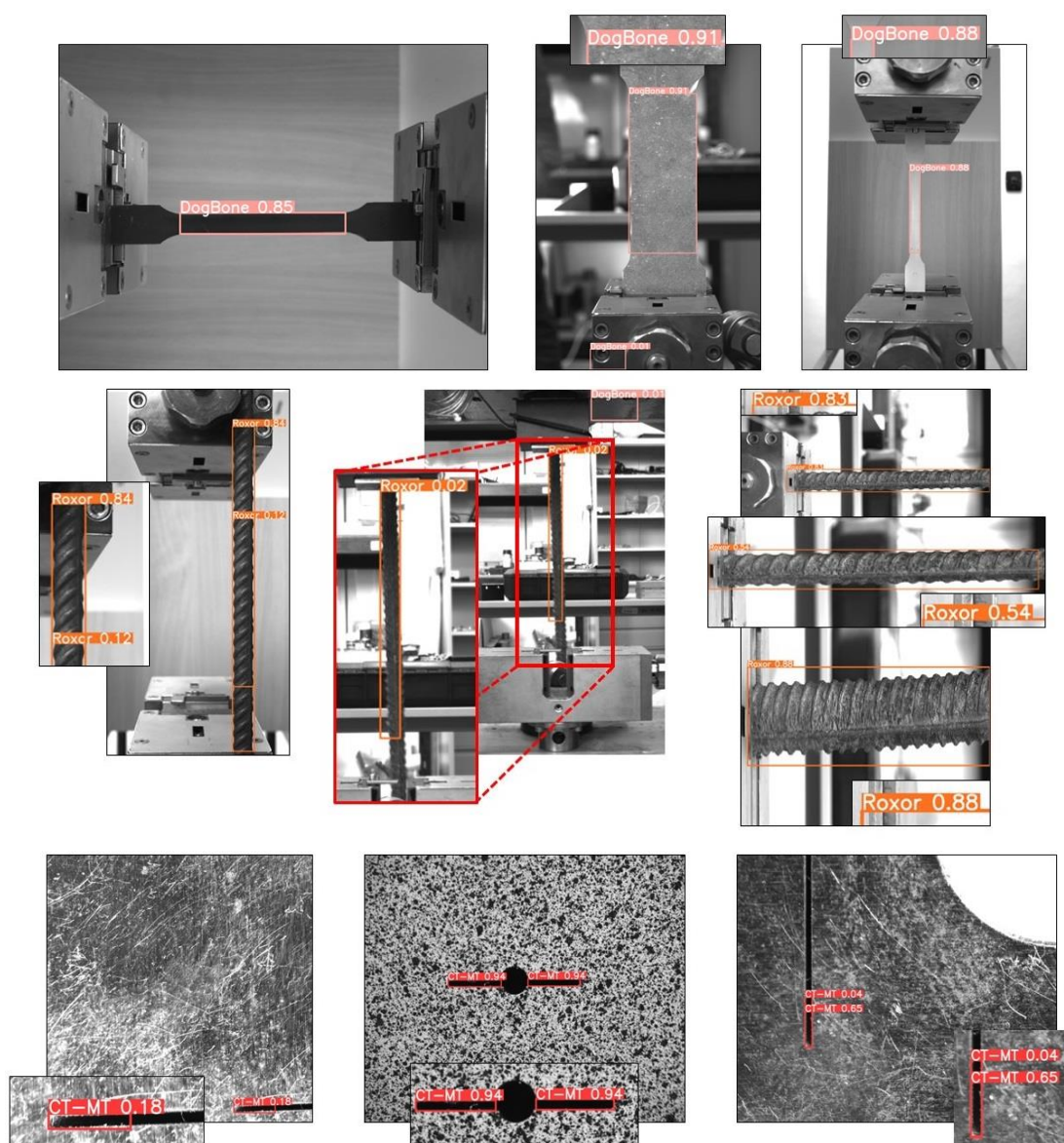
optimální práh konfidence, jenž byl pro RS44 na této testovací množině 0.643. Z obrázku vlevo je patrný nárůst falešných detekcí, a to především u třídy betonářské výztuže, nicméně je patrný i nárůst celkových pravých detekcí (TP), což koresponduje s průběhy na obrázku 45. Dále je patrné, že k záměně tříd, respektive predikování správného hraničního obdélníku se špatnou třídní příslušností docházelo na obou prazích konfidence ojedinele a pouze mezi třídou betonářské výztuže a třídou tahových vzorků, což je pochopitelné, protože jsou z vizuálního hlediska příbuznější, než obě třídy vůči třídě vrubovaných těles.

Na obrázku 47 jsou uvedeny příklady snímků z testovací množiny v2 na výstupu z modelu. Na každém snímku je hraniční obdélník reprezentující detekovanou oblast, třída příslušnosti podle pracovního označení a příslušná konfidence. Práh konfidence byl pro jednotlivé predikce nastaven na 0.01. Celkem je uvedeno 9 snímků, kde horní 3 obsahují tahové vzorky, prostřední 3 obsahují pruty betonářské výztuže a dolní 3 obsahují instance konců vrubů vrubovaných těles, respektive instance třídy vrubovaných těles.

Na horní trojici snímků obrázku 47 lze pozorovat poměrně vysoké konfidence detekovaných tahových vzorků, což se ukázalo jako typické pro tuto třídu, nicméně určitým omezením byla skutečnost častých falešných detekcí, jež se tvořily v pozadí snímků, což je patrné například na prostředním snímku v horní řadě v jeho dolní části nebo prostředním snímku v prostřední řadě vpravo nahoře. Tyto falešné detekce (FP) mají sice malou konfidence, nicméně podobné FP měl model tendenci často vytvářet i s většími konfidencemi, které byly v některých případech v řádech desítek procent (např. obrázek 49 druhá řada uprostřed).

Na prostřední trojici snímků obrázku 49 vlevo je naznačen poměrně častý jev chování modelu vůči třídě betonářské výztuže, při němž model vytvořil místo jednoho 2 hraniční obdélníky na jednom prutu. Toto chování bylo často zapříčiněnou určitou lokální změnou v rámci osy prutu, jenž může být např. povrchová nedokonalost, případně stín vlivem nevhodného osvětlení nebo neúplná vertikální či horizontální orientace. Tento jev lze potlačit snížením prahu IoU pro NMS (Non-max suppression), jenž eliminuje duplikáty hraničních obdélníků na jednom objektu. Hodnota tohoto prahu by mohla být teoreticky nastavena na extrémně nízké číslo, protože není očekáváno, že se na jednom obrázku budou vyskytovat více než 2 instance tříd objektů navíc blízko u sebe (stanovené předpoklady), a proto je v rámci detekce důležitý vždy hraniční objekt s maximální konfidencí, nicméně tento jev lze potlačit i

poměrně jednoduchou implementací podmínek pro práci s holými výstupy z modelu (popsáno v kapitole 6). Dále je v prostřední trojici snímků uprostřed uveden příklad velice nízké konfidence a neúplného hraničního obdélníku, což lze vysvětlit tím, že při pořizování tohoto snímku došlo k pohnutí s prutem a tím i k mírnému rozmazání, což do značné míry snižuje konfidenci modelu. Navíc je patrné, že se jedná o neúplně vertikálně zarovnaný prut, jenž je tenký, což je z hlediska detekce pro model obtížné. Na pravém snímku v prostřední řadě je nahoře uveden jeden snímek a pod ním jeho dvě úpravy. Z horního snímku je patrné, že pro model nebyla jeho detekce obtížná. Uprostřed je uvedena oříznutá varianta tohoto snímku tak, aby byl vidět pouze prut betonářské výztuže, což konfidenci modelu snížilo, a to z toho důvodu, že v dostupné databázi byly obvykle podobné snímky (kde v nich prut zabírá většinu snímku) s velkým nepoměrem stran (uvedeno v kapitole 4, kde byl i naznačen postup úpravy těchto snímků, aby byl jejich vliv omezen). U dolního snímku, který byl vytvořen z oříznutého pomocí změny velikosti na výchozí velikost pro vstup do modelu (960x960), kde se projevuje vizuální zkreslení (lze interpretovat jako fiktivní stlačení v ose prutu), naopak konfidence vzrostla, což



Obrázek 47: Ukázky detekcí na testovací množině v2

lze odvodit již zmíněnou skutečností charakteru snímků z dostupné databáze v trénovací množině.

V dolní řadě na obrázku 47 vlevo je uveden snímek, kde došlo k správné detekci konce vrubu, nicméně s nižší konfidencí, což reprezentuje určitý jev snižování konfidence v závislosti vertikálním či horizontálním nezarovnání vrubu vůči snímku. V dolní řadě uprostřed jsou uvedeny detekce na MT tělese a vpravo přizpůsobení hraničního obdélníku vůči konci vrubu. Na snímku vpravo je patrná další detekce stejného konce vrubu s nižší konfidencí, což souvisí s podobným jevem popsaným v rámci třídy vrubovaných těles, jehož vliv lze omezit stejným způsobem jako byl již popsán (snížení IoU prahu NMS, případně implementací podmínek na výstupy z modelu). Snímek vpravo dále přímo nesplňuje předpoklad vznesený na třídu vrubovaných těles z toho důvodu, že obsahuje i část tělesa mimo definovanou oblast (část díry vzorku).

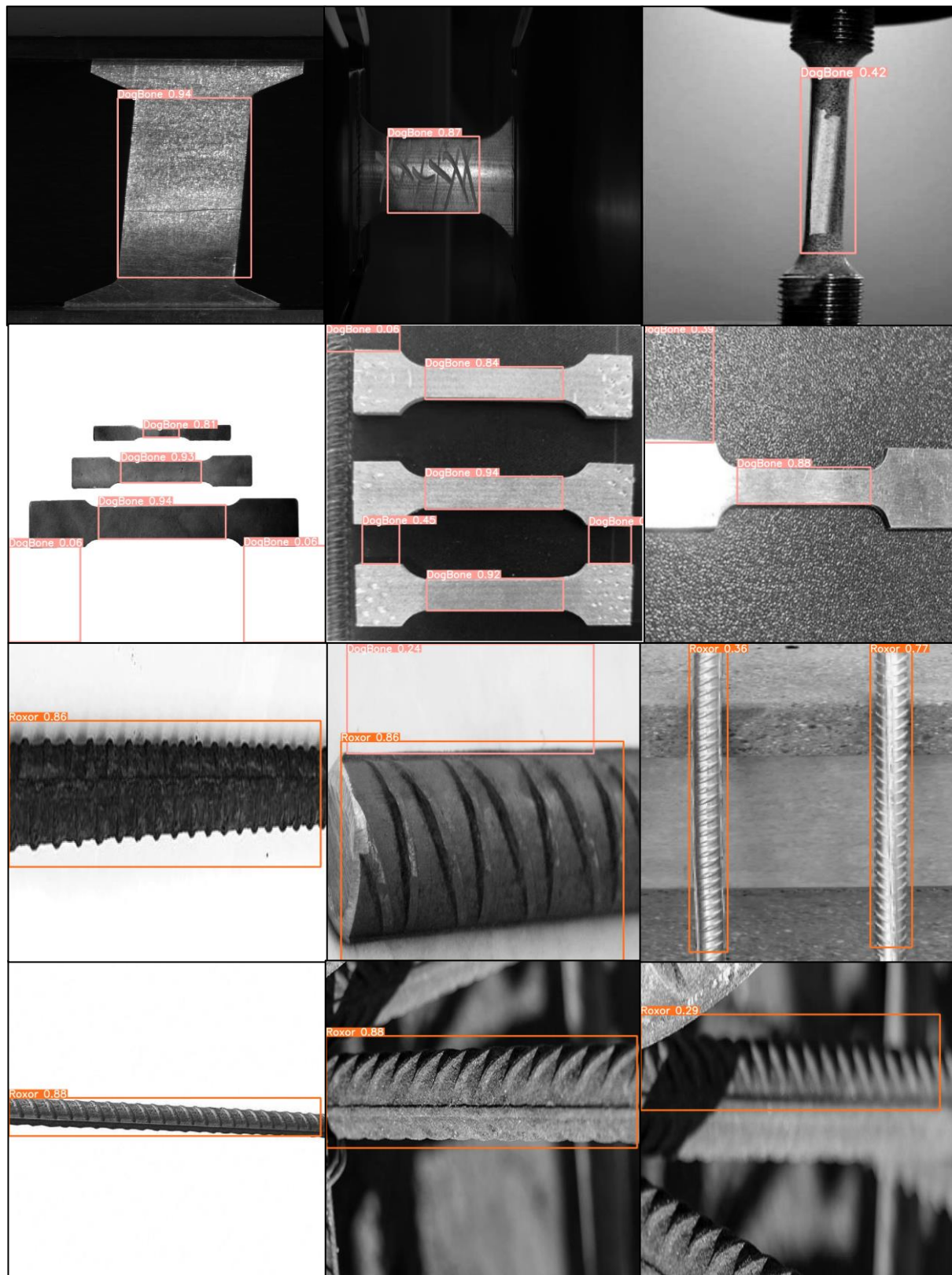
Na obrázku 48 je uvedeno několik ukázek detekcí na snímcích z testovací množiny v3. Z většiny uvedených snímků je patrné, že se jedná o snímky nesplňující stanovené předpoklady na detekované třídy, pro tyto predikce (detekce) byl nastaven práh konfidence na 0.01 a nebylo vzneseno omezení na maximální počet predikovaných hraničních obdélníků. U některých snímků je patrné, že se jedná o snímky v netypickém kontextu z hlediska použití DIC (např. dolní řada vpravo) nebo o umělé snímky bez pozadí (např. dolní řada vlevo).

U mnoha snímků je patrné, že se projevuje jev, jenž byl popsán na ukázce detekcí z testovací množiny v2, a to výskyt FP tahových vzorků v pozadí snímků. Tento jev je dobře pozorovatelný na všech snímcích druhé řady a na prostředním snímku třetí řady. Vysvětlení tohoto jevu patrně souvisí se sémantickou podstatou. Model se v procesu učení učil tato tělesa, respektive anotovanou část těchto těles, rozpoznávat primárně pomocí typického tvaru, jenž je ze sémantického hlediska velmi jednoduchý a při pozorném pohledu například na obrázek v druhé řadě uprostřed si lze všimnout, že se tento tvar často vyskytuje v oblasti umístění falešného hraničního obdélníku (FP). Avšak lze konstatovat, že riziko výskytu podobné situace je značně umocněno skutečností, že se často jedná o umělé snímky s typicky znatelně nižším rozlišením než snímky pro DIC. Skutečné snímky jsou navíc kontextuálně složitější (snímky z testovací množiny v2 na obrázku 47), nicméně je tento jev nutné brát v potaz při práci s modelem a jeho ladění pro konkrétní aplikaci.

Dále je na obrázku 48 uvedeno chování modelu vůči nezarovnaným instancím objektů (dolní řada vlevo), výskytu znatelné perspektivy (předposlední řada vlevo) a snímkům v netypickém kontextu (předposlední řada uprostřed a vpravo a dolní řada uprostřed a vpravo). U všech uvedených snímků, na kterých se tyto jevy vyskytovaly, je patrné, že model provedl příslušné detekce, avšak obecně lze říct, že při jejich výskytu se model chová nepředvídatelně, často klesá konfidence detekce a přesnost lokalizace hraničního obdélníku. Takové snímky, respektive instance, jsou často vyhodnoceny jako falešný negativ (FN), což není na obrázku 48 přímo znázorněno, nicméně to lze odvodit z tabulky 21, ve které je patrné, že citlivost všech modelů se na testovací množině v3 pohybovala kolem 60 %.

Lze předpokládat, že kdyby v procesu augmentace (první nebo druhé fáze) byly použity takové augmentační techniky, jež by do trénovací množiny zanesly podobné snímky, tak by s nimi model pracoval lépe, nicméně v kontrastu uvedeného tvrzení je vhodné opět připomenout, že se jedná o model, jenž byl vytvořen za účelem použití pro záznamy při testování materiálů pomocí DIC, která typicky s takovými snímky nepracuje. Navíc s ohledem na několikrát zmíněnou sémantickou jednoduchost detekovaných těles lze předpokládat, že by

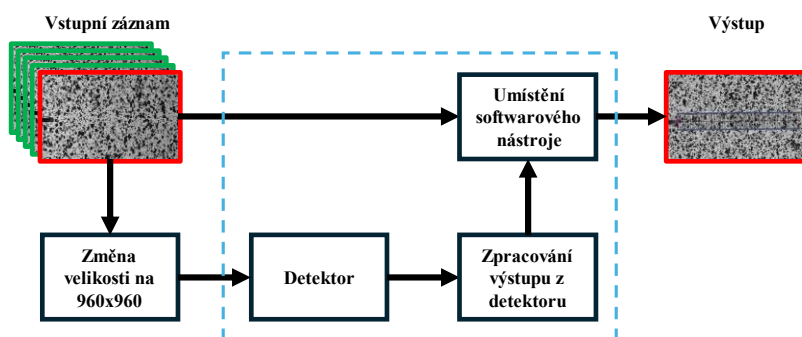
model mohl mít větší tendenci vytvářet falešné detekce na strukturách v pozadí, které by se mohly z tohoto hlediska více přiblížit k augmentovaným snímkům, což souvisí s popsáním jevem častého výskytu FP v pozadí u třídy tahových vzorků.



Obrázek 48: Ukázky detekcí na testovací množině v3

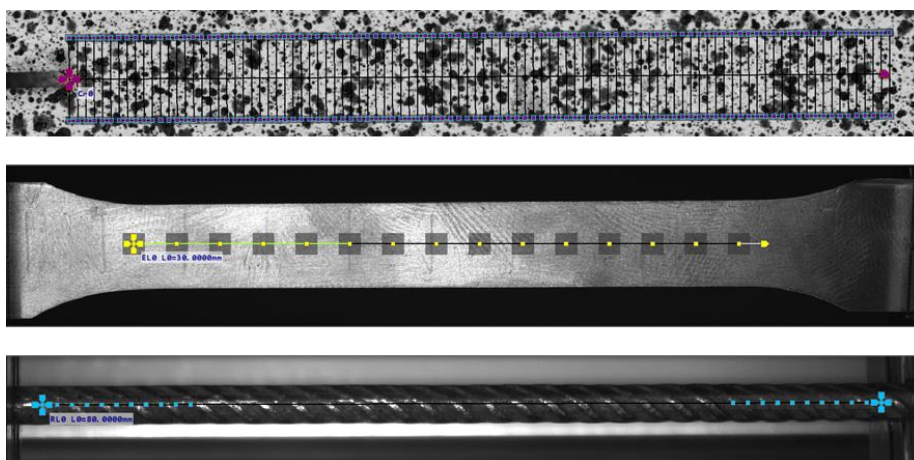
6 Práce s modelem

V předchozí kapitole je popsán celý proces implementace a testování, jenž vedl k volbě nejlepšího modelu označeného RS44. Tato kapitola se zabývá prací s tímto modelem jako detektorem integrovaným do softwaru k vyhodnocení záznamů pořízených pomocí videoextenzometrů. Účel tohoto detektoru byl vytvořit základ pro umístění vhodného softwarového nástroje do správné polohy na vzorek v reálném čase. Na obrázku 49 je znázorněno schéma tohoto procesu, kde do příslušného softwaru vstupuje záznam pořízený pomocí videoextenzometru, jehož první snímek vstupuje do detektoru, který určí typ vzorku a souřadnice hraničního obdélníku, pomocí nichž je umístěn softwarový nástroj vybraný dle typu vzorku. V uvedeném schématu jsou další dva další dílčí kroky zpracování, a to změna velikosti snímku před vstupem do detektoru, jelikož byl tento detektor přizpůsoben na snímky o velikosti 960x960 (viz. kapitola 5) a zpracování výstupu z detektoru, jenž je primárním předmětem této kapitoly.



Obrázek 49: Schéma práce s detektorem

Na základě jednotlivých tříd vzorků byly určeny tři typy softwarových nástrojů, jenž se používají v softwaru Alpha poskytovaném společností X-Sight, jejíž databáze byla základem pro implementaci detektoru (modelu). Prvním typem byl CRACK PROBE, což je nástroj pro měření délky trhliny při statickém nebo dynamickém testování na vrubovaných vzorcích [56]. Druhým typem byly nástroje LINE, EXTREME LINE a TRANS LINE, jenž se používají na tahových vzorcích pro testování materiálových vlastností [56]. Třetím typem byl nástroj ROD LINE, jenž se používá u měření na prutech betonářské výztuže a je navržen tak, aby



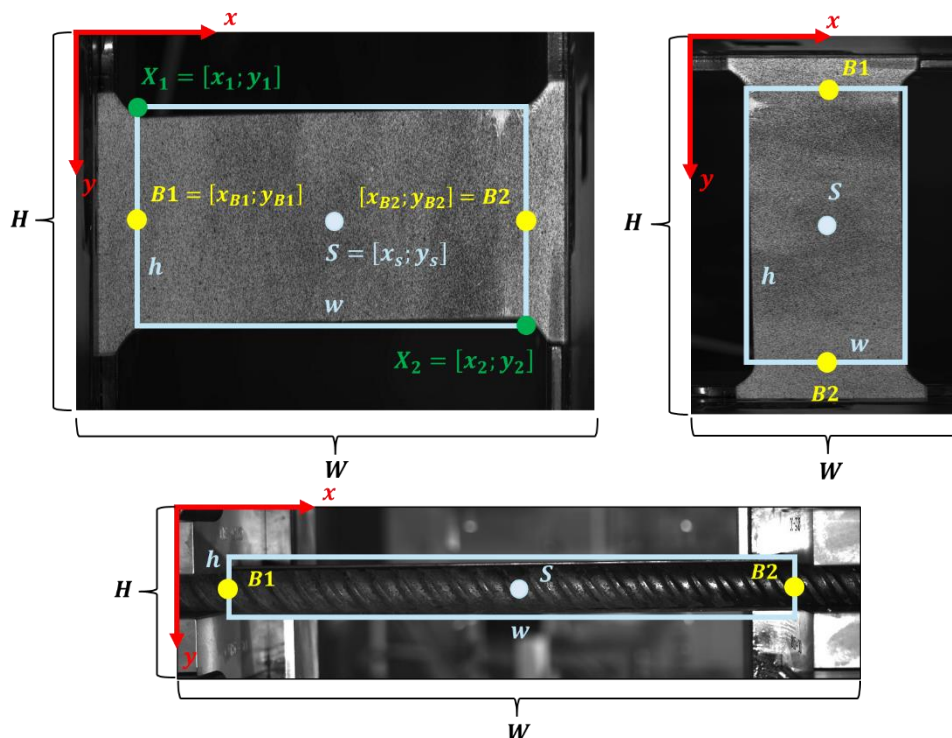
Obrázek 50: CRACK PROBE (nahore), EXTREME LINE (uprostřed) a ROD LINE (dole)

kompenzoval odpadávání zoxidované vrstvy, jenž se typicky na těchto prutech vyskytuje [56]. Uvedené tři typy softwarových nástrojů jsou znázorněny na obrázku 50.

6.1 Zpracování výstupu z detektoru

Vizuální reprezentace výstupu z detektoru je znázorněna například na obrázcích 47 a 48 v předchozí kapitole. Matematická reprezentace hraničního obdélníku jako výstupu z detektoru může být dvojice bodů jeho opačných rohů (obrázku 51 vlevo zeleně) nebo bod udávající střed toho obdélníku s jeho šířkou a výškou (obrázek 51 zvýrazněna světle modrou). Obě uvedené formy mohou být normalizovány vůči rozměrům vstupního snímku.

Typ softwarového nástroje lze vyhodnotit dle třídy vzorku, která je spolu s konfidencí připojena k souřadnicím hraničního obdélníku. V předchozí kapitole bylo popsáno chování detektoru z výsledků na testovacích množinách v2 a v3, bylo pojednáno o stanovení minimální konfidence s jakou model (detektor) mohl provádět detekce (predikce) a dále bylo pojednáno o maximální počtu relevantních výstupů z detektoru, jenž vychází ze stanovených předpokladů na třídy detekovaných objektů (kapitola 4). Na výstupu z detektoru může být obecně N predikcí, jejichž konfidence přesahuje stanovený práh, nicméně na základě popsanych skutečností jsou významné pouze dvě s největší konfidencí. Pomocí konfidence těchto dvou predikcí je možné vyhodnotit o jaký typ vzorku se jedná. V tabulce 24 je uveden přehled možností, které se mohou na výstupu z detektoru vyskytnout a je k nim uvedeno jejich vyhodnocení v rámci zpracování výstupu z detektoru. V uvedených tabulkách sloupec 1. pred. reprezentuje predikci s největší konfidencí a sloupec 2. pred. s druhou největší konfidencí.



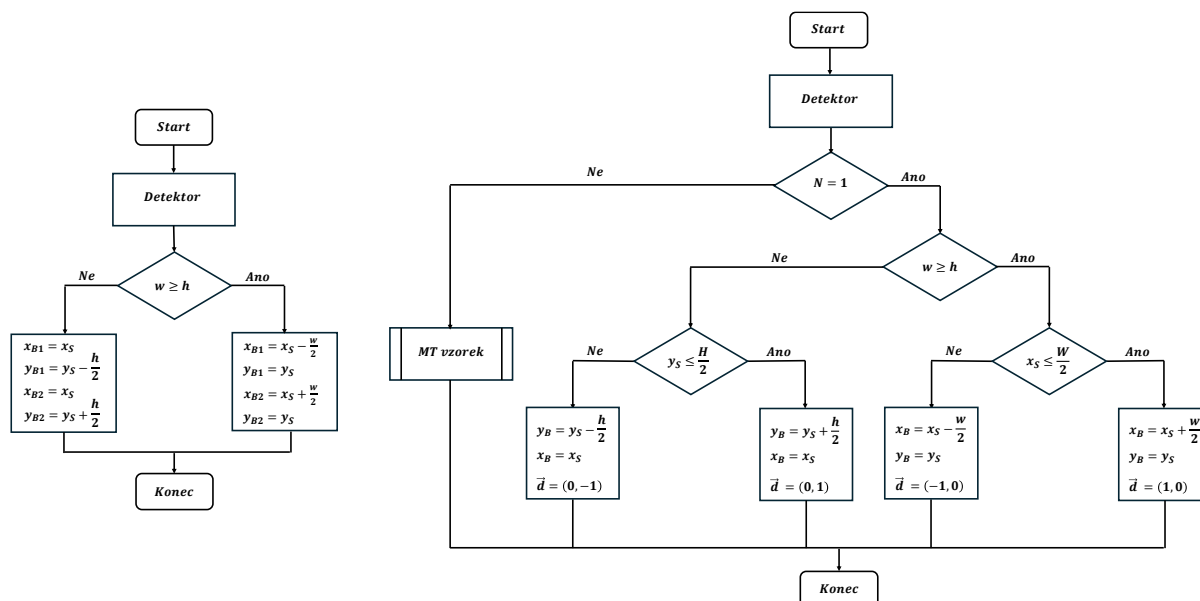
Obrázek 51: Popis hraničního obdélníku jako výstupu z detektoru a podstatné body z hlediska určení polohy sw. nástroje

Tabulka 24: Vyhodnocení výstupu z modelu (třída vrubovaných těles je označena 0, třída tahových vzorků označena 1 a třída betonářské výztuže označena 2)

1. pred.	2. pred.	Vyhodnocení (uvedená trojčíslí odkazují na podkapitoly)
0	Není	Sw. nástroj je umístěn dle 6.1.3
1	Není	Sw. nástroj je umístěn dle 6.1.1
2	Není	Sw. nástroj je umístěn dle 6.1.2
0	0	Pravděpodobně se jedná o MT těleso. Sw. je umístěn dle 6.1.3.
1	1	2. predikce je pravděpodobně FP. Umístění sw. nástroje je provedeno na základě 1. predikce dle 6.1.1.
2	2	2. predikce je pravděpodobně FP nebo se jedná o dvě predikce na jednom prutu (viz. podkapitola 5.6). Sw je umístěn na základě 1. predikce dle 6.1.2.
0	1 nebo 2	2. predikce je pravděpodobně FP. Sw. nástroj je umístěn na základě 1. predikce dle 6.1.3
1 nebo 2	0	2. predikce je pravděpodobně FP. K umístění sw. nástroje je použita pouze 1. predikce dle 6.1.1 nebo 6.1.2
1	2	2. predikce je pravděpodobně FP, ale může se jednat o záměnu tříd, nicméně s ohledem na chování modelu v matici záměn (obrázek 46) je to nepravděpodobné. Sw. nástroj je umístěn na základě 1. predikce dle 6.1.1.
2	1	Stejně jako v předchozím řádku. Sw. nástroj umístěn na základě 1. predikce dle 6.1.2.

6.1.1 Třída tahových vzorků

Pro vyhodnocení pozice příslušných softwarových nástrojů byly u třídy tahových vzorků podstatné dva body odvozené ze souřadnic hraničního obdélníku. Tyto body jsou naznačeny na horních dvou snímcích na obrázku 51 žlutě a představují ohrazení umístění nástrojů. Snímky na vstupu do detektoru mohou mít obecně vzorek zarovnaný vertikálně nebo horizontálně (na obrázku 52 horizontálně vlevo a vertikálně vpravo), což souvisí se správou orientací nástroje vůči vzorku. Tato orientace lze odvodit z poměru výšky ku šířce predikovaného hraničního obdélníku za předpokladu, že se jedná o vzorek jehož délka je větší než jeho šířka, což je pro tahové vzorky typické. Je nutno připomenout, že do detektoru vstupují snímky o velikosti 960x960 a ze samotných výstupů z detektoru může dojít k nekorektnímu určení orientace vlivem změny velikosti. Z toho důvodu je nutné tuto orientaci vždy vyhodnotit na základě výstupních souřadnic přepočítaných do původní velikosti snímku. Určení bodů popisujících ohrazení softwarového nástroje je uvedeno ve vývojovém diagramu znázorněném na obrázku 52 vlevo. V uvedeném vývojovém diagramu (podobně jako i v ostatních) je znázorněn proces popsáný jako detektor, jenž reprezentuje průchod snímku detektorem, na jehož výstupu jsou všechny údaje potřebné pro další zpracování dle vyhodnocení predikcí v tabulce 24. Dále je vhodné doplnit, že určení bodů *B1* a *B2* koresponduje s uvedeným obrázkem 52 nahoře, tedy bod *B1* je při horizontálním zarovnání vzorku vlevo a při vertikálním zarovnání blíže k ose *x* znázorněného souřadného systému (blíže k horní straně snímku).



Obrázek 52: Vývojové diagramy určení polohy sw. nástroje (levý pro třídu tahových vzorků a třídu betonářské výztuže, pravý pro třídu vrubovaných těles)

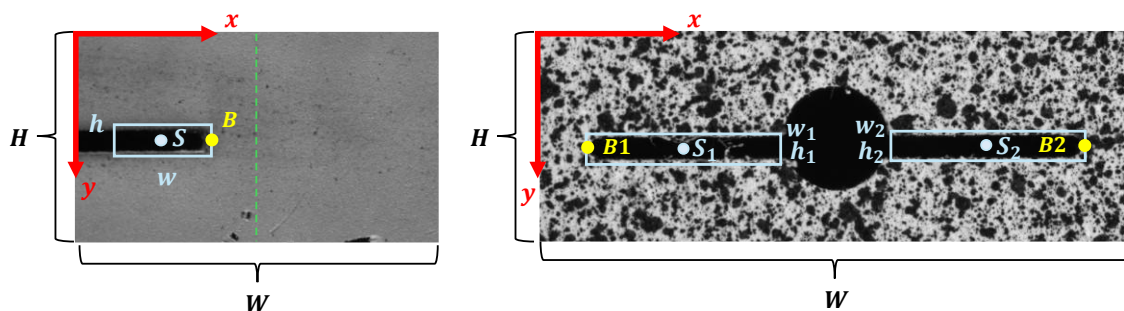
6.1.2 Třída betonářské výztuže

Vyhodnocení pozice příslušného softwarového nástroje bylo u třídy betonářské výztuže velice podobné jako u třídy tahových vzorků a platí pro něj stejné předpoklady, jež jsou popsány v předchozí podkapitole. Na obrázku 52 dole je znázorněn snímek obsahující prut betonářské výztuže spolu s hraniční obdélníkem a se žlutě zvýrazněnými body, které ohraničují příslušný softwarový nástroj v horizontálním zarovnání prutu (při vertikálním zarovnání označení podstatných bodů koresponduje s označením u třídy tahových vzorků). Určení těchto bodů vychází ze stejného principu jako u tahových vzorků, proto je uvedeno na stejném vývojovém diagramu na obrázku 52 vlevo.

6.1.3 Třída vrubovaných těles

Pro vyhodnocení pozice příslušného softwarového nástroje byl u třídy vrubovaných těles podstatný bod, jenž leží na hraně ohraničující konec vrubu (naznačeno na obrázku 53 pro obecné těleso vlevo a MT těleso vpravo), a dále směr vrubu pro určení správného směru umístění nástroje na vzorek. U této třídy docházelo ke zkomplikování situace vlivem možné přítomnosti dvou instancí u MT tělesa, proto bylo nutné definovat dva postupy vyhodnocení správné pozice sw. nástroje (viz. tabulka 24).

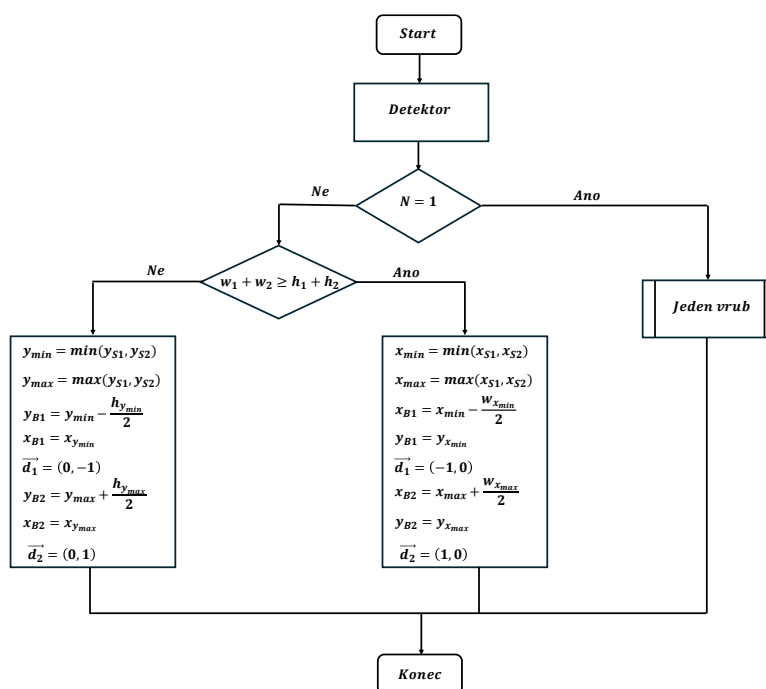
V případě, že je na výstupu z detektoru pouze jedna instance této třídy, pozice podstatného bodu je podobně jako u předchozích tříd vyhodnocena na základě vztahu šířky a výšky hraničního obdélníku (na obrázku 53 vlevo znázorněn žlutě), ale navíc je zohledněna poloha samotného obdélníku vůči rozměrům snímku. Například v případě, že je šířka hraničního obdélníku větší než výška a současně je x -ová složka středového bodu obdélníku menší než šířka původního snímku, lze předpokládat, že je vrub horizontálně zarovnaný a vyskytuje se v levé části snímku a sw. nástroj by měl být orientovaný od lokalizovaného bodu u kořene vrubu směrem doprava (předpokládaný směr šíření trhliny). Informaci o směru umístění sw. nástroje nese vektor označený \vec{d} . Popsaný příklad je znázorněn obrázku 53 vlevo, kde je patrné, že šířka w je větší než výška h a zároveň střed S , respektive jeho



Obrázek 53: Podstatné body z hlediska umístění sw. nástroje u třídy vrubovaných těles

x -ová složka je menší než $\frac{W}{2}$ (naznačeno zeleně) a vektor $\vec{d} = (1,0)$ udává příslušný směr od bodu B doprava. Tento způsob vyhodnocení orientace vrubu ve snímku s sebou nese určitá omezení. Prvním omezením je předpoklad, že vrub je ve snímku dostatečně viditelný na to, aby bylo možné ze vztahu rozměrů hraničního obdélníku určit jeho horizontální, případně vertikální zarovnání. Druhým omezením je předpoklad, že se vrub vyskytuje svojí délkou pouze v jedné polovině snímku, jinak by byla pozice i orientace sw. nástroje vyhodnocena zrcadlově. Z uvedených omezení je možné usuzovat, že způsob nalezení pozice vrubu není efektivní, nicméně je nutné zmínit, že se silně opírá o charakter měření lomových vlastností, respektive o typické a vhodné zarovnání vzorku v poli kamery snímající záznam, kde je podstatný kořen vrubu a oblast potenciálního šíření trhliny, jenž je směrem do tělesa. Vývojový diagram uvedeného procesu vyhodnocení je znázorněn na obrázku 52 vpravo. První podmínka ve vývojovém diagramu reprezentuje rozdíl ve zpracování výstupu z detektoru pro tuto třídu při jedné predikci nebo při dvou. Jedna predikce odpovídá popsanému procesu v tomto odstavci a tedy snímku, na kterém se vyskytuje pouze jeden kořen vrubu.

Ve vývojovém diagramu na obrázku 52 vpravo se nesplnění první podmínky zpracování odkazuje na proces zpracování snímku s MT tělesem. Tento proces je doplněn na obrázku 54 v příslušném vývojovém diagramu. Zarovnání vzorku, respektive jeho vrubů, je odvozeno ze



Obrázek 54: Vyhodnocení pozice sw. nástroje u MT tělesa

vztahu součtu šířek a součtu výšek hraničních obdélníků. Z tohoto vztahu lze přímo odvodit i pozici podstatných bodů a směry od kořene vrubu do tělesa, což je patrné z obrázku 53 vpravo. Ve vývojovém diagramu jsou u některých veličin v dolním indexu uvedena minima, případně maxima souřadnic, což naznačuje, že se jedná o hodnotu korespondující s příslušným hraničním obdélníkem, ke kterému patří souřadnice určená pomocí minima nebo maxima.

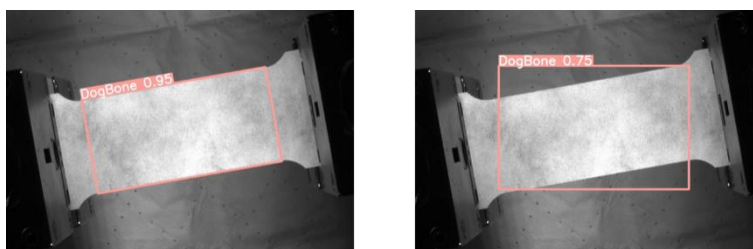
6.2 Korekce nezarovnaných vzorků

Určitým omezením uvedeného principu vyhodnocení pozice sw. nástroje je skutečnost, že z predikovaného hraničního obdélníku nelze odvodit případné natočení vzorku v poli kamery (obrázek 55). Korekci tohoto natočení lze provést zásahem do umístění sw. nástroje odvozeného dle popsaného postupu v předchozím textu. Zásahy uživatele jsou obecně nezbytné vždy, nicméně pro jejich minimalizaci byly navrženy dvě možnosti na základě kterých by bylo možné do jisté míry popsané omezení eliminovat. První možností je použít orientované hraniční obdélníky a druhou určení významných směrů v obraze.

Orientované hraniční obdélníky

Celý doposud uvedený proces spadal do tzv. horizontální detekce, tedy detekce, kdy je na výstupu vždy horizontálně orientovaný hraniční obdélník. Tento způsob detekce je obecně nejpoužívanější a existuje pro něj nejvíce modelů, nicméně nese s sebou určité omezení, jenž bylo naznačeno v kontextu řešené úlohy v úvodu této podkapitoly. Jiným způsobem detekce je detekce natočených objektů, kdy jsou na výstupu z detektoru tzv. orientované hraniční obdélníky (OBB z „*Oriented bounding box*“ někdy též „*Rotated bounding box*“), které jsou popsány středem, výškou, šířkou a na rozdíl od horizontálních hraničních obdélníků (BB) i úhlem natočení. Detekce natočených objektů se často používá při zpracování leteckých snímků [57]. Rozdíl mezi jednotlivými detekcemi je uveden na obrázku 55.

Společnost Ultralytics, jenž implementovala algoritmus YOLOv8 použitý v této práci, poskytuje i několik modelů umožňujících detekci pomocí natočených obdélníků, nicméně v rámci této práce nebyly implementovány. Pro charakter snímků při testování materiálů s použitím DIC, o němž bylo mnohokrát pojednáno, byla horizontální detekce dostatečně přesná a natočené vzorky, pro které by byl OBB vhodnější, lze uživatelem zkorigovat při doladění pozice příslušného sw. nástroje. Nicméně je to model, jehož použití by mohlo vytvořit invarianci detektoru vůči jakkoli natočeným vzorkům a jehož implementace by byla z hlediska postupu velice podobná jako popsaná implementace modelu (detektoru). Navíc v rámci této práce vytvořený dataset (kapitola 4) byl anotován ve formátu polygonů, jež lze převést na orientované hraniční obdélníky. Zmíněná fakta by mohla otevřít cestu k navázání na tuto práci s použitím modelu umožňující detekci natočených objektů pomocí OBB.



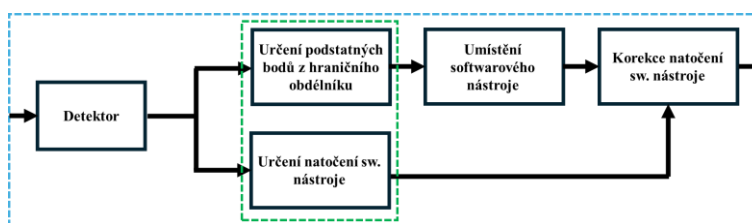
Obrázek 55: OBB vlevo a BB vpravo

Určení významných směrů v obraze

Druhou možností korekce nezarovnaných (natočených) vzorků je systém navržený pro zpracování výstupu z detektoru, jenž určuje úhel tohoto natočení, který spolu s podstatnými body odvozenými z hraničních obdélníků umožňuje přesnější umístění příslušného sw. nástroje. Tento systém by tedy mohl eliminovat nepřesné umístění sw. nástroje např. vlivem chyby uživatele videoextenzometru při zarovnávaní vzorku v jeho úhlu záběru.

Před popisem principu navrženého systému je nutné zmínit, že se jedná o návrh řešení korekce nezarovnaných (natočených) vzorků a jeho popis by měl sloužit jako případná inspirace pro rozšíření funkčnosti detektoru v konkrétní aplikaci, pokud by to bylo nutné.

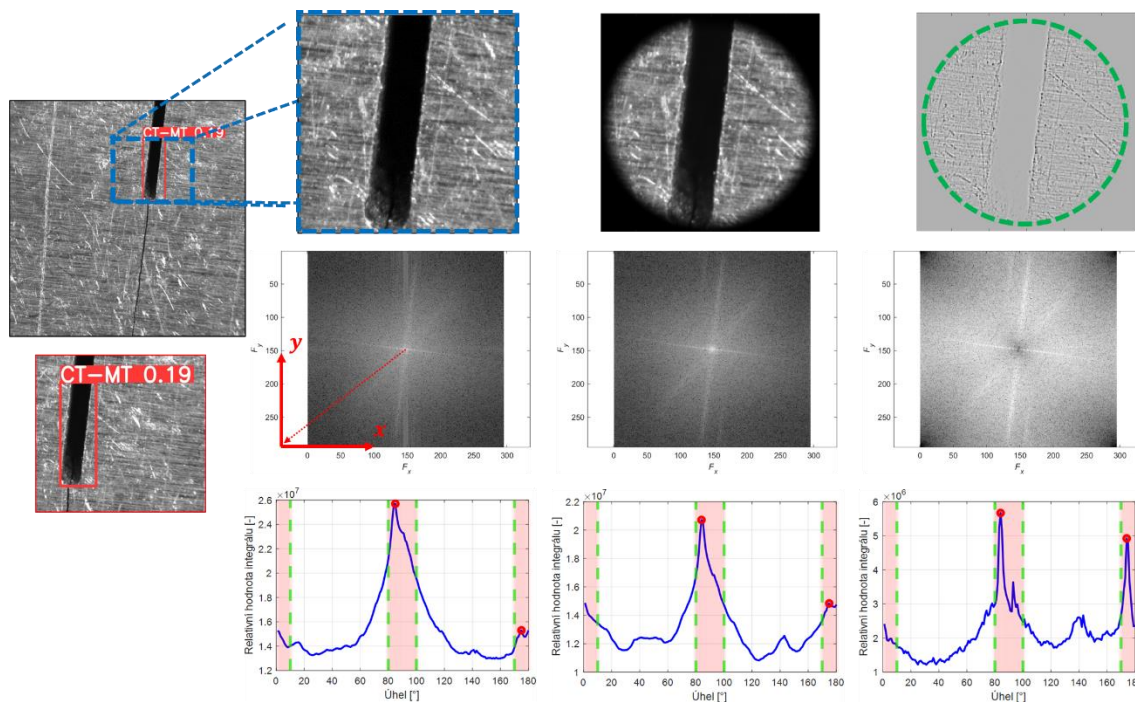
Princip navrženého systému je schematicky znázorněn na obrázku 56 a jedná se o rozšíření modře zvýrazněné části schéma na obrázku 49. Zeleně ohraničené bloky reprezentují rozšíření bloku zpracování výstupu z detektoru z obrázku 49, kde oproti původnímu významu tohoto bloku jako určení podstatných bodů z hraničního obdélníku přibylo ještě určení natočení sw. nástroje.



Obrázek 56: Schéma navrženého systému korekce natočení vzorku

Princip určení tohoto natočení spočívá v oříznutí oblasti ohraničené hraničním obdélníkem, určení Fourierova amplitudového spektra takto vzniklého obrazu (oříznuté oblasti) dvourozměrnou diskretní Fourierovou transformací a pomocí tohoto amplitudového spektra odvození úhlu natočení vzorku. Jednotlivé kroky naznačeného principu jsou znázorněny na obrázku 57.

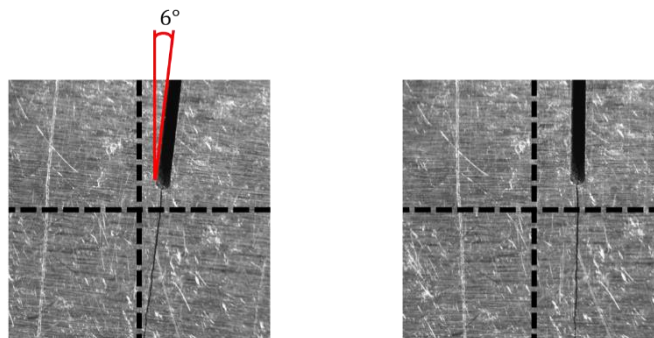
Na obrázku 57 vlevo nahoře (ohraničeno modře) je patrné, že detektor detekoval konec vrubu a přiřadil k němu příslušný hraniční obdélník s nižší konfidencí vlivem zřetelného natočení. Oříznutá oblast je větší než oblast vymezená hraničním obdélníkem, což bylo provedeno z důvodu, jenž bude vysvětlen později. Na obrázku 57 v prostřední řadě vlevo je uvedeno Fourierovo amplitudové spektrum oříznuté oblasti. Ve Fourierově amplitudovém spektru (dále jen spektru) se s největší intenzitou projevují významné směry v obraze, což mohou být například hrany, rovné struktury nebo v kontextu práce vruby nebo pruty (z geometrického hlediska) [58]. Tyto směry se ve spektru projeví jako výrazné pruhy (paprsky) a lze je určit pomocí numerické integrace podél polopřímek vycházející z jeho středu (je vhodné podotknout, že se jedná o logaritmované amplitudové spektrum po prohození jeho kvadrantů viz. [58]). Zmíněný algoritmus numerické integrace je uveden v [58]. Tuto integraci je nutno provést pouze v intervalu $\langle 0^\circ, 180^\circ \rangle$, jelikož se jedná o středově symetrické spektrum a jeho další vlastnost spočívá v tom, že je pootočené o 90° oproti původnímu obrazu [58]. Jinými slovy, horizontální struktura se ve spektru projeví jako vertikální a naopak. Výsledkem numerické integrace podél polopřímek je závislost jednotlivých hodnot integrálů na úhlu ve spektru od vyznačené osy x (v obrázku je osa x v posunutém souřadném systému ze středu do rohu, aby nezakrýval podstatné informace). V této závislosti uvedené na obrázku vlevo dole je zřetelné maximum na úhlu 84° , což odpovídá směru natočení vrubu (v této a i v dalších



Obrázek 57: Dílčí kroky určení natočení vzorku

závislostech je zohledněno zmíněné pootočení oproti skutečnému obrazu). Pomocí tohoto úhlu lze pak odvodit, natočení příslušného sw. nástroje jehož umístění by bylo provedeno na základě bodu na konci vrubu dle postupu na obrázku 52 vpravo. Konkrétněji v popisovaném případě by se jednalo o natočení 6° se středem v bodu na konci vrubu a ve směru hodinových ručiček. Natočení oříznuté oblasti dle odvozeného úhlu je znázorněno na obrázku 58.

K popsanému systému v předchozím odstavci je vhodné doplnit několik doporučení k jeho použití a implementaci. První spočívá v naznačené skutečnosti o oříznutí větší oblasti snímku než pouze oblasti hraničního obdélníku. To souvisí s tím, že Fourierovo spektrum musí být vytvořeno ze čtvercového obrazu (čtvercové oblasti), aby nedošlo ke zkreslení úhlů směřů v obraze a zároveň s tím, že ve Fourierově spektru se projevují hrany obrazu jako významné směry, což lze pozorovat na obrázku 57 v prostřední řadě vlevo, kdy je dobře viditelný vliv horizontální hrany (ve spektru jí odpovídá vertikální pruh) [58]. Horizontální pruh reprezentující vertikální hranu není tak dobře patrný, nicméně ve spektru se vyskytuje. Tento vliv lze eliminovat aplikováním kruhové masky (na obrázku v horní řadě uprostřed), což se ve spektru projeví i eliminováním pruhů od hran (na obrázku v prostřední řadě uprostřed) [58]. U některých snímků může být vytvoření čtvercové oblasti s ohledem na původní hraniční obdélník obtížné kvůli tomu, že vzorek může zabírat velkou část snímku a zároveň detekovaná oblast je obdélník. Tím pádem nemusí vzniknout prostor pro rozšíření hraničního obdélníku, což lze řešit například vyhodnocením těchto skutečností na začátku popisovaného procesu a vybráním vhodné oblasti uvnitř hraničního obdélníku vstupující do dalšího zpracování jako například čtverec o stejné délce strany jako kratší ze dvou stran hraničního obdélníku se středem totožným jako u hraničního obdélníku. Dalším možným způsobem je například umístění středu čtvercové oblasti do středu hraničního obdélníku podobně jako u předchozího způsobu, následně určení vzdálenosti tohoto středu ve směru kolmém k delší straně hraničního obdélníku k bližší straně snímku a z této vzdálenosti následně odvození maximální možné délky strany čtvercové oblasti.



Obrázek 58: Natočení oříznuté oblasti dle určeného úhlu

Druhé doporučení souvisí s se zvýrazněnými oblastmi v závislostech uvedených na obrázku 57 dole. V těchto závislostech jsou zvýrazněny intervaly $(0,10^\circ)$, $(80^\circ,100^\circ)$ a $(170^\circ,180^\circ)$, jenž reprezentují naznačené rozsahy, ve kterých je možné hledat natočení příslušného sw. nástroje, jelikož chování detektoru vůči více natočeným vzorkům bylo z popsaných důvodů nepředvídatelné. Ve kterém z uvedených intervalů se hledaný úhel natočení vyskytuje, lze odvodit z porovnání výšky a šířky hraničního obdélníku. V případě, že je výška větší než šířka (obrázek 57) lze odvodit, že se jedná o neúplně vertikálně zarovnaný vzorek. V tomto případě hledaný úhel náleží do intervalu $(80^\circ; 100^\circ)$ a v opačném případě při neúplně horizontálním zarovnání by hledaný úhel náležel do zbylých dvou intervalů.

Třetí doporučení souvisí s pravým sloupcem na obrázku 57. U této čtvercové oblasti byla kromě kruhové masky (naznačeno zeleně) navíc provedena konvoluce snímku s následujícím konvolučním jádrem:

$$k = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Výsledkem této konvoluce je obraz, ve kterém jsou zvýrazněny hrany objektů (obrázek 57 nahoře vlevo). Tato technika zvýraznění hran a konvoluční jádro (též označováno jako „Laplacian filter“) jsou převzaty z [62]. Z Fourierova spektra a závislosti intenzity směrů na úhlech je patrné, že zvýraznění hran významně zvětšilo lokální maxima úhlů významných směrů. Úhel v závislosti vpravo dole v prostředním intervalu odpovídá již určenému natočení vrubu, avšak je patrné, že došlo ke zvýraznění dalšího úhlu v intervalu vpravo. Jeho hodnota je 173° a určuje dominantní orientaci struktury na povrchu vzorku, jenž vznikla technologickou operací při jeho přípravě. Lze konstatovat, že použití uvedené konvoluce pro zvýraznění hran může značně zvýraznit lokální maxima v závislosti významných směrů na úhlech v obraze, což by mohlo být výhodné pro snímky se složitějším pozadím (např. obrázek 47 uprostřed).

Navržený systém může do jisté míry eliminovat nutnost zásahu uživatele při umístění sw. nástroje odvozeného ze samotného výstupu z detektoru, nicméně určitá míra uživatelských zásahů je vždy nutná, a proto je ve znázorněném schématu na obrázku 56 rozšíření zpracování výstupu z detektoru o určení natočení sw. nástroje naznačeno paralelně k samotnému umístění sw. nástroje dle hraničního obdélníku. Paralelní umístění bylo zamýšleno tak, že korekce natočení sw. nástroje se ve výchozím nastavení neprovede (přednastavena na 0°) a paralelní větev s určením natočení pro případnou korekci se zaktivuje, pouze pokud to uživatel umožní.

7 Diskuse

I přes dobré výsledky vytvořeného detektoru RS44 na testovací množině v2 a ucházející výsledky na testovací množině v3 (s ohledem na podstatu jejího vytvoření viz. podkapitola 4.8) lze zhodnotit jeho funkčnost a užitečnost až při použití v praktické aplikaci. Tedy při jeho integrování do softwaru pro práci s videoextenzometrií.

U vytvořeného detektoru se vyskytují dva jevy, na které je při jeho použití vhodné upozornit. Prvním jevem souvisí s poměrně častým výskytem falešných pozitivů tahových vzorků, jenž se mohou vyskytovat na strukturách v pozadí snímků. O tomto jevu bylo pojednáno v podkapitole o vyhodnocení detektoru (podkapitola 5.6). Jeho vliv je do značné míry eliminován skutečností, že pro práci s detektorem je v kontextu testování materiálů s použitím DIC podstatná pouze detekce s největší konfidencí, případně první dvě detekce s největšími konfidencemi (MT vzorek), které detektor v drtivé většině vytvářel správně.

Druhý zmíněný jev se může u detektoru vyskytnout ve chvíli, kdy je vzorek vůči videoextenzometru natočený (nezarovnaný). V této situaci lze u detektoru očekávat pokles citlivosti. Detektor je vůči tomuto jevu do jisté míry invariantní, jelikož byl vytvořen s pomocí databáze výrobce videoextenzometrů, ve které se tento jev vyskytoval a v podkapitole 6.2 byl navržen systém pro korekci umístění sw. nástroje v intervalech potenciálního natočení vzorku (např. vlivem nepřesnosti v zarovnání uživatelem videoextenzometru) v oblasti předpokládané citlivosti detektoru. Nicméně v rámci augmentace datasetu nebylo další natočení snímků z uvedených důvodů zaneseno (kromě otočení o 90°), a proto je zmíněná invariance pouze v souladu s průměrnými odchylkami od standardních záznamů v oblasti testování materiálů s použitím DIC. Při větších odchylkách (v extrémním případě vzorek v poli kamery diagonálně) může dojít k vyhodnocení vzorku jako falešného negativu a tím pádem nedojde k umístění příslušného sw. nástroje. Pro vytvoření invariantního detektoru vůči jakkoli natočeným vzorkům byl v podkapitole 6.2 navržen postup pomocí detekce natočených hraničních obdélníků.

U obou uvedených jevů lze předpokládat, že nebudou z hlediska funkčnosti detektoru zásadně omezující, nicméně je nutné je při práci s detektorem brát v potaz.

V rámci následující části diskuse jsou naznačeny oblasti práce, které by mohly být rozšířeny nebo by na ně mohlo být navázáno.

První oblastí možného rozšíření je skutečnost, že základní model (označený jako SGD dle jeho opt. algoritmu) dosahoval na testovacích množinách v2 a v3 srovnatelných výsledků jako ostatní modely. Při procesu optimalizace hyperparametrů byly jako opt. algoritmy pro optimalizované modely zvoleny algoritmy Adam a AdamW z popsáných důvodů, nicméně lze odhadovat, že při optimalizaci modelů s algoritmem SGD by bylo možné vytvořit takový model, jenž by mohl ve výsledku konkurovat nejlepšímu modelu algoritmu označeném RS44. Nicméně pro prověření tohoto závěru by bylo nutné příslušnou optimalizaci provést.

Druhou oblastí práce, na kterou nebylo v jejím vypracování navázáno je analýza vlivu jednotlivých hyperparametrů na chování a výkonnostní metriky modelů. Počet optimalizovaných hyperparametrů byl poměrně velký (obsahoval 9 hyperparametrů) a v rámci náhodného prohledávání prostoru hyperparametrů byl proveden trénink celkem 60 modelů, tedy 60 náhodně vytvořených kombinací hyperparametrů, nicméně kromě určení nejlepších třech modelů nebylo na výsledky nijak navázáno např. pomocí citlivostní analýzy. Tato analýza

by mohla vytvořit větší náhled na vliv jednotlivých hyperparametrů a při pokročilejší optimalizaci zredukovat jejich počet na základě jejich významnosti.

8 Závěr

Tato diplomová práce se zabývala automatizací použití videoextenzometrů s využitím metod umělé inteligence, a to konkrétně implementací vhodného algoritmu pro rozpoznání typu měřeného vzorku a určení jeho polohy.

První kapitola pojednává o současném stavu poznání v oborech, do kterých práce zasahuje. Celá tato kapitola by se dala rozdělit do tří celků. První pojednává o metodě digitální korelace obrazu, která úzce souvisí s použitím videoextenzometrů v praxi. Druhý se zabývá oborem počítačového vidění se zaměřením na rozpoznávání objektů v obraze a jsou v něm uvedeny jeho základní přístupy. Třetí a nejrozsáhlejší celek pojednává o strojovém učení. Jsou zde podrobně popsány neuronové sítě a jejich modifikace, učení neuronových sítí a dále jsou uvedeny architektury hlubokých konvolučních neuronových sítí pro detekci objektů v obraze, které jsou závěrem teoretické části práce porovnány na veřejných testovacích datasetech.

Na pomezí mezi teoretickou a praktickou částí je uvedena analýza problému a jsou určeny metody k jeho řešení na základě poznatků z teoretické části. Jako nejvhodnější algoritmus byl zvolen algoritmus YOLOv8, který umožňuje detekci (tedy klasifikaci a lokalizaci pomocí hraničního obdélníku) vzorků v reálném čase. V době vypracování této práce byl označován jako SOTA („state-of-the-art“) mezi detekčními algoritmy a je implementován v programovacím jazyce Python, který byl také pro práci s tímto algoritmem použit.

První kapitola praktické části práce pojednává o tvorbě datasetu pro implementaci zvoleného detekčního algoritmu. Je v ní popsána databáze výrobce videoextenzometrů X-Sight, která byla zpřístupněna za účelem vypracování této práce, a na základě které byly vybrány tři typy vzorků pro detekci. Jednalo se o tahové vzorky, vzorky prutů betonářské výztuže a vzorky vrubovaných těles. Na tyto vzorky, respektive na snímky tyto vzorky obsahující, byly stanoveny předpoklady s ohledem na použití v kontextu testování materiálů pomocí DIC a byl vytvořen dataset. Tento dataset byl rozšířen o vlastní snímky pořízené v laboratoři Ústavu mechaniky těles FSI VUT, dále upraven pomocí několika augmentačních technik a rozšířen o snímky bez objektů pro zvětšení jeho robustnosti. Na konci kapitoly bylo pojednáno o rozdělení datasetu do trénovacích, dvou validačních a tří testovacích množin a o anotaci snímků. Dataset byl anotován pomocí polygonů, které je možné převést do různých anotačních formátů pro různé detekční algoritmy a dataset tak není vázán pouze k anotačnímu formátu využívaném algoritmem YOLOv8.

Druhá kapitola praktické části práce se zabývá implementací algoritmu YOLOv8. Na jejím začátku je pojednáno o nastavení tréninku algoritmu z pohledu hyperparametrů a optimalizačního algoritmu. Pro porovnání optimalizačních algoritmů byly trénovány tři modely. První s opt. algoritmem SGD a další dva s opt. algoritmy Adam a AdamW. Z porovnání těchto modelů bylo patrné, poslední dva zmíněné modely dosahovaly větší citlivosti a metrik na bázi mAP, a proto byly použity při procesu optimalizace hyperparametrů. Cílem optimalizace hyperparametrů bylo najít takové hyperparametry, které vytvoří efektivní podmínky pro trénink a vylepší celkovou výkonost optimalizovaných modelů. Pro tento proces byly použity dvě metody. První metoda byla optimalizace pomocí genetického algoritmu a druhá optimalizace pomocí náhodného prohledávání prostoru hyperparametrů. Nejlepší optimalizované modely byly na závěru této kapitoly porovnány spolu s neoptimalizovanými modely na testovacích množinách. Na základě porovnání byl zvolen nejlepší model s označením RS44 získaný z náhodného prohledávání prostoru hyperparametrů. Tento model

dosahoval na testovací množině splňující požadavky použití DIC při testování materiálů přesnost 92.9 %, citlivost 88.9 %, mAP50 96.8 % a mAP50-95 79.1 %. Na testovací množině, která byla vytvořena za hranice použitelnosti DIC při testování materiálů, dosahoval přesnost 79.1 %, citlivost 64.4 %, mAP50 75.5 % a mAP50-95 53.9 %.

Dále bylo podrobně pojednáno o chování nejlepšího modelu RS44 na konkrétních příkladech snímků z testovacích množin a o závislostech přesnosti, citlivosti a parametru F1 na prahu konfidence modelu. Jako určitý nežádoucí jev se ukázalo, že model často vytvářel falešné detekce třídy tahových vzorků na strukturách v pozadí snímků. Nicméně tyto detekce v drtivé většině případů nedosahovaly větší konfidence než detekce skutečné a při praktické aplikaci modelu v kontextu testování materiálů s využitím DIC (videoextenzometrů) je typicky významná pouze detekce s největší konfidencí.

V poslední kapitole práce je pojednáno o modelu RS44 jako detektoru integrovaném do softwaru pro práci s videoextenzometry. Na základě jeho chování byl navržen postup pro vyhodnocení jeho výstupů, jenž jsou hraniční obdélníky reprezentující oblast výskytu detekovaného vzorku v obraze, a princip umístění příslušných softwarových nástrojů. Závěrem této kapitoly byly navrženy dva postupy pro korekci natočení vzorků při umístění softwarového nástroje, které nelze přímo vyhodnotit z výstupů z detektoru. Jeden z těchto postupů byl podrobněji rozpracován formou doporučení při jeho integraci spolu s detektorem.

Přínosem této práce je model RS44, jenž lze s ohledem na stanovené předpoklady použití integrovat jako detektor pro určení typu vzorku a jeho polohy do softwaru pro práci s videoextenzometry.

BIBLIOGRAFIE

- [1] ZHAO, Jianlong; SANG, Yong a DUAN, Fuhai. *The state of the art of two-dimensional digital image correlation computational method*. Online. *Engineering reports (Hoboken, N.J.)*. 2019, roč. 1, č. 2. ISSN 2577-8196. Dostupné z: <https://doi.org/10.1002/eng2.12038>. [cit. 2024-02-17].
- [2] ORTEU, Jean-josé. *3-D computer vision in experimental mechanics*. Online. *Optics and lasers in engineering*. 2009, roč. 47, č. 3, s. 282-291. ISSN 0143-8166. Dostupné z: <https://doi.org/10.1016/j.optlaseng.2007.11.009>. [cit. 2024-05-18].
- [3] CNN Series Part 1: *How do computers see images?* Online. Medium. 2020. Dostupné z: <https://medium.com/analytics-vidhya/cnn-series-part-1-how-do-computers-see-images-32462a0b33ca>. [cit. 2024-05-18].
- [4] LAVA, Pascal; COOREMAN, Steven; COPPIETERS, Sam; DE STRYCKER, Maarten a DEBRUYNE, Dimitri. *Assessment of measuring errors in DIC using deformation fields generated by plastic FEA*. Online. 2009. ISSN 1873-0302. Dostupné z: <https://doi.org/10.1016/j.optlaseng.2009.03.007>. [cit. 2024-02-23].
- [5] KLETTE, Reinhard. *Concise Computer Vision: An Introduction into Theory and Algorithms*. 2014. London: Springer Nature, 2014. ISBN 1447163206. ISSN 1863-7310. Dostupné z: <https://doi.org/10.1007/978-1-4471-6320-6>.
- [6] HUANG, T. *Computer Vision: Evolution And Promise*. Online. 1996. Dostupné z: <https://doi.org/10.5170/CERN-1996-008.21>. [cit. 2024-05-18].
- [7] FOFI, David; PRICE, Jeffery R.; MERIAUDEAU, Fabrice; SLIWA, Tadeusz a VOISIN, Yvon. *A comparative survey on invisible structured light*. Online. In: . 2004, 90-. Dostupné z: <https://doi.org/10.1117/12.525369>. [cit. 2024-05-18].
- [8] WU, Xiongwei; SAHOO, Doyen a HOI, Steven C.H. *Recent advances in deep learning for object detection*. Online. *Neurocomputing (Amsterdam)*. 2020, roč. 396, s. 39-64. ISSN 0925-2312. Dostupné z: <https://doi.org/10.1016/j.neucom.2020.01.085>. [cit. 2024-03-03].
- [9] KOZA, John R.; BENNETT, Forrest H.; ANDRE, David a KEANE, Martin A. *Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming*. Online. In: GERO, John S. a SUDWEEKS, Fay (ed.). *Artificial Intelligence in Design '96*. Dordrecht: Springer Netherlands, 1996, s. 151-170. ISBN 978-94-010-6610-5. Dostupné z: https://doi.org/10.1007/978-94-009-0279-4_9. [cit. 2024-05-18].
- [10] TRISAL, Adya a MANDLOI, Dheeraj. *MACHINE LEARNING: AN OVERVIEW*. Online. *International journal of research - granthaalayah*. 2021, roč. 9, č. 7, s. 343-348. ISSN 2394-3629. Dostupné z: <https://doi.org/10.29121/granthaalayah.v9.i7.2021.4120>. [cit. 2024-03-10].

- [11] HU, Junyan; NIU, Hanlin; CARRASCO, Joaquin; LENNOX, Barry a ARVIN, Farshad. *Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning*. Online. IEEE transactions on vehicular technology. 2020, roč. 69, č. 12, s. 14413-14423. ISSN 0018-9545. Dostupné z: <https://doi.org/10.1109/TVT.2020.3034800>. [cit. 2024-03-10].
- [12] ALPAYDIN, Ethem. *Introduction to machine learning*. 2nd ed. Massachusetts: MIT Press, 2010. ISBN 978-0-262-01243-0.
- [13] NASTESKI, Vladimír. *An overview of the supervised machine learning methods*. Online. HORIZONS.B. 2017, roč. 4, s. 51-62. ISSN 18578578. Dostupné z: <https://doi.org/10.20544/HORIZONS.B.04.1.17.P05>. [cit. 2024-05-18].
- [14] RUSSELL, Stuart J. (Stuart Jonathan) a NORVIG, Peter. *Artificial intelligence: a modern approach*. Third edition; Authorized adaptation from the United States edition 2010. Boston ; London: Pearson, 2016. ISBN 978-1-292-15396-4.
- [15] V, Sindhu. AN EMPIRICAL SCIENCE RESEARCH ON BIOINFORMATICS IN MACHINE LEARNING. Online. JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES. 2020, roč. spl7, č. 1. ISSN 09738975. Dostupné z: <https://doi.org/10.26782/jmcms.spl.7/2020.02.00006>. [cit. 2024-05-18]. Non-linear distance based large scale data classifications
- [16] KECCMAN, V. *Support Vector Machines – An Introduction*. Online. In: WANG, Lipo (ed.). *Support Vector Machines: Theory and Applications*. Studies in Fuzziness and Soft Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, s. 1-47. ISBN 978-3-540-24388-5. Dostupné z: https://doi.org/10.1007/10984697_1. [cit. 2024-05-18].
- [17] TAMMINA, Srikanth. *Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images*. Online. International Journal of Scientific and Research Publications (IJSRP). 2019, roč. 9, č. 10. ISSN 2250-3153. Dostupné z: <https://doi.org/10.29322/IJSRP.9.10.2019.p9420>. [cit. 2024-05-18].
- [18] CHIH-WEI HSU a CHIH-JEN LIN. *A comparison of methods for multiclass support vector machines*. Online. IEEE Transactions on Neural Networks. Roč. 13, č. 2, s. 415-425. ISSN 10459227. Dostupné z: <https://doi.org/10.1109/72.991427>. [cit. 2024-05-18].
- [19] BHARTENDOO VIMAL. *Application of Logistic Regression in Natural Language Processing*. Online. International Journal of Engineering Research and. 2020, roč. V9, č. 06. ISSN 2278-0181. Dostupné z: <https://doi.org/10.17577/IJERTV9IS060095>. [cit. 2024-05-18].
- [20] CS231n: *Convolutional Neural Networks for Visual Recognition*. Online. Dostupné z: <https://cs231n.github.io/>. [cit. 2024-05-18].
- [21] ALZUBAIDI, Laith; ZHANG, Jinglan; HUMAIDI, Amjad J.; AL-DUJAILI, Ayad; DUAN, Ye et al. *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*. Online. Journal of Big Data. 2021, roč. 8, č. 1. ISSN 2196-1115. Dostupné z: <https://doi.org/10.1186/s40537-021-00444-8>. [cit. 2024-05-18].

- [22] RUDER, Sebastian. *An overview of gradient descent optimization algorithms*. Online. ArXiv.org. 2017. Dostupné z: <https://doi.org/10.48550/arxiv.1609.04747>. [cit. 2024-05-18].
- [23] KINGMA, Diederik P a BA, Jimmy. *Adam: A Method for Stochastic Optimization*. Online. ArXiv.org. 2017. Dostupné z: <https://doi.org/10.48550/arxiv.1412.6980>. [cit. 2024-05-18].
- [24] *What Is Overfitting?* Online. MathWorks. Dostupné z: <https://www.mathworks.com/discovery/overfitting.html>. [cit. 2024-05-21].
- [25] *Fine-Tuning Stable Diffusion With Validation*. Online. 2023. Dostupné z: <https://medium.com/@damian0815/fine-tuning-stable-diffusion-with-validation-3fe1395ab8c3>. [cit. 2024-05-18].
- [26] GIRSHICK, Ross; DONAHUE, Jeff; DARRELL, Trevor a MALIK, Jitendra. *Rich feature hierarchies for accurate object detection and semantic segmentation*. Online. ArXiv.org. 2014. Dostupné z: <https://doi.org/10.48550/arxiv.1311.2524>. [cit. 2024-05-18].
- [27] GIRSHICK, Ross. *Fast R-CNN*. Online. In: 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, 2015, s. 1440-1448. Dostupné z: <https://doi.org/10.1109/ICCV.2015.169>. [cit. 2024-05-18].
- [28] REN, Shaoqing; HE, Kaiming; GIRSHICK, Ross a SUN, Jian. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. Online. ArXiv.org. 2016. Dostupné z: <https://doi.org/10.48550/arxiv.1506.01497>. [cit. 2024-05-18].
- [29] REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross a FARHADI, Ali. *You Only Look Once: Unified, Real-Time Object Detection*. Online. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016, s. 779-788. Dostupné z: <https://doi.org/10.1109/CVPR.2016.91>. [cit. 2024-05-18].
- [30] *Ultralytics YOLOv5 Architecture*. Online. Ultralytics. 2023. Dostupné z: https://docs.ultralytics.com/yolov5/tutorials/architecture_description/. [cit. 2024-05-21].
- [31] TERVEN, Juan a CORDOVA-ESPARZA, Diana. *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*. Online. 2023. Dostupné z: <https://doi.org/10.48550/arxiv.2304.00501>. [cit. 2024-05-18].
- [32] LI, Yanghao; CHEN, Yuntao; WANG, Naiyan a ZHANG, Zhao-xiang. *Scale-Aware Trident Networks for Object Detection*. Online. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019, s. 6053-6062. Dostupné z: <https://doi.org/10.1109/ICCV.2019.00615>. [cit. 2024-05-18].
- [33] LIU, Haiying; SUN, Fengqian; GU, Jason a DENG, Lixia. *SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode*. Online. Sensors (Basel, Switzerland). 2022, roč. 22, č. 15, s. 5817. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s22155817>. [cit. 2024-05-18].
- [34] *Ultralytics YOLOv8 Docs*. Online. Ultralytics. 2023. Dostupné z: <https://docs.ultralytics.com/>. [cit. 2024-05-19].

- [35] *Brief summary of YOLOv8 model structure*. Online. Github. 2023. Dostupné z: <https://github.com/ultralytics/ultralytics/issues/189>. [cit. 2024-05-21].
- [36] *Why YOLOv5*. Online. Dostupné z: <https://github.com/ultralytics/yolov5>. [cit. 2024-05-18].
- [37] *Object Detection*. Online. Github. 2023. Dostupné z: <https://docs.ultralytics.com/tasks/detect/>. [cit. 2024-05-18].
- [38] *Tips for Best Training Results*. Online. Ultralytics YOLOv8 Docs. 2023. Dostupné z: https://docs.ultralytics.com/yolov5/tutorials/tips_for_best_training_results/. [cit. 2024-05-19].
- [39] *Fracture mechanics (LEFM, YFM)*. Online. ZwickRoell. Dostupné z: <https://www.zwickroell.com/industries/materials-testing/fracture-mechanics/>. [cit. 2024-05-19].
- [40] SHORTEN, Connor a KHOSHGOFTAAR, Taghi M. *A survey on Image Data Augmentation for Deep Learning*. Online. Journal of big data. 2019, roč. 6, č. 1, s. 1-48. ISSN 2196-1115. Dostupné z: <https://doi.org/10.1186/s40537-019-0197-0>. [cit. 2024-05-19].
- [41] *Getting Started with Roboflow*. Online. Roboflow. 2023. Dostupné z: <https://blog.roboflow.com/getting-started-with-roboflow/>. [cit. 2024-05-19].
- [42] *What Is a Pretrained AI Model?* Online. Nvidia. 2022. Dostupné z: <https://blogs.nvidia.com/blog/what-is-a-pretrained-ai-model/>. [cit. 2024-05-19].
- [43] *README*. Online. Github. Dostupné z: <https://github.com/ultralytics/ultralytics/blob/main/README.md>. [cit. 2024-05-21].
- [44] WANG, Xufei a SONG, Jeongyoung. *CIoU: Improved Loss Based on Complete Intersection Over Union for Bounding Box Regression*. Online. IEEE access. 2021, roč. 9, s. 105686-105695. ISSN 2169-3536. Dostupné z: <https://doi.org/10.1109/ACCESS.2021.3100414>. [cit. 2024-05-19].
- [45] LI, Xiang; WANG, Wenhai; WU, Lijun; CHEN, Shuo; HU, Xiaolin et al. *Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection*. Online. ArXiv.org. 2020. Dostupné z: <https://doi.org/10.48550/arxiv.2006.04388>. [cit. 2024-05-19].
- [46] *Ultralytics YOLOv8 Docs*. Online. Ultralytics. Dostupné z: <https://docs.ultralytics.com/usage/cfg/>. [cit. 2024-05-19].
- [47] LOSHCHILOV, Ilya a HUTTER, Frank. *Decoupled Weight Decay Regularization*. Online. ArXiv.org. 2019. Dostupné z: <https://doi.org/10.48550/arxiv.1711.05101>. [cit. 2024-05-19].
- [48] FEURER, Matthias a HUTTER, Frank. *Hyperparameter Optimization*. Online. In: HUTTER, Frank; KOTTHOFF, Lars a VANSCHOREN, Joaquin (ed.). *Automated Machine Learning. The Springer Series on Challenges in Machine Learning*. Cham: Springer International Publishing, 2019, s. 3-33. ISBN 978-3-030-05317-8. Dostupné z: https://doi.org/10.1007/978-3-030-05318-5_1. [cit. 2024-05-19].

- [49] *Tuner*. Online. Github. Dostupné z: <https://github.com/ultralitics/ultralitics/blob/main/ultralitics/engine/tuner.py>. [cit. 2024-05-21].
- [50] *Adam*. Online. Pytorch. Dostupné z: <https://pytorch.org/docs/stable/generated/torch.optim.Adam.html>. [cit. 2024-05-19].
- [51] MITCHEL, Melanie. *An Introduction to Genetic Algorithms*. Cambridge: MIT Press, 2002. ISBN 0-262-63185-7.
- [52] *How the Genetic Algorithm Works*. Online. Mathworks. Dostupné z: <https://www.mathworks.com/help/gads/how-the-genetic-algorithm-works.html>. [cit. 2024-05-23].
- [53] BERGSTRA, James a BENGIO, Yoshua. *Random Search for Hyper-Parameter Optimization*. Online. The Journal of Machine Learning Research. Č. 13. Dostupné z: <https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>. [cit. 2024-05-19].
- [54] *Random search*. Online. 2024. Dostupné z: https://en.wikipedia.org/w/index.php?title=Random_search&oldid=1222133590. [cit. 2024-05-19].
- [55] *Precision/Recall Tradeoff*. Online. Medium. 2020. Dostupné z: <https://medium.com/analytics-vidhya/precision-recall-tradeoff-79e892d43134>. [cit. 2024-05-19].
- [56] *DIGITAL IMAGE CORRELATION SOFTWARE*. Online. X-Sight. Dostupné z: <https://www.xsight.eu/digital-image-correlation/>. [cit. 2024-05-19].
- [57] YANG, Xue; YANG, Xiaojiang; YANG, Jirui; MING, Qi; WANG, Wentao et al. *Learning High-Precision Bounding Box for Rotated Object Detection via Kullback-Leibler Divergence*. Online. ArXiv.org. 2022. Dostupné z: <https://doi.org/10.48550/arxiv.2106.01883>. [cit. 2024-05-19].
- [58] DRUCKMÜLLEROVÁ, Hana. *Poznámky z aplikací Fourierovy analýzy*. Online, Skripta. Brno. Dostupné z: <https://math.fme.vutbr.cz/cz/ke-stazeni>. [cit. 2024-05-19].
- [59] *Precision and recall*. Online. In: . 2014. Dostupné z: <https://commons.wikimedia.org/wiki/File:Precisionrecall.svg>. [cit. 2024-05-18].
- [60] *What is Object Detection?* Online. Medium. 2020. Dostupné z: <https://medium.com/ml-research-lab/what-is-object-detection-51f9d872ece7>. [cit. 2024-05-18].
- [61] KREJSA, Jiří. *Umělá inteligence (přednáška)*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Květen 2023
- [62] DRUCKMÜLLER, Miroslav. *Technické aplikace metod umělé inteligence (přednáška)*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Duben 2022
- [63] *Multiclass Classification Using Support Vector Machines*. Online. Baeldung. Dostupné z: <https://www.baeldung.com/cs/svm-multiclass-classification>. [cit. 2024-05-19].

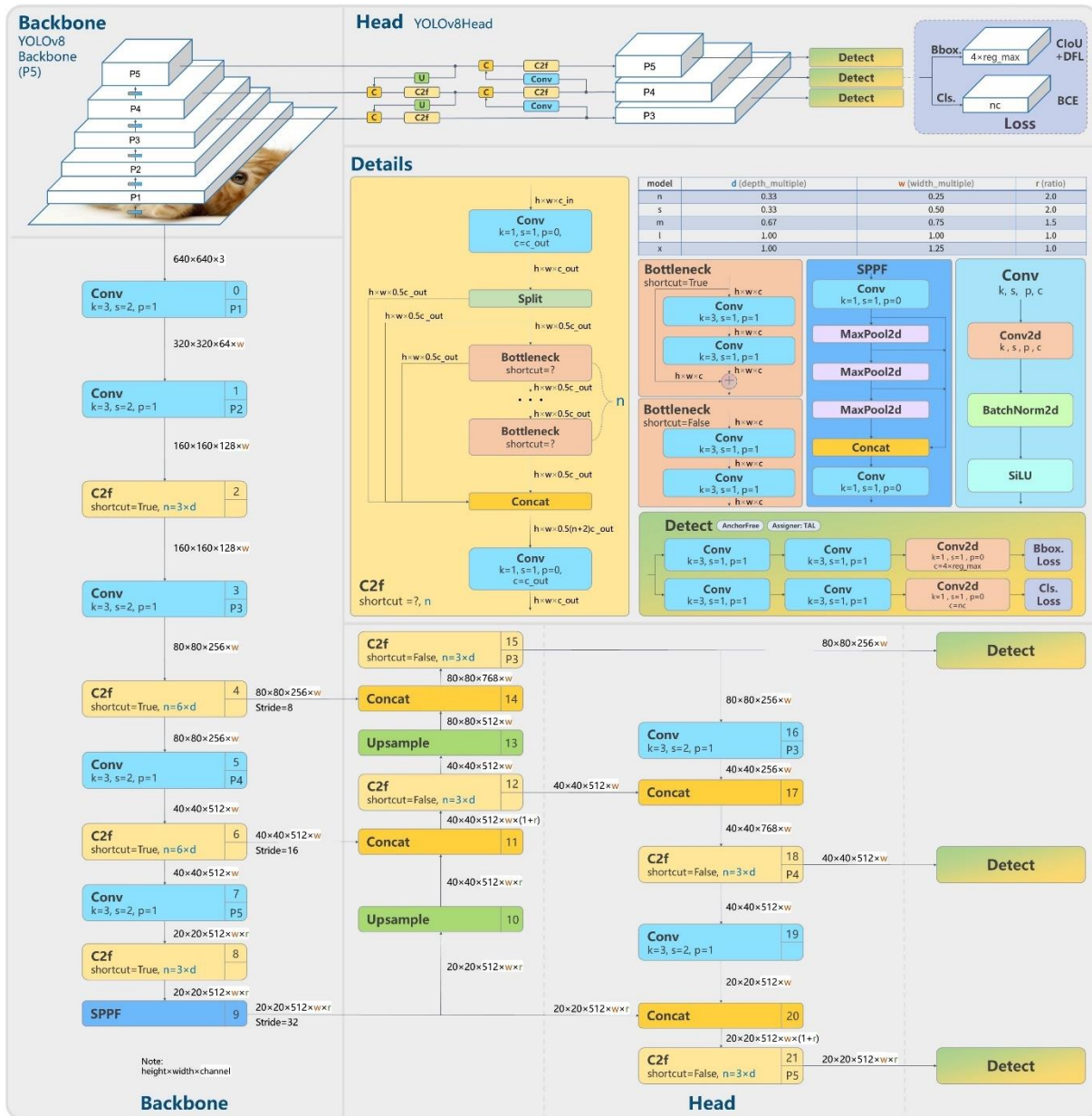
- [64] AL-BEHADILI, Husam; GRUMPE, Arne; DOPP, Christian a WOHLER, Christian. *Non-linear distance based large scale data classifications*. Online. In: *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)*. IEEE, 2015, s. 613-617. ISBN 9781467380867. Dostupné z: <https://doi.org/10.1109/PIC.2015.7489921>. [cit. 2024-05-19].
- [65] REZATOFIGHI, Hamid; TSOI, Nathan; GWAK, Junyoung; SADEGHIAN, Amir; REID, Ian et al. *Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression*. Online. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, s. 658-666. Dostupné z: <https://doi.org/10.1109/CVPR.2019.00075>. [cit. 2024-05-19].
- [66] *X-Sight: Advanced DIC measuring systems for optical investigation of your experiments* [online]. Brno: X-Sight s.r.o, 2023. Dostupné z: <https://www.xsight.eu/>. [cit. 2024-05-19]
- [67] International Digital Image Correlation Society. *A Good Practices Guide for Digital Image Correlation*. Online. 2018. Dostupné z: <https://doi.org/DOI:10.32720/idics/gpg.ed1/print.format>. [cit. 2024-05-20].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

Symbol	Veličina	Jednotka
<i>DIC</i>	Digitální korelace obrazu	
<i>ZNSSD</i>	Zero-mean normalized sum-of-square difference criterion	
<i>ZNCC</i>	Zero-mean normalized cross-correlation criterion	
<i>P</i>	Přesnost	-
<i>R</i>	Citlivost	-
<i>TP</i>	Skutečný pozitiv	
<i>FP</i>	Falešný pozitiv	
<i>IoU</i>	Intersection over Union	-
<i>RoI</i>	Regions of interest	
<i>S_{pred}</i>	Plocha určeného hraničního obdélníku	-
<i>S_{gt}</i>	Plocha skutečného hraničního obdélníku	-
<i>SIFT</i>	Scale Invariant Feature Transform	
<i>SURF</i>	Speeded Up Robust Features	
<i>SVM</i>	Support vector machine	
<i>RoI, ROI</i>	Regions of interest	
<i>y</i>	Výstup z neuronu	-
φ	Aktivační funkce	-
<i>w_i</i>	Váhy	-
<i>x_i</i>	Vstupy do neuronu	-
ϑ	Práh neuronu	-
<i>p</i>	Potenciál neuronu	-
<i>CNN</i>	Convolution neural network	
<i>MSE</i>	Mean squared error	-
<i>E_{loss}</i>	Chybová funkce	-
$\eta, lr0$	Learning rate	-
<i>SGD</i>	Stochastic gradient descent	
<i>MGD</i>	Mini-batch gradient descent	
<i>Adam</i>	Adaptive moment estimation	
α	Momentum	-
λ	Weight decay	-
β_1, β_2	Pokles setrvačnosti	-
<i>R-CNN</i>	Regions with CNN feature	
<i>NMS</i>	Non-max suppression	
<i>YOLO</i>	You only look once	
<i>CSP</i>	Cross-stage-Partial-connections	

<i>SPPF</i>	Spatial pyramid pooling fusion	
<i>PAN</i>	Pyramid attention network	
<i>FPN</i>	Feature pyramid network	
<i>COCO</i>	Common objects in context	
<i>AP</i>	Average precision	-
<i>mAP</i>	Mean average precision	-
<i>SOTA</i>	<i>state-of-the-art</i>	
x_s, y_s	Souřadnice středů hraničního obdélníku	-
w, h	Šířka a výška hraničního obdélníku	-
<i>CIoU</i>	Complete Intersection over Union	-
<i>BCE</i>	Binary cross entropy	-
<i>DFL</i>	Distribution Focal Loss	-
<i>lrf</i>	Činitel kroku učení	-
<i>GA</i>	Genetic algorithm	
<i>RS</i>	Random search	
<i>GS</i>	Grid search	
<i>BB</i>	Bounding box	
<i>OBB</i>	Oriented bounding box	
F_1	Harmonický průměr přesnosti a citlivosti	-
S	Střed hraničního obdélníku	-
W, H	Šířka a výška obrázku	-
B, B_1, B_2	Podstatné body odvozené ze souřadnic hraničního obdélníku	-

PŘÍLOHY



Úplná architektura YOLOv8 (převzato z [35])