VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

DETEKCE SEMI-STRUKTUROVANĚ ROZMÍSTĚNÝCH POLOTOVARŮ Z PLOCHY POMOCÍ METOD UMĚLÉ INTELIGENCE

DETECTION OF A SEMI-STRUCTURED SEMI-FINISHED PRODUCT FROM A DEFINED AREA USING ARTIFICIAL INTELLIGENCE METHODS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. Jan Zmrzlý
AUTHOR

VEDOUCÍ PRÁCE — Ing. Martin Juříček
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	**Bc. Jan Zmrzlý**
Studijní program:	Aplikovaná informatika a řízení
Studijní obor:	bez specializace
Vedoucí práce:	**Ing. Martin Juříček**
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:

Detekce semi–strukturovaně rozmístěných polotovarů z plochy pomocí metod umělé inteligence

Stručná charakteristika problematiky úkolu:

Práce bude zahrnovat rešerši v oblasti strojového vidění s důrazem na koncept Průmyslu 4.0. Teoretická část práce bude také zahrnovat rešerši dostupných metod pro detekci semi-strukturovaně rozmístěných objektů z plochy. Předmětem práce bude rešerše současného stavu robotické buňky EDUset ONE s důrazem na řešení problému strojového vidění, výběr HW řešení (kamera, osvětlení, apod.), návrh a implementace různých metod pro detekci více druhů objektů semi-strukturovaně rozmístěných na ploše s důrazem na modulárnost, efektivitu a přesnost. Závěr práce bude věnován porovnání navržených metod pro řešení problému a ověření funkčnosti navrženého řešení v interakci s reálnou robotickou buňkou.

Práce předpokládá aktivní přístup studenta a nutnost práce v laboratoři. Práce také předpokládá základní znalost strojového učení, OOP (zejména v jazyce Python 3+) a pokročilou znalost matematiky. Možnost psát práci v CZ, SK a EN (doporučeno).

Práce bude realizována ve spolupráci s firmou Intemac Solutions s.r.o., která poskytne řídicí prostředky a část know-how v pojednávané oblasti.

Cíle diplomové práce:

– Proveďte rešerši v oblasti strojového vidění s důrazem na koncept Průmyslu 4.0.
– Analyzujte dostupné metody pro problém detekce semi-strukturovaně rozmístěných objektů z plochy.
– Proveďte rešerši současného stavu robotické buňky EDUset ONE s důrazem na problém strojového vidění.
– Navrhněte HW řešení (kamera, osvětlení, apod.) pro řešení problému.
– Navrhněte a implementujte různé metody pro detekci více druhů objektů semi-strukturovaně rozmístěných na ploše.
– Porovnejte navržené metody pro řešení problému a proces vyhodnoťte.
– Ověřte funkčnost vytvořeného řešení v interakci s reálnou robotickou buňkou.

Seznam doporučené literatury:

SICILIANO, Bruno a KHATIB, Oussama, ed. Springer handbook of robotics. 2nd edition. Berlin: Springer, 2016. ISBN 978-3-319-32550-7.

Elgendy, M., Deep Learning for Vision Systems. Manning Publications, 2020.

Raut, R., Krit, S. and Chatterjee, P., Machine Vision for Industry 4.0. 1st edn. CRC Press, 2022.

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

V Brně, dne

L. S.

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

ABSTRAKT

Tato práce se zabývá problematikou strojového vidění v kontextu Průmyslu 4.0, s důrazem na detekci semi-strukturovaně rozmístěných objektů z plochy. V první části práce jsou probrány teoretické aspekty úlohy, včetně vybraných algoritmů strojového vidění a využití neuronových sítí v této oblasti. Dále je provedena rešerše dostupných metod pro řešení tohoto problému, jakož i současného stavu robotické buňky *EDUset ONE* s ohledem na strojové vidění. Na základě analýzy je navrženo hardwarové řešení v podobě kamery, osvětlení a dalších komponent. Následně je proveden návrh a implementace různých metod detekce více druhů objektů, s důrazem na modularitu, efektivitu a přesnost. V závěru práce jsou tyto metody porovnány a je ověřena funkčnost v interakci s reálnou robotickou buňkou.

ABSTRACT

This thesis addresses the issue of machine vision in the context of Industry 4.0, with an emphasis on the detection of semi-structured objects from surfaces. The first part of the thesis discusses the theoretical aspects of the task, including selected machine vision algorithms and the use of neural networks in this area. Furthermore, a survey of the available methods for solving this problem is conducted, as well as the current state of the art of the EDUset ONE robotic cell with respect to machine vision. Based on the analysis, a hardware solution in the form of camera, lighting and other components is proposed. Subsequently, the design and implementation of different methods for detecting multiple types of objects is carried out, with emphasis on modularity, efficiency and accuracy. Finally, the work compares these methods and verifies their functionality in interaction with a real robotic cell.

KLÍČOVÁ SLOVA

robotická buňka, EDUset ONE, detekční framework, detekce objektů, porovnání metod, YOLO, SSD, R-CNN, strojové vidění, výběr komponent, OPC UA

KEYWORDS

robotic cell, EDUset ONE, detection framework, object detection, method comparison, YOLO, SSD, R-CNN, machine vision, component selection, OPC UA

ÚSTAV AUTOMATIZACE A INFORMATIKY

2024

BIBLIOGRAFICKÁ CITACE

ZMRZLÝ, Jan. *Detekce semi-strukturovaně rozmístěných polotovarů z plochy pomocí metod umělé inteligence.* Brno, 2024. Dostupné také z: https://www.vut.cz/studenti/zav-prace/detail/157232. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Martin Juříček.

PODĚKOVÁNÍ

Chtěl bych touto cestou poděkovat panu Ing. Petru Janusovi a Ing. Vojtěchu Markovi z firmy Intemac Solutions s.r.o. za jejich cennou pomoc při tvorbě této závěrečné práce. Velké uznání patří také Ing. Romanu Parákovi a Ing. Martinu Juříčkovi za jejich neocenitelné rady a připomínky, které významně přispěly k posunu práce kupředu. Děkuji rovněž své rodině a blízkým za jejich podporu během mého studia.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 20. 5. 2024 ……

Jan Zmrzlý

OBSAH

1 ÚVOD 15
2 STROJOVÉ VIDĚNÍ 17
2.1 Průmysl 4.0 17
2.1.1 OPC Unified Architecture 19
2.2 Zrakové vnímání 20
2.2.1 Systémy vidění 20
2.2.2 Humanoidní systém vidění 21
3 ALGORITMY STROJOVÉHO VIDĚNÍ 23
3.1 Reprezentace obrázků v počítači 23
3.2 Detekce hran 23
3.2.1 Klasické metody 24
3.2.2 Metody hlubokého učení 25
3.3 Extrakce příznaků 25
3.4 Analýza hlavních komponent 26
3.5 Ohraničení nejmenším obdélníkem 28
4 ALGORITMY VYUŽÍVAJÍCÍ NEURONOVÉ SÍTĚ 31
4.1 Konvoluční neuronové sítě 31
4.2 Augmentace dat 33
4.3 Transfer learning 34
4.4 Frameworky pro detekci objektů 35
4.5 Rodina sítí R-CNN 36
4.6 Single Shot Detector 37
4.7 Rodina sítí You Look Only Ones 38
4.7.1 YOLOv8 39
4.8 Srovnání jednotlivých modelů 39
5 ROBOTICKÁ BUŇKA EDUSET ONE 43
5.1 EDUset ONE 43
5.2 Volba senzoru pro sběr dat 44
5.3 Možnosti umístění kamery v buňce 45
5.4 Zavedení nové konstrukce pro strojové vidění 46
5.5 Výběr vizuálního systému 47
5.6 Polotovary určené pro k detekci 47
5.7 Výběr kamery a objektivu 48
5.8 Výběr osvětlení a pozadí 50

6 VÝVOJ DETEKČNÍHO SOFTWARU 53
6.1 Tvorba vlastního datasetu 54
6.2 Knihovny pro implementaci v jazyku Python 55
6.3 Implementace a porovnání detekčních frameworků 57
6.3.1 Podpora vývojářské komunity a dostupnost modelů 58
6.3.2 Jednoduchost implementace a flexibilita 58
6.3.3 Výkony frameworků na vlastním datasetu 59
6.3.4 Shrnutí porovnání detekčních frameworků 60
6.4 Strategie určení polohy polotovaru 61
6.5 Komunikace prostřednictvím OPC UA 63
7 ZHODNOCENÍ A PŘIPOMÍNKY KE ZLEPŠENÍ 65
7.1 Možnosti navázání na práci 66
8 ZÁVĚR 69
SEZNAM POUŽITÉ LITERATURY 71
SEZNAM ZKRATEK A SYMBOLŮ 75
SEZNAM OBRÁZKŮ 77
SEZNAM TABULEK 79
SEZNAM PŘÍLOH 81

1 ÚVOD

V dnešní době se zvyšuje pozornost věnovaná automatizaci a optimalizaci průmyslových procesů, což vede k rostoucímu významu oblasti strojového vidění. Tato oblast, dříve považovaná za vědeckou fikci, se stále více stává nedílnou součástí moderního průmyslu, zejména v rámci Průmyslu 4.0. Diplomová práce je pečlivě zaměřena na analýzu a implementaci moderních algoritmů strojového vidění, s důrazem na jejich aplikaci v průmyslovém prostředí, konkrétně v projektu robotické buňky *EDUset ONE*.

První část práce je věnována teoretickým základům strojového vidění a jeho roli v Průmyslu 4.0. Zde jsou pečlivě zkoumány základní koncepty zpracování obrazu, které výrazně napomohou při tvorbě detekčního softwaru. Sleduje se vývoj od tradičních metod k moderním technikám založeným na neuronových sítích, které přinášejí revoluci v přesnosti a efektivitě. Následuje analýza konkrétních algoritmů strojového vidění. S rostoucím zájmem o hluboké učení se představují moderní přístupy, včetně konvolučních neuronových sítí a jejich aplikací v detekci objektů a segmentaci.

Následující část práce je detailně věnována aplikaci algoritmů založených na metodách umělé inteligence v projektu robotické buňky *EDUset ONE*. Jsou zde prezentovány jednotlivé milníky projektu, včetně výběru senzorů, strategií umístění kamer a optimalizace osvětlení pro efektivní aplikaci strojového vidění. Dále se provádí porovnání frameworků využívajících neuronové sítě, mezi něž patří SSD, R-CNN a YOLO. Také jsou popsány postupy vývoje detekčního softwaru a strategie určení polohy detekovaných objektů.

V závěrečné části práce jsou prezentovány dosažené výsledky a provedeno jejich zhodnocení. Diskutuje se o možnostech dalšího využití a rozvoje aplikací strojového vidění v průmyslovém prostředí a navrhují se možná zlepšení v implementaci a využití těchto technologií, otevírajíce tak cestu pro další pokrok v oblasti strojového vidění a jeho aplikací v průmyslové automatizaci.

2 STROJOVÉ VIDĚNÍ

Obor strojového vidění se velice perspektivně rozvíjí během posledních dvaceti let. Za tyto roky jde pokrok nazvat pravou revolucí, která radikálně změnila způsob, jakým se dříve přistupovalo k řešení mnoha věcí souvisejících s oborem. Společně s vývojem přinesla i mnoho vzrušujících příležitostí pro ty, kteří si nové techniky rychle osvojí a zvládnou je efektivně implementovat. Rozvoj jde ruku v ruce s vývojem výpočetního hardwaru, který je vybaven grafickými procesory. Současná situace nabízí širokou škálu aplikací strojového vidění – doposud extrémně náročných nebo v mnoha případech dříve nemožných automatizačních úloh. Mnohými jsou skoky spojené se strojovým viděním, internetem věcí (IoT) a dostupností velkých dat (*Big Data*) označovány jako čtvrtá průmyslová revoluce [1]. Zmíněná transformace dramaticky zvýšila efektivitu stávajících aplikací strojového vidění, například ve výrobě (kontrola a zajištění kvality, detekce přítomností komponent, bezpečnost operátorů), medicíně (detekce rakoviny, klasifikace histologických snímků, detekce poškození měkkých tkání) [2]. Naopak v jiných oborech (farmacie, zemědělství, stavebnictví) zcela zpřístupnila nové oblasti využití.

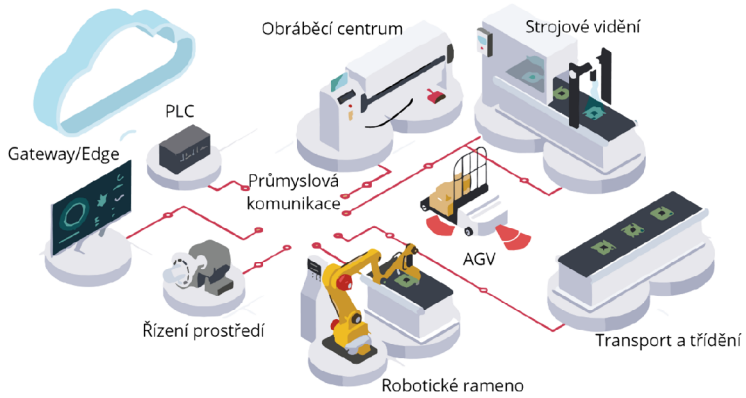
Nejprve je důležité uvést pojem strojového vidění na pravou míru a správně jej vymezit. Strojové vidění představuje metodu řešení průmyslových výzev, která si klade za cíl snížit náklady, zvýšit produktivitu a poskytovat důležitá data [1]. Tento přístup také nabízí možnost nahradit méně dostupnou kvalifikovanou pracovní sílu nebo uvolnit pracovníky z nebezpečných nebo namáhavých pracovních pozic. V dnešní době se strojové vidění chápe jako praktická implementace schopnosti porozumění obrazu, což zahrnuje soubor technik zaměřených na řešení praktických průmyslových problémů, které mají významný vizuální aspekt.

V současnosti je kladen velký důraz na propojení strojového vidění se strojovým učením, které se taktéž velmi radikálně podílelo na tvoření oboru. Zapříčiněno to bylo rychlým vývojem umělé inteligence, která následně posloužila, aby se některé schopnosti strojového vidění více přiblížili vlastnostem lidského zraku. Jde o nenaplněnou ambici, které existuje již od 60. let 20. století [1; 2].

2.1 Průmysl 4.0

Termín s označením Průmysl 4.0 již několik let rezonuje technickou společností a stále se jedná o jeden z nejdiskutovanějších pojmů. Obecně Průmysl 4.0 odkazuje na několik poměrně nových technologií, které mohou zásadně změnit výrobu. Mezi nové technologie patří: robotika, umělá inteligence, strojové vidění, velká data a cloudové výpočty [3]. Z názvu je zřejmé, že se jedná o čtvrtou průmyslovou revoluci, která přináší zlepšení udržitelnosti, snižuje znečištění a zvyšuje účinnost výrobků. Strojové vidění je klíčovou součástí této revoluce, představující dynamickou strategii chytré továrny, která umožňuje výpočetní technice vnímat okolní prostředí. Systém strojového vidění usnadňuje automatizovanou kontrolu, detekci a revizi [2]. V možných aplikacích výborně

nahrazuje úkony, které byly historicky založeny na lidském posouzení. Nově příchozí systémy strojového vidění dokáží hlídat rozměry, udržovat konstantní výrobu a eliminovat odpadový materiál [3]. Je patrné, že technologie umožňuje společnosti plně využít trvanlivých výhod zvýšené přesnosti ve výrobě.



Obr. 1: Průmysl 4.0 v provedení moderního výrobního závodu [4]

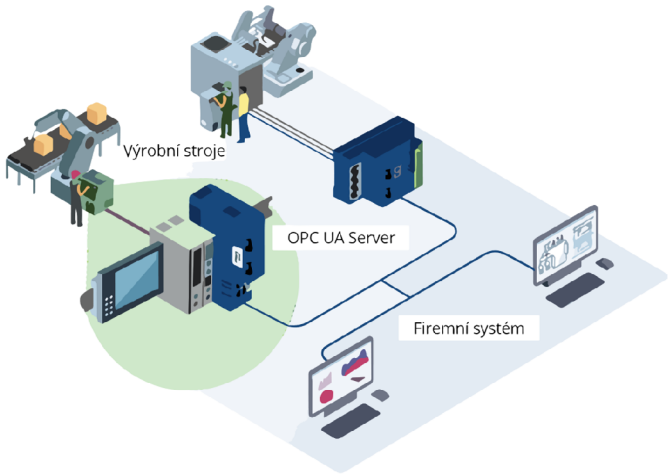
Stejně jako dříve i poslední průmyslová revoluce formuje společnost, ve které lidé žijí. První průmyslová revoluce přinesla stroje poháněné vodou, což vytvořilo vhodné podmínky pro velkovýrobu [3]. Druhé revoluce je charakterizovaná přechodem z vody na elektřinu. Třetí průmyslová revoluce přinesla rozvoj elektroniky, počítačů a programovatelných automatů (PLC), což vedlo k automatizaci výrobních linek a složitých procesů. Průmysl 4.0 zlepšuje automatizaci továren s přesnějšími daty a strojovým učením. Jeho přední myšlenky aplikované ve výrobě jsou na obr. 1. Vidění skrze kameru je velmi důležité při sběru dat z fyzického světa a přeměně na digitální data. Získaná data lze pak vyhodnocovat a zpracovávat v reálném čase, aby se zvýšila provozní účinnost. Automatizace v rámci Průmyslu 4.0 propojuje stroje a vytváří tak ekosystém známý jako Internet věcí (IoT) [3]. Integrace strojového vidění s automatizačními systémy zlepšuje kvalitu a množství sbíraných dat, což umožňuje vytvoření chytré továrny.

Z přechozího textu je nadmíru patrné, že oblastí aplikace strojového vidění je opravdu nepřeberné množství. Spolu s umělou inteligencí také výrazně přispívá ke splnění cílů poslední průmyslové revoluce. Továrny produkují udržitelnější výrobky v kratším čase, dokáží pracovat nepřetržitě, nahrazují ubíjející repetitivní lidskou činnost [3]. Výčet všech přínosů by mohl zabrat několik dalších stran, je tedy na místě vyvodit, že strojové vidění je nedílnou součástí Průmyslu 4.0 a stojí za jeho rozvojem. Jedná se také o trh s rostoucím potenciálem a nadějnou budoucností, přičemž jeho jednoduché použití hraje klíčovou roli. Odvětví, která tento trend již přijala, získávají výhody, zatímco konkurenti, kteří zůstávají pozadu, zaostávají.

2.1.1 OPC Unified Architecture

Klíčovou roli v Průmyslu 4.0 hraje komunikace mezi jednotlivými zařízeními propojenými ve výrobním závodě. Velká touha, ze strany manažerů, je získávat, co nejvíce dat, která mohou být následně využita například pro strojové učení. Tento trend neusnadňuje rozmanitost mechatronických systémů, automatizačních nástrojů a komunikačních systémů. Zásadní úlohu ve zmíněném procesu hraje OPC UA (*Open Platform Communication Unified Architecture*), což je v průmyslové automatizaci široce uznávaný standard pro výměnu dat [5]. Data lze díky OPC UA sdílet výrobním závodem vertikálně, což zahrnuje senzory, akční členy přes řídící a komunikační systémy až po centrální servery a cloudová úložiště.

Průmyslový standard OPC UA je poskytován organizací *OPC Foundation*. Zpočátku se OPC UA zaměřovalo pouze na potřeby průmyslové automatizace, ale jelikož je nezávislý na dodavatelích a operačních systémech, může se stát rychle nepostradatelným i pro další oblasti využití [5]. Všechny zmíněné výhody přispěly k odstranění překážek integrace v malých a středních podnicích. Integrace komunikačního standardu je zobrazena v rámci obr. 2.



Obr. 2: Průmyslová standardizace OPC UA v malém podniku [6]

Specifikace OPC UA funguje na principu přenosu dat mezi OPC UA klientem a OPC UA serverem prostřednictvím mapování a navazování spojení, zabezpečení komunikace a struktury odeslaných dat [7]. OPC UA protokol je definován jako služebně orientovaná architektura SOA (*Service Oriented Architecture*), což znamená, že jsou definovány služby, na něž se klient může dotazovat, a server na každý dotaz reaguje příslušnou odpovědí. Služby poskytované serverem vytvářejí abstraktní komunikační model. Po navázání fyzického spojení se pomocí těchto služeb vytváří a udržuje zabezpečený kanál (*SecuredChannel*) a relace (*Session*). Aktivní zabezpečený kanál je nezbytný pro veškerou komunikaci, zatímco relace je nutná pro klientovo dotazování se

na služby serveru. Tyto části komunikace jsou často označovány společně jako komunikační zásobník (*Stack*). Adresní prostor je popsán pomocí uzlů.

Zabezpečení v rámci OPC UA je navrženo s ohledem na moderní hrozby kybernetických útoků a zahrnuje jak vnější, tak vnitřní opatření. Vnější zabezpečení může zahrnovat šifrování a digitální podpis zpráv nebo využití zabezpečeného kanálu pro přenos dat [7]. Vnitřní bezpečnostní opatření zase zahrnují uzavírání nadbytečných spojení a monitorování aktivity na serveru prostřednictvím auditu. Každá aplikace účastnící se komunikace pomocí OPC UA (klient nebo server) musí mít svůj vlastní aplikační certifikát, který jednoznačně identifikuje danou aplikaci a zařízení, na němž tato aplikace běží.

Díky standardu OPC UA je možné vyvíjet například aplikace strojového vidění, které nejsou vázané na průmyslový automat PLC. To otevírá cestu k využití vyšší objektově orientovaných programovacích jazyků pro vývoj těchto aplikací, což zlepšuje flexibilitu a možnosti implementace. Různí výrobci PLC používají vlastní komunikační protokoly, které nejsou vzájemně kompatibilní.[1] Proto je možné vyvíjet aplikace například v jazyce *Python*, které jsou schopné komunikovat s libovolným PLC pomocí klienta. Podmínkou je pouze, aby dané PLC disponovalo vestavěným OPC UA serverem, což se stává stále častějším standardem v průmyslových zařízeních. Tato integrace umožňuje širší rozsah možností při vývoji a implementaci aplikací strojového vidění v průmyslovém prostředí.

2.2 Zrakové vnímání

Zrakové vnímaní je v nejzákladnějším slova smyslu pozorování vzorů a objektů zrakem. V případě autonomního vozidla znamená vizuální vnímaní pochopení okolních objektů a jejich specifických detailů [8]. Po automobilu je požadováno rozpoznání chodců, detekce jízdních pruhů, detekce dopravních značek a mnoho dalšího. V případě automobilu není potřebné získat jen vjem, ale je taky nutné pochopit jeho význam. Není snahou zachytit pouze okolní prostředí, ale vytvořit systémy, které budou okolnímu prostředí skutečně rozumět na základě vizuálních vstupů.

2.2.1 Systémy vidění

V počátcích strojového vidění byly klasické techniky považovány za systémy strojového vidění, ale označení není v dnešní době vůbec přesné [8]. Stroj, který pouze zpracovává obraz, není srovnatelný se strojem, který obraz zpracuje a rozumí tomu, co se v něm děje. Jde o úkol, který není ze své podstaty absolutně triviální. Zpracování obrazu je jen podsystémem složitějšího celku, jehož hlavním cílem je interpretovat obsah obrazu.

[1] Protokoly jednotlivých výrobců nebudou blíže specifikovány, jelikož jejich využití nepřispělo k vypracování práce. Uvedenými zástupci mohou být: *Beckhoff Automation (EtherCAT), Mitsubishi Electric (CC-Link), Siemens (PROFINET)*. Společnost *Siemens* byla hlavním investorem při vzniku OPC UA.

2.2.2 Humanoidní systém vidění

Evolučně nejlépe vyvinutý zrak mají lidé, hmyz a většina živých organismů. Zrak se z hlediska technické praxe skládá ze senzoru, vodičů a řídícího počítače [8]. Senzorem je oko, které zachycuje okolní obraz. Za vodiče jdou pokládat nervy, jenž mají za úkol přenos informace, ale také cévy, které senzor napájí. Posledním článkem je mozek, který obraz zpracovává a interpretuje. Systém pak na základě údajů získaných z obrazu vyprodukuje předpověď jeho složek.

Je předpokládáno, že má být interpretovaný obr. 3 skupiny horolezců zdolávající horu *Ama Dablam*. Obyčejný člověk se podívá na fotku a pochopí, že na obrázku jsou zachyceni dva horolezci. Člověk, s vášní pro zdolávání nejvyšších vrcholků planety, zjistí nejen, že se jedná o skupinu horolezců. Určí také, jaké mají vybavení nebo zná dokonce název hory, jejíž vrchol se pokouší zdolat. Klasifikace a detekce objektů je pro člověka zcela přirozená a je dána skutečností, že byl léta trénován v rozpoznání hor a svých druhů pohybujících se v nich.

Nyní by bylo vzato v potaz, že člověk nikdy dříve neviděl horolezce. Je mu tedy poprvé ukázán jeho obrázek a on není schopen identifikovat, co je smyslem obrázku. Pokračuje se předkládáním dalších obrázků. Po několika takových pokusech dokáže osoba zcela přesně identifikovat horolezce. Nyní je subjektu předložen obr. 4 lyžaře. Ten se na obrázek podívá a začne s identifikací: člověk, teple oblečený, na sněhu, v horách. Teoretiky by to mohl být horolezec. Nicméně se jedná o zmíněného lyžaře. Jeho mozek pak, upraví některé parametry svého algoritmu, aby se naučil rozeznávat rozdíly mezi lyžařem a horolezcem. Nyní je jeho mozek vycvičen na klasifikaci dvou těchto dvou osob.

Obr. 3: Výstup na Ama Dablam [9]

Obr. 4: Lyžař v horách při sjezdu

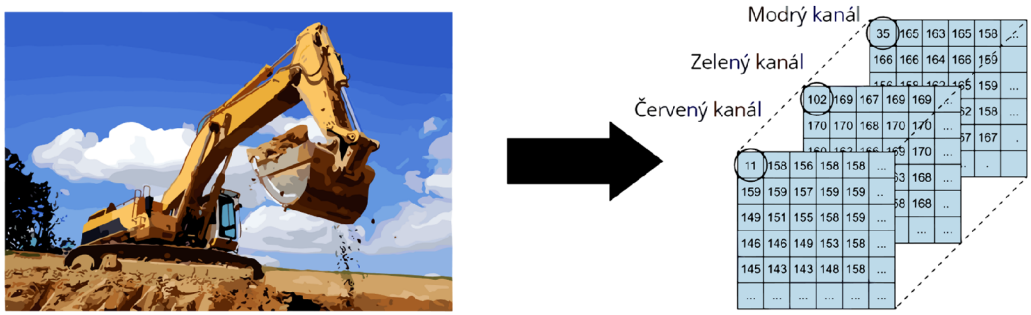
Podobné chování sledujeme i u počítačů. Stroje je možné vycvičit, aby určily a identifikovaly většinu objektů [8]. Avšak rozdíl spočívá v množství dat. Člověku může stačit pouze několik málo obrázků k tomu, aby byl schopen s vysokou pravděpodobností určit daný předmět. Na druhou stranu strojům je potřeba předložit stovky až tisíce obrázků, aby se naučily daný objekt rozpoznat. Ve své podstatě jde prohlásit, že se vědci inspirovali lidským zrakovým systémem a v posledních letech se jim podařilo úspěšně zdokonalit zrakové schopnosti strojů [8].

3 ALGORITMY STROJOVÉHO VIDĚNÍ

Aplikace strojového vidění fungující na bázi strojového učení jsou velmi dobře zavedené v průmyslové výrobě již několik let. Nicméně před příchodem umělé inteligence byly velké snahy, aby stroje efektivně porozuměly obrazu. Velké množství technik, které byly vyvinuty se používají i v nejmodernějších algoritmech. Velká část postupů našla uplatnění v předzpracování obrazu. Klasických algoritmů a přístupů bylo zavedeno mnoho a není nutné všechny uvádět v této práci, ale nezbytné je uvést ty, které ve značné míře pomohly k dosažení cílů publikace.

3.1 Reprezentace obrázků v počítači

Pokud se člověk dívá na obrázek, pak vidí předměty, krajinu, zvířata. Tak to ale nefunguje v případě počítače. Pro něj je nejjednodušším černobílým obrázkem 2D matice, jejíž prvky jsou hodnoty pixelů [8; 10]. Pixely reprezentují intenzitu v celém barevném spektru. Matice není doplněna o žádný kontext, jsou zde jen surová data v podobě čísel. Posuzován je monochromatický obrázek focený v kvalitě 4K tzn. 3840 x 2160 pixelů. K dispozici je tedy 8294400 hodnot, které představují intenzitu jasu v každém pixelu. Nula představuje černou a 255 bílou [8; 10]. V obrazech ve stupních šedi představuje každý pixel intenzitu pouze jedné barvy, zatímco ve standardním systému RGB mají barevné obrazy tři kanály (červený, zelený a modrý). Praktická realizace je taková, že barevné obrazy obsahují tři matice. První matice určuje hodnoty červené barvy, druhá matice vymezuje intenzitu zelené barvy a poslední determinuje barvu modrou. Barevný obrázek se tedy bude skládat ze tří kanálů s tím, že reprezentace v počítači je jako 3D matice, která je vykreslena na obr. 5.



Obr. 5: Obrázek v počítači zobrazen jako 3D matice s kanály RGB [8; 11]

3.2 Detekce hran

Velmi často používanou operací při zpravování obrazu je detekování hran. Jde o proces, který se pokouší charakterizovat změny v obraze z hlediska fyzikálních procesů, které je

vyvolaly [12]. Tato technologie se široce používá při segmentaci obrazu, detekci oblastí zájmu a dalších technologiích zpracování obrazu na vysoké úrovni. Stěžejním úkolem je detekce a charakteristika významných změn intenzity. Existuje řada klasických postupů využívajících filtraci s pomocí konvolučních jader. V nedávné době výzkumníci navrhli řadu algoritmů detekce hran založených na hlubokém učení. Z výsledků zkoumání je patrné, že výkonost současných algoritmů detekce hran se přibližuje úrovni lidského zraku, nebo ji dokonce překračuje [13]. Cílem následujícího textu není rozklíčovat mnohé přístupy při hledání hran, ale spíš přiblížit fungování základních algoritmů pro jejich hledání.

3.2.1 Klasické metody

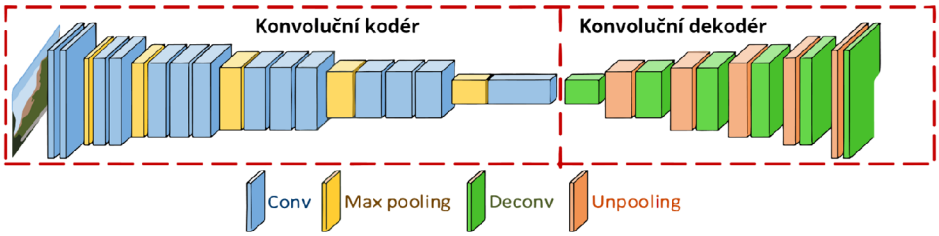
Jak už bylo zmíněno klasické metody byly navrženy podstatně dříve něž metody založené na hlubokém učení. Lze je tedy považovat za vyzrálejší, jednodušší, ale zato velmi účinné. I když se zdá, že přístup pomocí hlubokého učení překoná v mnohých aspektech ve všech oborech své předchůdce, není tomu tak. Klasické metody jsou stále zdokonalovány, je stále velká snaha jejich nedostatky odstranit a zlepšit jejich výkonost [13]. Klasické přístupy jde rozdělit do čtyř hlavních kategorií: metody založené na změně gradientu, na základě Gaussovy diference, za základě více škálových příznaků a na základě strukturovaného učení.

Nejvíce přímočarou metodou je detekování hran na základě změny gradientu. Rozumí se tím náhlá změna mezi sousedními hodnotami pixelů obrazu, kde se výrazně projeví rozdíl mezi dvěma hodnotami pixelů. Detekce hran tedy spočívá v určení a generování gradientu hran obrazu pomocí vhodného konvolučního jádra [12; 13]. Jádro lze označit také jako maska nebo operand. Nejvíce používané operandy lze rozdělit na: prvního řádu (Sobelův, Prewittův a Robertův) a druhého řádu (Laplaceův, LOG a optimální Cannyho operátor). Detekované hrany pomocí Sobelovy masky jsou na obr. 6.

Obr. 6: Detekované hrany na obrázky lyžaře pomocí Sobelovy masky

3.2.2 Metody hlubokého učení

S rychlým rozvojem metod hlubokého učení se vyvíjí i metody detekce hran na nich založené. V posledních letech tak vznikají v oblasti digitálního zpracování obrazu různé algoritmy. Kvůli mnoha faktorům, jako je přesnost lokalizace, přesnost detekce hran, citlivost na šum a také kvůli různé přesnosti detekčních algoritmů se však metody neustále zdokonalují [13]. Přístupy využívající principu strojového učení lze rozdělit do tří hlavních typů: metody založené na základě kodeku (*codec-based*), na základě rekonstrukce sítě (*reconstruction-based*) a na základě fúze více škálových prvků (*multi-scale feature fusion-based*).



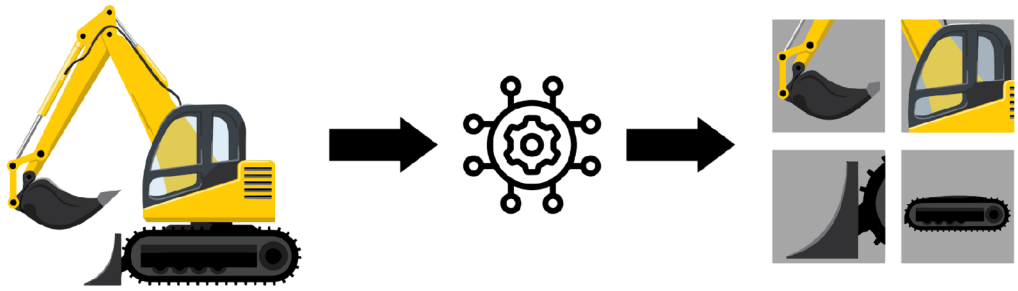
Obr. 7: Architektura konvoluční sítě CEDN využívaná pro detekci hran [13]

Jako nejvhodnější zástupce byla zvolena metoda na základě kodeku (*codec-based*). Vzhledem ke skutečnosti, že konvoluční sítě zmenšují velikost obrazu po vícenásobných konvolucích, neodpovídá konečný výstup každému pixelu původního obrazu. Řešením se stalo zavedení kodeků, které mohou přijímat vstupní obrazy libovolné velikosti a vytvářet výstupní obrazy stejné velikosti. Úkolem kódovací sítě je vytvářet příznakové obrazy se sémantickou informací. Zatímco úkolem dekódovací sítě je mapovat příznakový obraz s nízkým rozlišením, který je výstupem kódovací sítě, zpět na velikost vstupního obrazu pro klasifikaci pixel po pixelu [13]. Zástupci těchto principů jsou sítě: CEDN, CASENet a nejmodernější RINDNet. Na obr. 7 je architektura moderní konvoluční sítě pro detekci.

3.3 Extrakce příznaků

Extrakce příznaků je klíčovou součástí strojového vidění. V podstatě lze konstatovat, že celý proces hlubokého učení pracuje na bázi extrakce užitečných rysů, které zřetelně definují objekty v obraze [8]. Příznakem se rozumí měřitelný údaj na obrázku, který je zcela jedinečný pro daný objekt. Takovými příznaky jsou například barva, hrana, segment obrazu. Naopak drobný rys slouží k odlišení objektů od sebe navzájem. Pokud je jako rys získáno kolo a následně se má rozhodnout zda se jedná o bagr nebo horolezce, je zcela jasné, že v tomto konkrétním případě půjde spíše o bagr. Ovšem pokud je znovu jako rys získáno kola a má být rozhodnuto zda se jedná o auto nebo bagr, pak není tento rys dostatečně silný, aby mohly být objekty jasně rozlišeny. Je nutné hledat další znaky, které společně definují jeden objekt. V případě bagru to bude například radlice, kabina, pásový

pohon, v případě horolezce batoh, lano, cepín. Ve strojovém učení je nutné převést surová data na vektor příznaků, jenž je předložen učícímu algoritmu [8]. Obecně platí, že výsledný model je tak dobrý, jak dobré jsou mu předkládány příznaky při učení.



Obr. 8: Extrakce rysů z obrázku bagru

V původních klasických procesech byly příznaky vybírány manuálně. Člověk tedy musel strávit hodiny hledáním kvalitních příznaků. Většinou bylo nutné spolupracovat s odborníky z konkrétní oblasti nebo se spoléhat na svou orientaci v problematice. Velkým skokem jsou algoritmy hlubokého učení, kde odpadá nutnost extrahovat příznaky ručně [8]. Samotná síť dokáže automaticky příznaky z obrázků vytáhnout a určit jejich důležitost tím, že na výstup aplikuje váhy. Zmíněné sítě si lze představit jako balíček, obsahující extraktory příznaku a klasifikátory. Lze je tedy trénovat od začátku do konce, na rozdíl od klasických přístupů strojového učení, které vyžadují ručně vytvořené příznaky.

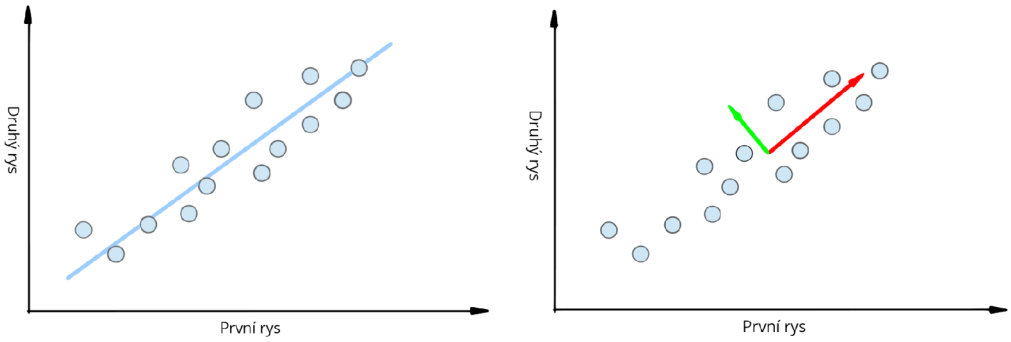
Nyní by mohla padnout otázka, proč je vůbec nutné mít postup, který z obrázku vytáhne příznaky. Nedělá to práci složitější než být musí. Odpověď je jednoduchá. Důvodem je, že vstupní obrázek obsahuje příliš mnoho informací z nichž je velká část zbytečná [8]. Proto je zcela logickým krokem zjednodušení dat pomocí extrakce důležitých informací. Extrakcí důležitých barev nebo obrazových segmentů je možné dosáhnout zmenšení vstupních dat na menší množiny znaků. Tím se celý proces klasifikace zjednoduší a urychlí.

3.4 Analýza hlavních komponent

Analýza hlavních komponent (*Principal Component Analysis*) je oblíbenou statistickou technikou pro parametrizaci tvaru, vzhledu a pohybu v obrázcích. PCA se ukázala jako vhodná technika pro aplikace strojového vidění [14]. K nim náleží například rozpoznání objektů a obličejů. Lze také použít ke snížení dimenze dat s co nejmenší ztrátou informace [15]. Analýza hlavních komponent má mnoho využití, následně bude uvedeno využití, které napomohlo ke splnění cílů práce.

PCA pomáhá najít nejvýznamnější rys v souboru dat a usnadňuje vykreslení dat ve 2D a 3D prostoru. Spočívá v nalezení os v euklidovském prostoru [16]. Soubor dat si lze představit jako mrak bodů ve 2D grafu. Úkolem je stanovit první osu tak, aby byl rozptyl maximální. Pokud je blíže zkoumán obr. 9, pak je zjištěno, že existuje přímka, kterou

mohou být data proložena. Provedením PCA na souboru datových bodů jsou získány dvě hlavní komponenty, které se označují jako vlastní vektory [17]. Jsou to přímky, které zachycují rozptyl dat a obsahují informace o směru a velikosti datových bodů. Počátek vlastních vektorů je středem všech bodů v mraku dat.



Obr. 9: Vizualizace mraku dat v 2D grafu, určení os dat pomocí PCA [17]

Nyní je odvozeno, jak lze určit vlastní vektory a vlastní hodnoty [17]. Cílem je převést soubor dat X o rozměru p na jiný soubor dat Y o menším rozměru L. Je hledána matice Y, kde Y je Karhunenova-Loèvova transformace (KLT) matice X:

$$Y = KLT\{X\}. \tag{1}$$

Je předpokládáno, že je k dispozici soubor pozorování p a má být redukován tak, aby každé pozorování bylo možné pospat pouze pomocí proměnných L. S tím, že $L < p$. Dále je předpokládáno, že soubor dat je uspořádán v n datových vektorech x_1 až x_n, přičemž každé x_i představuje jedno pozorování. x_1 až x_n jsou řádkové vektory, které jsou zapsány do matice X, která má výsledné rozměry $n \times p$.

Následně je zjištěn empirický průměr u podél každé dimenze. Výsledný vektor u je pak o rozměrech $p \times 1$.

$$u[j] = \frac{1}{n}\sum_{i=1}^{n} X[i,j]. \tag{2}$$

Nyní je nutné odečtení střední hodnoty, pro nalezení hlavních komponent. Je postupováno tak, že se data vycentrují následujícím způsobem:

1. Odečte se empirický průměr u v každém řádku matice X.
2. Odečtená vycentrovaná data se uloží do matice B o rozměrech $n \times p$.

$$B = X - hu^T, \tag{3}$$

kde h je sloupcový vektor s rozměry $n \times 1$. Všechny jeho hodnoty jsou 1.

V předposledním kroku je nalezena empirická kovarianční matice C s rozměry $p \times p$.

$$C = \frac{1}{n-1} B^* \cdot B, \tag{4}$$

kde * je operátor konjugované transpozice. B skládá výhradně z reálných čísel, což je případ mnoha aplikací. Konjugovaná transpozice je stejná jako běžná transpozice.

Na závěr jsou vypočteny vlastní vektory a vlastní hodnoty kovarianční matice, ty se pak nacházejí v matici vlastních vektorů V.

$$V^{-1}CV = D, \tag{5}$$

kde D je diagonální matice vlastních čísel matice C. Matice V, rovněž o rozměrech $p \times p$, obsahuje p sloupcových vektorů, každý o délce p, které představují p vlastních vektorů kovarianční matice C. Vlastní čísla a vlastní vektory jsou uspořádány a spárovány. *j-té* vlastní hodnotě odpovídá *j-tý* vlastní vektor. Celé odvození je dostupné ze zdroje [17].

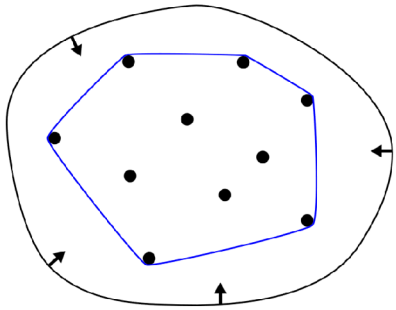
3.5 Ohraničení nejmenším obdélníkem

Dalším velmi důležitým algoritmem je nalezení plochy objektu a její ohraničení nejmenším možným obdélníkem [18; 19]. Z tohoto důvodu se bere v úvahu i natočení. Na následujícím obr. 10 jsou zobrazeny, mimo lyžaře, dva obdélníky zelený je normální ohraničující obdélník, zatímco červený je obdélník s minimální plochou.

Jak tato funkce pracuje, která dostane sadu bodů a k nim přiřadí nejmenší možný ohraničující obdélník. Nejprve se ze všech bodů vytvoří nejmenší konvexní obálku. Výsledným tvarem je polygon zahrnující všechny zadané body [19; 20]. Konvexní obálku lze definovat buď jako průnik všech konvexních množin obsahujících danou podmnožinu euklidovského prostoru, nebo ekvivalentně jako množinu všech konvexních kombinací bodů v této podmnožině. Konvexní polygon si lze představit na základě obr. 11 jako body, které jsou ohraničeny elastickou gumičku [20].

Obr. 10: Lyžař ve dvou rámečcích, červený box má minimální plochu [21]

Získanou obálku je nutné proměnit na nejmenší obdélník. Pokud je počet bodů v obálce větší než dva, je nutné využít Shamosův algoritmus. Technika, také nazývána jako *rotating calipers algorithm*, může být použita k řešení optimalizačních výzev [22]. Název metody je odvozen od myšlenky algoritmu, kdy se okolo vnější plochy konvexního tělesa otáčí pružinové posuvné měřítko. Pokud jedna čelist měřítka leží naplocho na hraně mnohoúhelníku, vytvoří protilehlou dvojici s bodem nebo hranou, která se dotýká opačné čelisti. Úplným otočením okolo polygonu jsou zajištěny všechny protilehlé dvojce, na než může být pohlíženo jako na graf. Algoritmus zajistí nejmenší obdélník, který může být natočený, ale stále obsahuje všechny zadané body.



Obr. 11: Konvexní obálka okolo bodů, které objímá elastická gumička

Předposledním možným stavem je, že počet bodů v polygonu je roven dvěma. Pak má obdélník šířku a délku, ale není nijak natočený [23]. Pokud je bodů méně jak dva, je obdélník definován pouze středem a nemá žádné rozměry, žádnou rotaci.

4 ALGORITMY VYUŽÍVAJÍCÍ NEURONOVÉ SÍTĚ

V předchozí části práce byly zmíněny klasické algoritmy, které lze použít ke zpracování obrazu. Nicméně jsou to jen dílčí algoritmy, které zapadnou do skládačky později. Nyní je nezbytné se zaměřit na přístupy jako je strojové učení, umělá inteligence nebo hluboké učení. Původní algoritmy strojového učení manuálně extrahovaly rysy, které byly následně klasifikovány učícím algoritmem. Zatímco neuronové sítě využívající hluboké učení fungují jako extraktor rysů a také jako klasifikátor [8]. Neuronová síť automaticky rozpoznává vzory a extrahuje rysy z obrázku. Navrhnout vlastní architekturu hluboké neuronové sítě není úplně snadné, protože to vyžaduje znalost složitých principů a mnoho experimentování. Existuje mnoho hotových modelů, které byly pečlivě vyvinuty a testovány na rozsáhlých datasetech, čímž dosahují vynikajících výsledků. Tyto předem vytvořené konvoluční neuronové sítě mohou být relativně snadno upraveny a použity pro nové úkoly, což se nazývá *transfer learning* [8]. Tento proces umožňuje efektivní využití znalostí a zkušeností nashromážděných při tréninku předchozích modelů. Pro úplné porozumění tématu se předpokládá základní povědomí o fungování neuronových sítí, což umožní zabrousit hlouběji do problematiky.

4.1 Konvoluční neuronové sítě

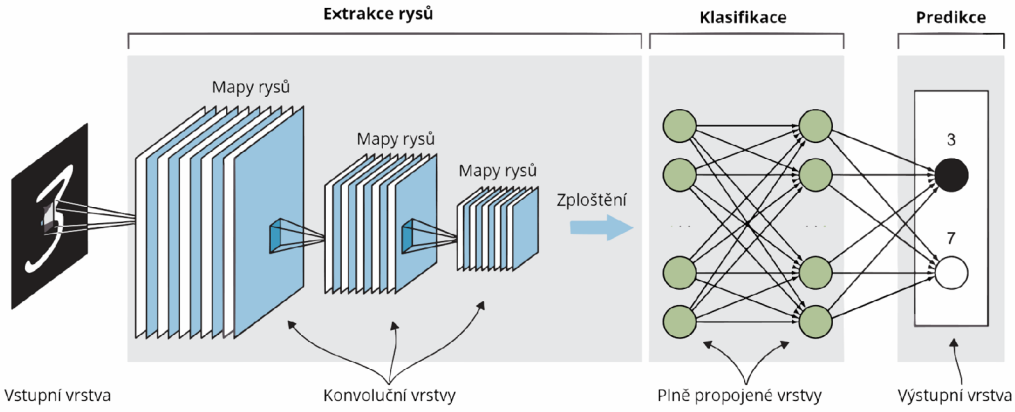
Běžné neuronové sítě obsahují více vrstev, stejně jako konvoluční neuronové sítě (CNN). První vrstva konvolucí se naučí některé základní rysy, kterými mohou být hrany, čáry [8]. Další vrstva se naučí další rysy složené ze základních. Naučené prvky jsou tedy o něco složitější. Mohou jimi být kruhy, čtverce, polygony a podobné útvary. Následující vrstva najde ještě složitější rysy (kola, psí vousy, cepín) a tak lze pokračovat dále. Je patrné, že architektura konvoluční neuronové sítě se řídí stejným schématem jako neuronová síť. Neurony ve skrytých vrstvách jsou vrstveny nad sebe, váhy jsou zpočátku náhodně inicializovány a učí se během trénování sítě, jsou aplikovány aktivační funkce, počítá se chyba *error loss*, a pomocí algoritmu zpětného šíření chyby *backpropagation* jsou aktualizovány nevyhovující váhy [8].

Před rozšířením hlubokého učení bylo nutné z obrázků manuálně extrahovat příznaky. Ty pak byly předány původním algoritmům strojového učení jako je *Support vector machine* (SVM) [8]. Díky konvolučním vrstvám lze nahradit manuální práci neuronovou sítí. Takže výsledná síť provádí jak určení příznaků, tak klasifikaci.

CNN se skládá ze čtyř typů vrstev, jak ilustruje obr. 12. V první řadě jde o vrstvy konvoluční, která jsou následovány vrstvami sdružovacími a plně propojenými. Základní funkce nejjednoduššího modelu konvoluční neuronové sítě lze rozdělit do čtyřech klíčových oblastí [8]:

1. Stejně jako v případě neuronové sítě obsahuje vstupní vrstva hodnoty pixelů obrázku.

2. Konvoluční vrstva určuje výstup neuronů, které jsou napojeny na lokální oblasti vstupu. Díky sdílení vah mezi různými částmi vstupních dat konvoluční vrstva snižuje počet parametrů a zvyšuje efektivitu tréninku.
3. Spojovací vrstva pak provede převzorkování *downsampling* podél prostorového rozměru vstupu. Dojde ke snížení parametrů v rámci aktivace.
4. Plně propojené vrstvy pak splní stejné úlohy jako v případě standardních neuronových sítí. Z aktivací se pokusí vytvořit skóre, které použije pro klasifikaci.



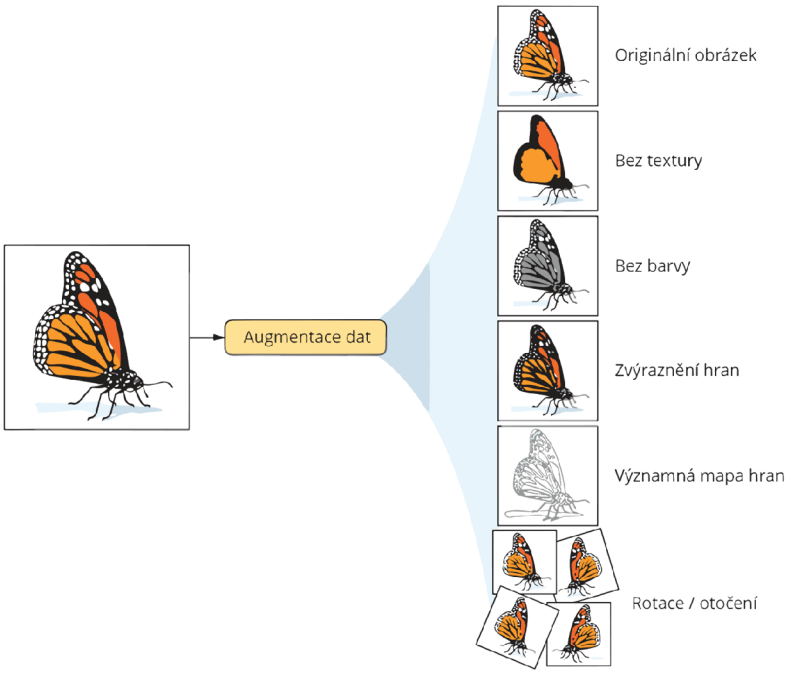
Obr. 12: Architektura konvoluční neuronové sítě rozdělená podle vrstev [8]

Tato jednoduchá transformační metoda umožňuje konvolučním neuronovým sítím postupně transformovat původní vstupní data pomocí konvolučních operací a techniky zmenšování vzorkování, které jsou aplikovány vrstva po vrstvě [24]. Následně vytváří skóre tříd pro účely klasifikace a regrese.

Byly představeny čtyři základní vrstvy, které se nacházejí téměř v každé konvoluční neuronové síti. Pro architekturu je typické, že se v ní mohou objevit i další vrstvy, které pomáhají předejít přeučení *overfitting* [8]. Špatná výkonost algoritmu strojového učení je nejčastěji spojena s problémem, kdy jsou data špatně interpretována. Geometrický útvar, kterým jsou reprezentována, neodpovídá jejich povaze. Cílem je najít vhodnou architekturu tak, aby nejvíc odpovídala zadání. Do modelu jsou zavedeny vylučovací *dropout* vrstvy [8]. Tyto sítě jsou schopny implementovat tzv. *dropout*, což je technika, která umožňuje náhodně deaktivovat určitý podíl neuronů v každé vrstvě během trénování. Tento podíl je specifikován jako hyperparametr při inicializaci sítě. Přístup může být považován za neintuitivní, ale postupným trénováním mohou některé neurony převažovat nad ostatními, nebo dokonce způsobovat velké chyby. Jejich vyřazení tak umožnuje vyvážit síť. Tedy pokud některý z neuronu udělá chybu, nepřeváží jeho výstup nad ostatními. V důsledku toho dochází ke kooperativní práci neuronů jako týmu, což znamená, že žádný jednotlivý uzel není příliš dominantní nebo příliš slabý, což přispívá k vyváženosti a efektivitě celého systému.

4.2 Augmentace dat

Je jasné, že konvoluční neuronové sítě dosahují velmi pozoruhodných výsledků v mnoha úlohách počítačového vidění. Ke svému správnému fungování ale potřebují velké množství surových dat, na kterých se učí [25]. Pokud je síti dodána velká množina učících dat, jde velmi dobře předejít přeučení *overfittingu*. Nicméně mnoho aplikačních domén nemá přístup k tolika datům. Lze to uvést na příkladu, kdy je cílem úlohy detekovat zmetek na výrobní lince. Aby mohl být zmetek odhalen pomocí metod strojového vidění, je nutné mít dataset obsahující vyšší stovky až tisíce obrázků. Pokud produkt není levný nebo se zmetek vyrobí zřídka, pak není jednoduché takovou datovou množinu získat. Augmentace dat je soubor technik, které zvětšují velikost a kvalitu trénovacích datových sad tak, aby z nich bylo možné sestavit lepší modely hlubokého učení [25]. Algoritmy pro rozšíření dat zahrnují geometrické transformace, barevné změny, konvoluční filtry, míchání obrazu, náhodné mazání obrazu. Technik je nepřeberné množství a opět záleží na konkrétní aplikaci, které je vhodné použít. Nejznámější příklady lze nalézt na obr. 13.



Obr. 13: Techniky augmentace obrazu vytváří nová vstupní data pro konvoluční síť [8]

Než bude uvedená samotná augmentace obrazu, je nutné opět připomenou, proč je rozpoznávání obrazu vůbec tak obtížným úkolem. Výzvami, které musí software překonat, jsou problémy s úhlem pohledu, osvětlením, pozadím, měřítkem a další [24]. Úkolem rozšíření dat je zpracovat zmíněné techniky, aby výsledné modely i přes tyto výzvy fungovaly obstojně.

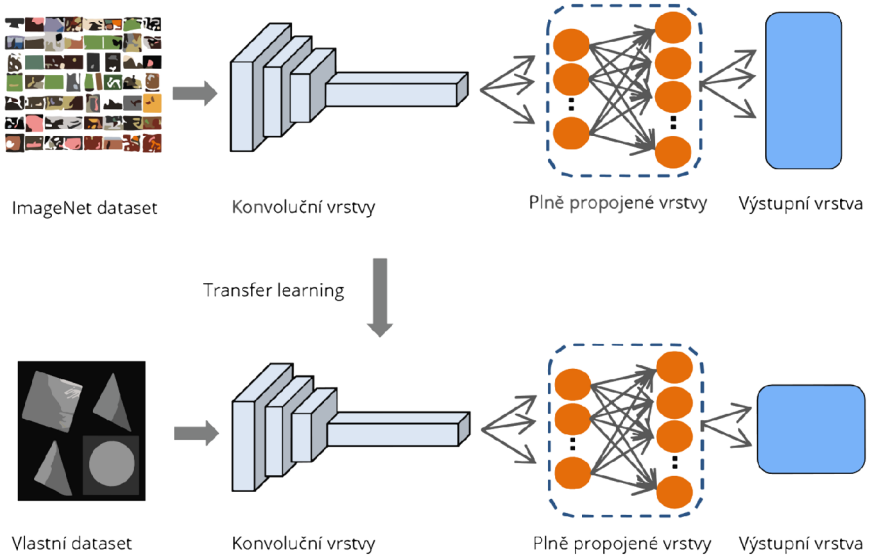
Nejsnáze pochopitelnou technikou pro rozšíření dat je skupina geometrických transformací. Třídu těchto postupů lze charakteristikovat snadnou implementací a jejich

pochopení přispívá k porozumění složitějším technikám. Otáčení obrázků je velmi rozšířeným přístupem. Lze jej rotovat vertikálně a axiálně. Za krátkou zmínku stojí bezpečnost techniky [24]. Znamená to, že zcela bezpečně může být překlopen obrázek horolezce nebo auta. Nicméně obrázek číslic šest a devět otáčet bezpečně nelze. Další geometrickou transformací je rotace, která způsobí, že vstupní obrázek může být rotován mezi 1° až 359°. Bezpečnost je zaručena, pokud je zvolen úhel od -20° do 20°. Jak bylo zmíněno, existuje velké množství technik, kterými lze dataset zvětšit. Model trénovaný na rozšířeném datasetu vykazuje lepší výsledky než původní.

4.3 Transfer learning

Pojem *transfer learning* by mohl být přeložen do češtiny jako přenesené učení. Nicméně český výraz není používán, a proto bude v textu užito anglické sousloví. Jde o jednu z nejdůležitějších technik hlubokého učení. Jak bylo uvedeno v oddíle věnujícím se augmentaci dat, při vytváření systému strojového vidění je obvykle nutné shromáždit velkou datovou množinu. Pro neuronovou síť jsou potřeba vyšší stovky až tisíce dat.

Lze tedy sestavit komplexní síť od samého počátku. Proces je velmi časově náročný a vyžaduje poměrně vysokou odbornost [8]. Existuje snazší přístup, kdy může být již hotová síť stažena, a použita jako odrazový můstek pro stávající úlohu. Hotová síť může být stažena již natrénovaná, tedy její váhy jsou smysluplně nastaveny a nejsou inicializovány náhodně. Výhodou je, že trénování vlastního modelu může být provedeno na menší datové sadě, trénování probíhá rychleji a lze dosáhnou kvalitnějších výsledků.



Obr. 14: Transfer learning se zachovanými konvolučními vrstvami a novým klasifikátorem [26]

Myšlenka je velmi jednoduchá. V prvním kroku je vycvičena hluboká konvoluční neuronová síť na velkém datasetu. Nejznámějšími velkými datasety jsou například

ImageNet, *MS COCO* a *Open Images* [8]. Během trénování síť extrahuje velké množství užitečných rysů, které lze použít k detekci objektů ve vlastním datasetu. Následně jsou rysy přenesené do nové neuronové sítě, která staví na síti původní. Ta je trénovaná na unikátní datové množině. Je předpokládáno, že model má být natrénován na rozpoznávání tvarů kruh, čtverec a trojúhelník. V specifickém zadání jsou pouze tři třídy. Nyní je nutné najít datovou sadu, která bude podobně odpovídat zadaným třídám. Jelikož se jedná pouze o tvary, tudíž jde především o správnou detekci hran, může být použit jakýkoliv velký dataset. Dále je nutné vybrat správnou architekturu konvoluční sítě jako je *VGGNet*, *GoogLeNet* nebo *ResNet* [8]. Aby mohl být algoritmus aplikován na vlastní dataset, je nutné odstranit původní klasifikátor a nahradit jej nově vytvořeným. Celý proces trénování na vlastním datasetu je vyobrazen v obr. 14.

Větším podrobnostem není nezbytné se v této práci věnovat. Nutné je shrnout signifikantní výhody *transfer learningu*. Používá se zejména v případech, kdy není k dispozici velký dataset, tudíž ve většině prakticích případů [8]. Lze jej použít třemi různými způsoby: využití předtrénované sítě jako klasifikátoru, jako extraktoru příznaků nebo pro jemné doladění vah *fine-tuning*. Přenositelnost extrahovaných rysů z jedné sítě do druhé je zaručeno velikostí zdrojových dat.

4.4 Frameworky pro detekci objektů

Doposud zmíněná fakta se týkají pouze klasifikace objektu. Jde tedy o jeden objekt zachycený na obrázku. Častějším případem je, že ve vstupních datech jsou zachyceny různé objekty. Cílem úlohy je objekty z obrázku klasifikovat a správně určit jejich pozici v obrázku. Úloha, s cílem klasifikace objektu a určením jeho pozice, se nazývá detekce objektů [8]. Výstupem je obrázek, ve kterém jsou nakresleny obdélníky ohraničující objekty. Každý obdélník obklopuje jeden objekt a je doplněný štítkem s názvem a skóre ohodnocení. Vykreslený obdélník se nazývá *bounding box*. Jinými slovy lze konstatovat, že systém nepředpovídá pouze třídu, jak je tomu v úlohách klasifikace obrazu, ale předpovídá také souřadnice *bounding boxu*, který odpovídá zjištěnému objektu. Je zcela patrné, že se jedná o velmi složitou úlohu strojového vidění, protože je nutná správná klasifikace i lokace. Detekce objektů je velmi rozšířenou technikou v průmyslu, autonomním řízení automobilů, lékařství, potravinářství a mnoha dalších oborech.

Nyní budou představeny nejoblíbenější algoritmy určené pro detekci objektů. Těmito přístupy jsou *R-CNN*, *SSD*, *YOLO* [8]. Budou probrány z hlediska architektury a způsobu fungování, nechybí ani jejich porovnání stěžejní pro tuto práci. Pro snazší pochopení funkcionality je nutné osvojení následujících pojmů [8]:

Návrh oblasti

Konvoluční neuronová síť se používá k vytvoření oblastí zájmů, které dále zpracovává. Systém se domnívá, že ve vybraných oblastech se vyskytují objekty. Výstupem je tedy obrázek s mnoha *bounding boxy* a jejich skóre. Nejlepší z nich jsou předány do dalších vrstev sítě [8].

Extrakce rysů a předpověď sítě

Pro každý rámeček jsou extrahovány jednotlivé rysy. Vytažené rysy se vyhodnotí a na základě vizuálních rysů se určí, zda se objekty v rámečcích nacházejí. Je vyhodnoceno jestli vůbec a jaký objekt v oblasti je [8].

Non-maximum suppression (NMS)

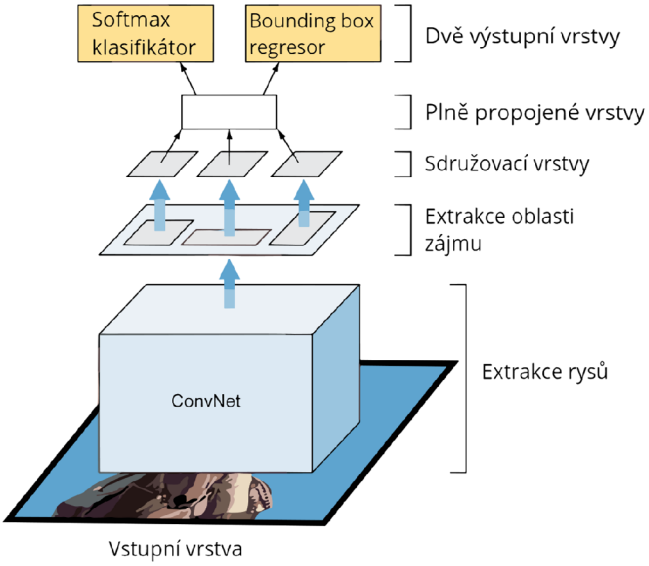
Většinou nastane případ, že model nalezne více *bounding boxů* pro stejný objekt. NMS je systém pro zabránění násobným detekcím téhož. Dosáhne se toho způsobem, že kombinuje nalezené rámečky do jednoho výsledného [8].

Metriky hodnocení

Jde o podobné techniky jako v případě klasifikace pro vyhodnocení výkonosti sítě. Velmi oblíbeným ukazatelem je průměrná přesnost *mean average precision* (mAP) [8]. Další důležitou metrikou pro měření rychlosti detekce je počet snímků za sekundu (FPS), který udává, kolik snímků za sekundu je model schopen zpracovat. Tyto metriky jsou klíčové pro posouzení výkonu detekčního modelu a jeho praktické použitelnosti v reálném čase.

4.5 Rodina sítí R-CNN

Již v roce 2014 byl publikován článek v oblasti strojového vidění týkající se frameworku R-CNN (*Region-based Convolutional Neural Network*) [27]. Jde o typ architektury používaný pro detekci objektů v úlohách strojového vidění. R-CNN připravil půdu pro mnohé inovace v oblasti detekce objektů. Je prvním publikovaným algoritmem z rodiny R-CNN, na nějž navázaly stejně úspěšné *Fast* R-CNN, *Faster* R-CNN (obr. 15) a *Mask* R-CNN [8]. Nicméně pro pochopení novějších variant je nezbytné vytvořit silné základy znalostí původní R-CNN.



Obr. 15: Architektura vyspělejší sítě Fast R-CNN

Návrh oblasti

V prvním kroku je vstupní obraz rozdělen na mnoho oblastí a podoblastí [27]. Proběhne vygenerování množiny potenciálních regionů v obraze. Vytváření navrhovaných oblastí není přímo implementováno v architektuře R-CNN, ale je využito externích metod, jako je selektivní vyhledávání. Zmíněná externí metoda funguje tak, že spojuje nebo rozděluje segmenty obrazu na základě různých obrazových podnětů, jako je barva, textura, tvar. Výsledkem je vytvoření rozmanité sady návrhů oblastí.

Extrakce rysů a předpověď sítě

Po získání zón zájmu, kterých mohou být až jednotky tisíců, je nutné připravit data jako konzistentní vstup pro konvoluční síť [27]. Vstupní oblasti jsou deformovány na stejnou velikost (224 x 224 pixelů), kterou konvoluční síť očekává. Dále proběhne známé získání rysů. Použitou sítí je velmi výkonná *ConvNet*, která využívá váhy získané z velkého datasetu [8]. Výstupem CNN je vícerozměrný vektor příznaků reprezentující obsah návrhů oblastí.

Získaný vektor je vložen do klasifikátoru strojového učení. R-CNN obvykle využívá ke klasifikaci metodu *Support Vector Machine* (SVM) [8]. Pro každou třídu je natrénován jedinečný SVM, který určí, zda oblast obsahuje instanci požadované třídy, či nikoliv. Při trénování jsou vzorky rozděleny na pozitivní a negativní s tím, že negativní třídu neobsahují. Mimo klasifikace je také prováděna regrese *bounding boxů* [27]. Pro každou třídu je natrénován samotný regresní model, který upřesňuje umístění a velikost boxu tak, aby nejlépe odpovídal hranicím skutečného objektu.

Non-maximum suppression

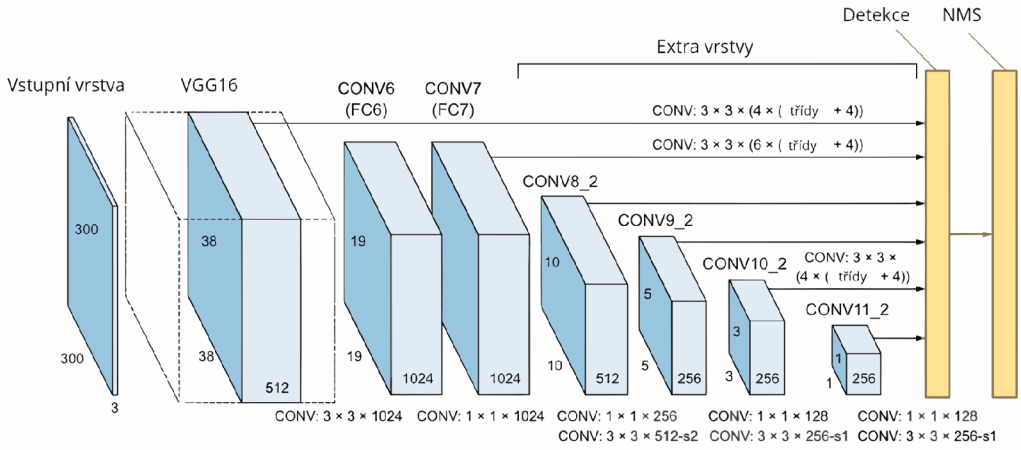
Po určení tříd je použita metoda *Non-maximum suppression.* Velice efektivně se odstraní duplicitní nebo překrývající se ohraničující rámečky. NMS zajistí, že ve výstupu jsou zanechány pouze nejjistější a nepřekrývající se rámečky. [8; 27]

4.6 Single Shot Detector

V roce 2016 byla představena revoluční metoda detekce objektů v oblasti strojového vidění nazvaná *Single Shot Detector* (SSD) [8]. Tento přístup v sobě kombinuje vysokou rychlost a přesnost detekce, což ho činí jedním z nejvýznamnějších algoritmů v této oblasti. SSD eliminuje potřebu složitých postupů, jako je generování regionů zájmu, a umožňuje přímou detekci objektů na vstupním obraze.

Návrh oblasti

SSD funguje na základě jediného konvolučního průchodu přes celý obrázek, přičemž zároveň produkuje rozmanitou sadu detekčních boxů různých velikostí a poměrů stran v různých částech obrazu [8]. Tento přístup umožňuje zachytit objekty různých velikostí a umístění v jednom kroku. Díky tomu je SSD mnohem efektivnější a rychlejší než předchozí metoda, která vyžadovala více průchodů obrázkem. Vstupní obrázek je požadován v rozměrech 300 x 300 pixelů. Architektura sítě je znázorněna na obr. 16.



Obr. 16: Architektura frameworku Single Shot Detector [8]

Extrakce rysů a předpověď sítě

Používanou konvoluční sítí je *VGG16* [8]. Po extrakci rysů z obrázku pomocí konvoluční sítě jsou předpovědi tříd a bounding boxů prováděny přímo z rysů získaných ze vstupního obrázku. To znamená, že SSD nepotřebuje žádné další metody pro generování oblastí zájmu nebo externí klasifikátory. Tento integrální přístup zajišťuje rychlou a přesnou detekci objektů bez zbytečných kroků.

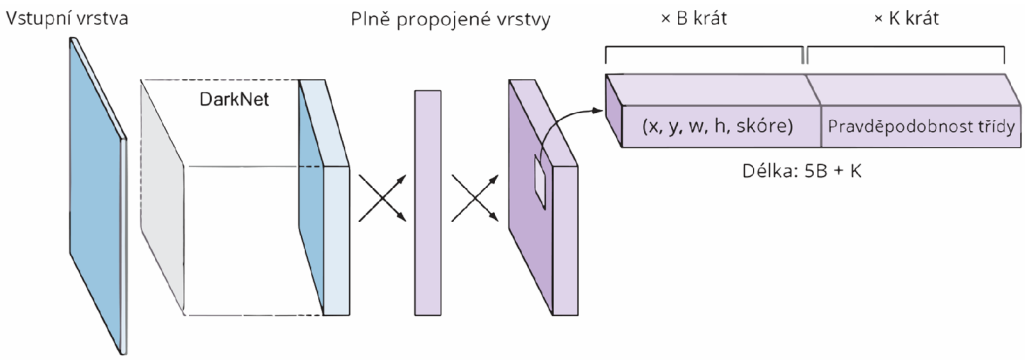
Non-maximum suppression

Stejně jako u předchozí metody, i u SSD je využita technika *Non-maximum suppression* k odstranění duplicitních detekcí a zajištění pouze jednoho detekčního boxu pro každý objekt [8]. To pomáhá udržet výstup algoritmu přehledný a eliminuje redundanci v detekovaných objektech. NMS je klíčovou součástí procesu detekce objektů v SSD a přispívá k jeho vysoké přesnosti.

4.7 Rodina sítí You Look Only Ones

Již v roce 2015 byl představen revoluční framework nesoucí název *You Look Only Ones* (YOLO) [8]. Tento přístup v sobě kombinuje vysokou rychlost a přesnost detekce, což ho činí jedním z nejvýznamnějších algoritmů v oblasti detekce obrazu. Ze zmíněných metod je více podobný rodině R-CNN než SSD a opět se jeho prvotní vydání dočkalo mnoha dalších verzí. Mezi ně patří populární YOLOv3, YOLOv5, YOLOv8 a nejnovější YOLOv9. Celá rodina YOLO je řazená k *end-to-end* modelům, jejíž velkou předností je možnost detekce objektů v reálném čase [8].

Tvůrci frameworku zvolili jiný přístup než předchozí uvedené modely. YOLO neprochází klasickým krokem navržení oblasti. Místo toho pouze provádí předpověď nad omezeným počtem *bounding boxů* [8]. Vstupní obrázek je rozdělen mřížkou na velké množství buněk a každá z nich přímo předpovídá klasifikaci objektu. Výsledkem je velký počet boxů, které jsou následně upraveny pomocí NMS. Modely YOLO lze trénovat na jediném GPU, což jej zpřístupňuje velkému množství vývojářů.



Obr. 17: Architektura sítí YOLO [8]

4.7.1 YOLOv8

Verze frameworku v8 je nejnovější do ledna roku 2024.[2] Lze jej použít pro detekci objektů, klasifikaci obrazu, segmentaci prvků obrazu. Současná podoba je vyvinuta a udržována společností *Ultralytics,* která vyvinula i předchozí velmi úspěšnou verzi v5 [28]. Lze konstatovat, že novou verzi lze považovat za vylepšení původního vydání v5. Nezbytné je taktéž zmínit, že vývoj nových funkcionalit probíhá velmi dynamicky a z toho důvodu může část této práce v krátkém horizontu ztratit na aktuálnosti.

S rychlým vývojem přichází i nemalé množství kuriozit. Jednou je, že YOLOv8 nemá stále zveřejněný vědecký článek, který by více osvětlil vhled do metodiky výzkumu a studií využitých při vytváření [28]. Ze zdrojového kódu je patrno mnoho dílčích vylepšení: volná detekce, nové konvoluce, uzavření *mosaic* augmentace.

4.8 Srovnání jednotlivých modelů

Při porovnáváním modelů se naráží na úskalí způsobené skutečností, že každý framework je vhodný pro jinou aplikaci. Jiný model bude dobrý pro hledání malých detailů ve složitých strukturách, jiný pro realtimové úlohy, jiný bude jednoduchý na implementaci. Je zřejmé, že aspektů pro posouzení je mnoho a finální rozhodnutí je vždy na technikovi, jaký přístup a jakou architekturu zvolí. Nyní budou uvedeny řešené příklady, kdy stěžejním úkolem bylo porovnání frameworků z rodiny R-CNN, SSD a rodiny YOLO.

V prvním případě je nutné klasifikovat medikamenty typu perorálních tablet. V Číně nejsou léky prodávány v obalech, jako je to v evropských zemích, ale jsou v lékárnách rozdělovány zaměstnanci [29]. Mnoho tablet je vizuálně podobných a lékárníkům může činit problém je od sebe po odstranění obalu rozeznat. Bylo by zcela

[2] V únoru 2024 byla uvedena na trh nejnovější a dosud nejvýkonnější verze detekčního systému YOLOv9, která prokázala výrazné zlepšení výkonu na vybraných benchmarkových datech ve srovnání s předchozími verzemi. Její implementace je dostupná také pod společností *Ultralytics.* Ta poskytuje velmi širokou podporu uživatelům na platformě *Discord*.

nepřirozené ke klasifikaci tablet nepoužít nějaký z algoritmů strojového vidění. V případě léčebných tablet jsou použity sítě YOLOv3, SSD a *RetinaNet*[3].

Sítě byly dotrénovány metodou transfer learningu s použitím inicializačních vah z datasetu *MS COCO* [29]. Výsledky, kterých bylo dotaženo trénováním, jsou uvedeny v následující tab. 1.

Tab. 1: Výsledky frameworků po trénování na datasetu medikamentů [29]

Algoritmus	Přesnost [%]	Rychlost [FPS]	mAP [%]
RetinaNet	64,98	17	82,89
SSD	63,69	32	82,71
YOLOv3	69,65	51	80,69

V závěru je konstatováno, že SSD má vysokou hodnotu mAP (82,71 %), ale rychlost detekce je pomalejší než YOLOv3 [29]. SSD je výkonnostně na střední úrovni mezi dalšími dvěma sítěmi. Dále je uvedeno, že nejlepším kompromisem je framework YOLOv3. Dokáže velmi rychle identifikovat léky, snížit pravděpodobnost vydání špatného léku a také může být implementován na platformu s nízkým výkonem.

Ve druhém případě bylo strojové vidění použito ke zkoumání skleníků s využitím snímků dálkového průzkumu země a bezpilotních letadel. Skleníky se velmi rychle rozšiřují v Číně, Evropě, severní Africe a na Blízkém Východě [30]. Rychlá výstavba však vyvolala mnoho problémů v hospodařením s půdou, jako je zabírání kvalitní orné půdy, poškození půdy a znečišťování plastovými odpady. Aplikace strojového vidění by mohla usnadnit vyhodnocování snímků. Zkoumány z hlediska své efektivity byly algoritmy *Faster* R-CNN, YOLOv3 a SSD.

Všechny sítě byly implementovány pomocí knihovny *PyTorch*, za jejímž vznikem stojí vývojový tým Facebooku [30]. Opět byla využita metoda transfer learning, nicméně není uvedeno z jakého datasetu byly použity výchozí váhy. Výsledky natrénovaných sítí ukazuje tab. 2.

Tab. 2: Výsledky frameworků pro detekci skleníku ze satelitních snímků [30]

Algoritmus	Přesnost [%]	Rychlost [FPS]	mAP [%]
Faster R-CNN	91,00	12	86,00
SSD	85,40	35	84,90
YOLOv3	87,50	73	90,40

[3] *RetinaNet* je algoritmus, který nebyl při řešení práce využit, a proto není blíže představen.

Z výsledků experimentu vyplývá, že přesnou a rychlou detekci nejlépe provede YOLOv3 [30]. Framework dosáhl střední přesnosti (87,50 %), nevyšší rychlosti (73 FPS) a nejlepší metriky mAP (90,40 %). Lze je tedy využít pro přesnou a rychlou detekci skleníků z družicových snímků.

5 ROBOTICKÁ BUŇKA EDUSET ONE

Následující kapitola je první praktickou částí práce. Dosud byly podrobně probrány teoretické aspekty práce, které se v dalších částech propojí v praktickou implementaci strojového vidění. Hlavní motivací diplomové práce je zavedení aplikace strojového vidění do robotické buňky *EDUset ONE*. Má být navrženo několik řešení v souladu s trendy Průmyslu 4.0. Ve zmíněném kroku bude nutné navrhnou umístění kamery nebo senzoru do buňky. V dalším kroku budou implementovány různé frameworky pro detekci objektů z plochy, z nichž bude vybrán ten nejvhodnější. Úspěšný framework bude použit v softwaru pro detekci semi-strukturovaně rozmístěných polotovarů z plochy pomocí metod umělé inteligence. V závěrečném kroku bude ověřena funkčnost aplikace s reálným hardwarem nacházejícím se v buňce.

5.1 EDUset ONE

Buňka *EDUset ONE* je multifunkční robotické pracoviště vytvořené za účelem prezentace možnosti rychlé rekonfigurace automatizovaného pracoviště. Toto pracoviště obsahuje 5osou frézku *slvEDU* od společnosti *SolidVision*, kolaborativního robota *ur10e*, gripper a kameru umístěnou na přírubě robota, oboje od společnosti *RobotiQ*. Díky těmto komponentám je možné provádět různé výrobní operace s vysokou mírou flexibility a přesnosti. Navíc, *EDUset ONE* má své vlastní digitální dvojče vytvořené v softwaru od společnosti *Siemens*. Toto digitální dvojče umožňuje testování nových výrobních cyklů, optimalizaci stávajících cyklů a testování nových či přídavných hardwarových komponent bez nutnosti fyzického vlastnictví těchto komponent, což značně zefektivňuje proces vývoje a implementace nových technologií.

Jednou z hlavních myšlenek projektu *EDUset ONE* je poskytnout studentům příležitost podílet se na jeho vývoji a využití. Studenti mohou být zapojeni do vývoje a testování nových technologií zejména formou bakalářských a diplomových prací, čímž získají cenné praktické zkušenosti. V budoucnu je plánováno využívat *EDUset ONE* také jako výukový nástroj, který umožní studentům vysokých i středních škol přijít do kontaktu s moderními automatizačními technologiemi. Tím se nejen zvýší úroveň výuky, ale také se doplní chybějící praktické dovednosti, což povede k lepší připravenosti absolventů na pracovní trh. Tento přístup může výrazně přispět k rozvoji technických dovedností a inovativního myšlení u studentů.

5.2 Volba senzoru pro sběr dat

Výzvou je volba vhodného snímače dat, kterým v aplikaci strojového vidění bývá kamera. V průmyslové praxi jsou nejčastěji aplikovány dva typy kamer. Prvním typem je samotná kamera, která je bez objektivu a přídavného osvětlení. V podstatě jde o snímač integrovaný do hliníkového těla, ke kterému je možné se připojit přes komunikační rozhraní. Nejznámějším zástupcem firmy, specializující se na výrobu uvedených kamer, je firma *Basler AG*. Druhým typem senzoru je chytrá kamera. Jedná se o integraci všech klíčových komponent strojového vidění do jednoho celku. Chytrá kamera je kompletní řešení pro aplikaci strojového vidění. Je v ní integrován objektiv, světlo, komunikační rozhraní a software pro strojové vidění. Největší hráči v oblasti průmyslové automatizace si většinou vyrábí i vlastní kamery. Výrobci s portfoliem programovatelných relé[4] a chytrých kamer jsou například firmy: *B&R Industrial Automation*, *SICK*, *Keyence Corporation*, *Epson*.

Obr. 18: Porovnání kamery firmy *Basler AG* a chytré kamery *B&R Industrial Automation*

Oba dva typy kamer poskytují značné množství výhod při implementaci a jejich možnosti použití zcela odráží trendy Průmyslu 4.0. Pro aplikaci je nutné zvolit pouze jednu kameru. Výhody a nevýhody obou typů kamer jsou uspořádány v obr. 18 [31; 32]. Na základě charakteru práce byla zvolena cesta s využitím modulární kamery od firmy *Basler AG*. Důvodů pro volbu modulárního řešení je hned několik. V první řadě není připravena scéna, ve které budou polotovary snímány. Není možné tedy dopředu definovat objektiv kamery, velikost a barvu osvětlení. Scénu je nezbytné vytvořit – znamená to vyzkoušet různé druhy snímačů, objektivů, pozadí a světel. Pro řešení práce bylo nezbytné vytvořit vlastní software detekce semi-strukturovaně rozmístěných polotovarů a firma *Basler AG* ke svým kamerám poskytuje aplikační programové rozhraní (API). Komunikace a sběr dat se tak usnadňuje a je zprostředkován pomocí

[4] Programovatelné relé je poměrně zastaralý výraz. Nicméně nelze použít pojem programovatelný automat, jelikož všechny zmíněné firmy PLC nevyrábí. Ovšem vyrábí například programovatelná bezpečností relé (firma *SICK*) nebo velmi jednoduchá programovatelná relé (firma *Keyence*).

programovacího jazyka *Python*. Dále je nutné počítat se stanicí s patřičným výpočetním výkonem. Díky těmto přednostem lze vytvořit komplexní balíček s vlastnostmi chytré kamery.

5.3 Možnosti umístění kamery v buňce

V předchozí kapitole byly vyhodnoceny dva různé typy kamer pro sběr surových dat z robotické buňky. Byl zvolen senzor vyhovující pro shromažďování dat ze scény. Kameru je nyní nezbytné umístit do stávající buňky. Je připomenuto, že se jedná o polotovary rozmístěné na pracovním stole. Polotovary jsou odebírány manipulátorem, který je umísťuje do obráběcího centra. Ve frézce jsou polotovary upnuty a obrobeny na základě vyvolaného programu. Kamera musí být tedy umístěna tak, aby mohla snímat celou plochu stolu, kde jsou budoucí obrobky umístěné.

V praxi je možné nejčastěji narazit na následující realizace umístění kamery: nahoře nad pracovní plochou, na manipulátoru, na stěně, pod pracovní plochou, na mobilním robotu a víceúrovňově. Z charakteru úlohy lze zavrhnout umístění na stěně, jelikož se jedná o autonomní robotickou buňku a zmíněné umístění by samostatnost buňce odebralo. Umístění pod pracovní plochou je nerealizovatelné, protože stůl není průhledný. Víceúrovňová instalace také nehraje roli, protože je k dispozici pouze jedna kamera. Zavedení kamery na autonomního robota by mělo smysl, pokud by se v buňce alespoň jeden nacházel. Zůstávají pouze dvě aplikovatelná řešení: instalace kamery nad pracovní plochu a na manipulátor.

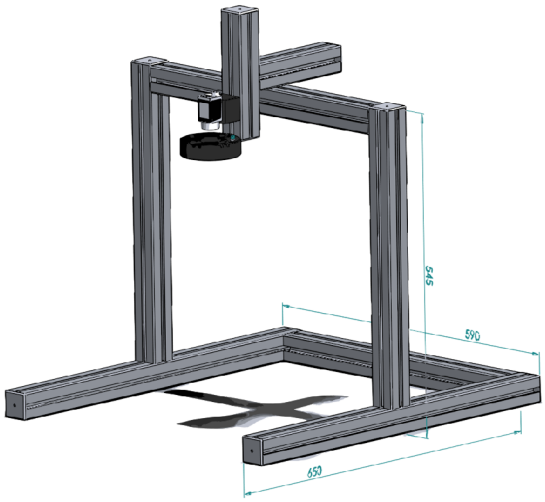
Pokud by se kamera nacházela na robotickém rameni, vyskytne se velká skupina možností sestavení scény. Robotické rameno může také splnit požadavek víceúrovnosti. Tedy kamera může dosáhnout různých vzdáleností od snímané plochy. Pracovní stůl s polotovary může být rozsáhlejší. Pokud by šlo o zásobník na polotovary, lze tímto přístupem zvýšit jeho kapacitu s využitím jedné kamery. Další výhodou je, že data mohou být snímána z různých úhlů. Pokud je polotovar detekován s nízkou pravděpodobností, může být kamera na manipulátoru natočena a objekt určen přesněji. Možnost pohybu pomůže s přesnějším určením pozice. Pokud je kamera osazena netelecentrickým objektivem dochází ke zkreslení předmětů vlivem soudkovitosti. Ta se projevuje více s větší vzdáleností od středu.

Umístění kamery na manipulátor nabízí více výhod než umístění nad pracovní plochu. Zmíněnými přednostmi umístění nad pracovní plochu nedisponuje. Není ale vyloučeno, že by kamera nemohla být přemístěna z jednoduché konstrukce na robotické rameno. Instalací kamery na jednoduchou konstrukci se ušetří práce s vyvazováním vodičů na manipulátor a veškeré testy a mohou probíhat mimo robotickou buňku. Také není nutné konstruovat speciální držák pro kameru a světlo. Následným přemístěním se dosáhne vyšší přesnosti a možnosti snímat větší plochu.

5.4 Zavedení nové konstrukce pro strojové vidění

Pro řešení práce je zvolena cesta, při které je vytvořena jednoduchá hliníková konstrukce. Důvodem je sestavení konstrukce mimo buňku, tedy vývoj může probíhat nezávisle na ní. Dalším praktickým důvodem je, že celá aplikace jde udělat v menším měřítku a následně převést do většího[5]. Tímto způsobem se za cenu malých nákladů otestuje celý proces.

V současné době konstruktéři automatizačních linek opouští od svařovaných příhradovin. Trend je takový, že místo svařenců se používají hliníkové profily spojené šrouby. Velkou výhodou je váha materiálu a snadnost realizace jednotlivých spojení. Svářeč musel být proškolený a značně zkušený, aby dokázal dodržet požadovanou přesnost. Nyní je proces značně jednodušší. Hliníkové profily si lze objednat od firem, které připraví již hotové přířezy. Společnost vyrábějící automatizační linky tedy nemusí vlastnit přesnou pilu a nemusí mít zaměstnaného zkušeného svářeče. Nařezané profily je pouze nutné závitovat a spojit šrouby v válcovou hlavou. Tímto způsobem vzroste samotná linka rychleji než dříve. Dalším kladným aspektem je značná modulárnost a přestavitelnost.



Obr. 19: Konstrukce pro aplikaci strojového vidění v buňce *EDUset ONE*

Společnostmi vyrábějící hliníkové profily a potřebný spojovací materiál jsou například: *Bosch Rexroth*, *MiniTech*, *Item Industrietechnik GmbH*. Výrobky nejsou navzájem kompatibilní. Profily pro konstrukci strojového vidění jsou typu *MiniTech*. Samotnou sestavu lze vytvořit v jakémkoliv 3D modeláři. Nicméně *MiniTech* poskytuje vlastního modeláře *iCAD Assembler*. Program je velmi intuitivní a sestavu lze exportovat ve formátu *stl*. Požadavky na konstrukci nejsou velké: musí být z minimálního množství materiálu, kamera musí být v dostatečné vzdálenosti od pracovní plochy, nesmí omezovat manipulační prostor robotického ramene. Z uvedených požadavků vznikla jednoduchá

[5] Podobný přístup je praktikován v podnikání, kdy je do praxe uvedeno MVP (*minimum valuable product*), který je na základě zpětné vazby zákazníku vylepšován.

sestava, jejíž model je na obr. 19. Konstrukce je poměrně robustní, nehrozí tedy převážení a následné poškození kamery. Také lze nastavit výšku kamery nad pracovní plochou. Prostor pro manipulaci robota je dostatečně velký. Na profily lze přidat zdroj 24 V napětí, který může napájet kameru a osvětlení.

5.5 Výběr vizuálního systému

V kapitole 5 byla představena robotická buňka *EDUset ONE*. Do buňky byla vytvořena sestava z hliníkových profilů. Taktéž byl zvolen typ kamery, kterým budou obrazová data sbírána. Výzva implementace strojového vidění neobsahuje ovšem jen tyto části. Kamera byla vybrána od firmy *Basler AG*, ale není zřejmé, který model přesně bude použit. Prvním krokem v této kapitole bude zvolit vhodnou kameru s objektivem. Jakmile bude kamera vybrána, není zaručen absolutní úspěch. Z praktických úloh vyplývá, že volba vhodného osvětlení a pozadí, jsou stejně důležité jako samotná volba kamery s objektivem. Bude představeno několik možných řešení, z nichž bude vybráno adekvátní zadané úloze. Nejde se bavit o nejlepším řešení, protože je potřeba vždy brát v úvahu dostupnost a cenu hardwaru pro aplikace strojového vidění. V praxi je častý případ, kdy je nutné pracovat nikoliv nejlepším hardwarem, ale dostupným.

5.6 Polotovary určené pro k detekci

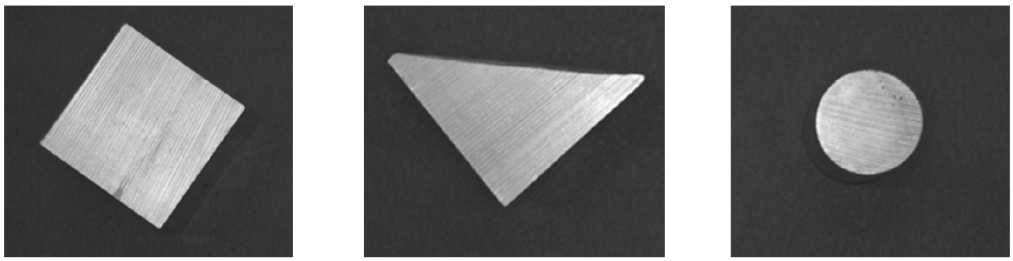
Je třeba představit polotovary, jež je nutné detekovat. Tedy určit o jaký polotovar se jedná a jaké je jeho umístění v obraze. Pokud jsou brány v úvahu polotovary, které se upínají do frézky, většinou se jde o modifikace kvádrů. Proces v velkosériové výrobě bývá takový, že obráběcí stroje jsou doplněny o velké zásobníky. Příkladem může být výroba pístnic do pneumatických pohonů. Tyče o délce asi 5 m jsou vloženy do zásobníku a obráběcí stroj si už sám polotovar posouvá a opracovává. Z velkosériového obráběcího stroje již tedy padají hotové produkty, bez větších zásahů operátora[6]. Ve zmíněném případě není aplikace strojového vidění nezbytná. Polotovar vstupující do stroje bude vždy stejný. Druhým případem je kusová výroba. Operátor obráběcího stroje má k dispozici polotovary různých velikostí a tvarů. Z nich je nezbytné zvolit vždy jeden vhodný pro konkrétní program. Ve většině případů stroj nevytvoří obrobek na jedno upnutí. Operátor musí změnit polohu obrobků několikrát.

V diplomové práci je uvažována právě kusová výroba. Z řídícího místa bude poslán pokyn, jaký má být program, tedy jaký je finální obrobek. Na základě toho se vyhodnotí vhodnost vstupního materiálu. Nebylo určeno, jaké polotovary budou ve frézce upínány. Vlastní volba byla založena na zvážení několika aspektů: vhodný materiál, geometrie a rozměry, povrchová úprava, barva, kompatibilita s osvětlením, přítomnost výrazných rysů. Po důkladném zvážení bylo rozhodnuto, že polotovary musí být ze stejného

[6] Operátor velkosériové výroby většinou jen mění program a nastavuje korekce na obráběcím stroji. Dokáže tak obsloužit více strojů najednou.

materiálu, jejich tvary musí být odlišné a povrch bude nést stopy po dělení pilou. Vhodné je použít tvary s výraznými rysy kruhu, čtverce a trojúhelníku. Vybrané polotovary jsou pohodlně upnutelné v obráběcím centru. Materiálem byl zvolen hliník, jelikož má dobré vizuální vlastnosti a obrábění je méně nákladné než oceli. Vstupní materiál úlohy je zachycen na obr. 20. Při výběru polotovarů je důležité provést pečlivou analýzu požadavků vaší konkrétní aplikace strojového vidění a vybrat polotovary, které nejlépe splňují tyto požadavky.

Vybírání polotovarů pro strojové vidění v kusové výrobě představuje klíčový krok v procesu. Zásadní je zohlednit požadované vlastnosti finálního produktu a zvolit polotovary, které umožní dosažení těchto parametrů efektivně a spolehlivě. V tomto případě bylo rozhodnuto použít polotovary ze stejného materiálu, ale s odlišnou geometrií a povrchem, což odpovídá požadavkům na variabilitu výsledných produktů. Důležité je také vzít v úvahu praktické aspekty, jako je pohodlné upínání polotovarů v obráběcím centru a finanční efektivita výběru materiálu. Hliník byl zvolen jako vhodný materiál pro své vlastnosti a ekonomickou výhodnost. Pečlivé zhodnocení těchto faktorů pomohlo zajistit, že vybrané polotovary budou optimálně odpovídat požadavkům aplikace strojového vidění a přispějí k úspěšné realizaci projektu.



Obr. 20: Zvolené tvary polotovarů pro aplikaci strojového vidění

5.7 Výběr kamery a objektivu

Pro zavedení aplikace strojového vidění do robotické buňky *EDUset ONE* je nezbytné vybrat vhodnou kameru a objektiv. K dispozici bylo poskytnuto několik kamer s příslušnými objekty. Úkolem nebylo vyzkoušení různých kombinací kamer a objektivů, ale z celkem tří sestav vybrat tu nejvhodnější. U kamer je kladen velký důraz na následující parametry: rozlišení, typ senzoru, rychlost snímání, uzávěrka a komunikační rozhraní. Zatímco o vhodnosti objektivu rozhodují následující parametry: ohnisková vzdálenost, clona, rozlišení a optický formát.

Srovnání vedoucí k výběru vhodné kamery je provedeno v rámci tab. 3. Parametry vedoucí ke srovnání kamer a objektivů jsou dostupné z: [32; 33; 34; 35; 36; 37]. V tabulce nejsou uvedeny všechny parametry, protože v mnohých se kamery shodují, není tedy nutné všechny uvádět. Zmíněny jsou pouze ty, které byly zásadní pro rozhodnutí. Všechny uvedené kamery používají technologii senzoru *CMOS* s rozlišením 5 MP.

Společným komunikačním rozhraním je *GigE Vision* a závit pro objektiv je *C*. Všechny objektivy mají pevné ohnisko a jejich závit je *C*.

Po důkladném srovnání a analýze parametrů kamer a objektivů je nezbytné provést detailní testování vybrané kombinace v reálném prostředí robotické buňky *EDUset ONE*. Testování bude zaměřeno na ověření funkcionality, výkonu a spolehlivosti vybraného vizuálního systému v různých pracovních podmínkách. Získané výsledky testování poskytnou důležité informace o tom, zda je vybraná kombinace kamer a objektivů schopna splnit požadavky projektu a efektivně podporovat účely aplikace strojového vidění v robotické buňce. Výsledky testování budou dále sloužit k optimalizaci nastavení a případným dalším úpravám, aby byla dosažena maximální účinnost a spolehlivost vizuálního systému.

Tab. 3: Srovnání dostupných sestav (kamer + objektiv) od firmy *Basler AG*

Parametr	**Sestava I**	**Sestava II**	**Sestava III**
Označení kamery	acA2500-14gm	acA2500-20gc	acA2440-20gc
Označení senzoru	MT9P031	PYTHON 5000	IMX264
Typ závěrky	Rolling Shutter	Global Shutter	Global Shutter
Barva	Černobílá	Barevná	Barevná
Frekvence [Hz]	14	21	23
Označení objektivu	C125-0418-5M-P	C10-1214-2M-S	C23-0824-5M-P
Ohnisk. délka [mm]	4	12,5	8
Clonové číslo	1,8 – 22	1,4 – 16	2,4 – 16
Rozlišení	5	2	5

Z dostupných možností je nejvíce vhodná Sestava III, která zahrnuje kameru *acA2440-20gc* a objektiv *C23-0824-5M-P*. Tato sestava nabízí řadu výhod, které odpovídají požadavkům projektu. Sestava III má stejné rozlišení jako Sestava I, což je důležité pro zachycení detailů v obrazech. Navíc má stejný typ závěrky jako Sestava II, což je *Global Shutter*, umožňující snímání obrazu bez zkreslení pohybu. S frekvencí až 23 snímků za sekundu je Sestava III nejrychlejší ze všech sestav. Barevná kamera umožňuje zachytit více informací o scéně a usnadňuje následnou analýzu a identifikaci objektů. Objektiv *C23-0824-5M-P* s střední ohniskovou vzdáleností a širokým rozsahem clonových čísel poskytuje flexibilitu při zaostřování a nastavení hloubky ostrosti v různých scénách. Celkově lze konstatovat, že Sestava III poskytuje vyváženou kombinaci vlastností, které odpovídají požadavkům projektu na rozlišení, typ závěrky, frekvenci, barevné zachycení a flexibilitu objektivu.

5.8 Výběr osvětlení a pozadí

V práci bylo zmíněno, že volba osvětlení je stejně důležitá jako výběr kamery a objektivu. Nestálost světelných podmínek může způsobit labilitu celého systému. Mohly by se navrhnout různé typy světel. Bylo by možné začít zadním osvětlením, přejít přes osvětlení s jasným zorným polem a temným zorným polem, až k nasvícení v ose kamery. Předmětem detekce nemá být pouze tvar polotovaru, ale v pozdější implementaci by také mohl být uvažován materiál a struktura povrchu. Z toho důvodu nepřipadá v úvahu zadní osvětlení, které pouze zobrazí tvar objektu, a také osvětlení s temným zorným polem, které zvýrazní především hrany předmětu. Realizovatelnost zamítnutých osvětlení je také náročnější.

Nejvíce vhodné by bylo difuzní světlo v ose kamery. Osvětlení tímto světlem způsobí, že kovový předmět bude velmi dobře viditelný bez odlesků. Nejjednodušší cestou jak zabránit odlesku na kovových předmětech je využití silně difuzního světla. Pokud jde o předměty nekovové, pomůže použití přidání polarizačního filtru na objektiv.[7] Příkladem difuzního světla je *DL-120* z portfolia firmy *SmartView s. r. o.* [38]. Uvedený produkt je dodáván v různých vlnových délkách, jeho intenzitu je možné nastavit pomocí kontroléru. Lze jej použít pro osvětlení předmětů s reflexním povrchem a detekci trhlin a jiných povrchových vad. Jmenovaným světlem může být zajištěno stabilní osvětlení, ovšem nemůže být použito z následujících důvodů. Světlo musí být v době pořizování snímku umístěno těsně nad polotovarem. Pokud by bylo takto umístěno trvale, značně se omezí manipulační prostor robotického ramene. Řešením by bylo umístit světlo na pohyblivou konstrukci, která by se před získáním snímku přiblížila nad předmět. Uvedené řešení je proveditelné, ovšem velmi nákladné a složité. Posledním důvodem je, že světlo dokáže nasvítit příliš malou plochu, která není dostatečná pro projekt. Největší dostupné světlo *DL-210* nasvítí kruhovou plochu o poloměru 210 mm.

Není-li možné použít kopulovité světlo z hlediska nízké zástavby, přichází možnost použití výkonného plochého světla. Firma *SmartView s. r. o.* nabízí tato světla pod označením *FL-320* [39]. Ploché světlo se používá často v aplikacích, kde robotické rameno odebírá předměty nebo k přímému homogennímu nasvícení pro detekci povrchových vad. Svými vlastnostmi se velmi podobá kupolovitému světlu, ale jeho výhodou je, že neomezuje manipulační prostor a dokáže osvítit větší plochu.[8]

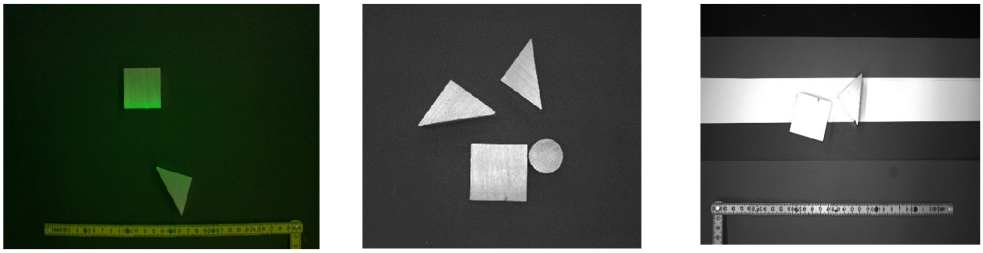
Proveditelnost je zaručena i s méně výkonným světlem. Pro pokrytí zadání postačí kruhové světlo *RL-100-II* od zmíněné firmy [40]. Jeho předními výhodami jsou: dodání v různých vlnových délkách, intenzitu je možné regulovat kontrolérem, zaručí homogenní nasvícení většiny povrchů. Vhodně je pro aplikace s kruhovými předměty a kontrolu defektů. Kruhové světlo *RL-100-II* poskytne dostatečné nasvícení plochy, ze

[7] Bylo vyzkoušeno přidání polarizačního filtru na objektiv. Ovšem změna v odlescích kovových předmětů se neprojevila. Důvodem je, že odražené světlo není polarizováno [41].

[8] Použití výkonného plochého světla se zdá jako nejlepší varianta. Problémem se stalo, že světlo nebylo v době vývoje aplikace k dispozici.

které budou předměty odebírány. Hlavním jeho přínosem je kompaktnost, tedy se nezmění prostor pro manipulaci robotického ramene.

Vybrané světlo bylo dodáno v různých provedeních, která se liší použitou vlnovou délkou. Testována byla konfigurace s bílým, modrým, červeným světlem. Vždy bylo zkoušeno jedno ze světel a jedno z pozadí. K volbě pozadí byly použity různobarevné podložky. Ze série testů, kde byly použity hliníkové polotovary, vyšla nejlíp kombinace bílého světla a velmi tmavého pozadí. Vhodnost zmíněné kombinace dokládá obr. 21. Jelikož nejsou uvažovány polotovary z jiných materiálu, je možné zvolit velmi tmavou podložku, která je kontrastní vůči hliníku. Pokud by materiál polotovarů byl různý, bylo by potřebné barvu podložky změnit. Nejvíc problematickým případem se jeví použití polotovarů z oceli. Tmavé polotovary by na pozadí zanikly a nemohly být detekovány. Na světlém pozadí by vynikly, ale hliníkové by nebylo možné určit.



Obr. 21: Hledání vhodné kombinace světla a pozadí, zleva: nevhodné, vhodné, srovnání

Na základě zjištění z testů lze odvodit důležitost správné volby osvětlení a pozadí pro dosažení optimálního kontrastu při detekci objektů. Výběr vhodné kombinace světla a pozadí je klíčový pro efektivní identifikaci a analýzu zkoumaných předmětů, a to zejména v kontextu rozmanitých materiálů a jejich optických vlastností. Zatímco bílé světlo a tmavé pozadí se ukázaly jako efektivní pro detekci hliníkových polotovarů, je nutné provést další úpravy, pokud by se používaly polotovary z jiných materiálů, aby se zajistil dostatečný kontrast a spolehlivá detekce. Tento poznatek podtrhuje význam systematického testování a přizpůsobení osvětlení a pozadí konkrétním podmínkám a požadavkům dané aplikace strojového vidění.

6 VÝVOJ DETEKČNÍHO SOFTWARU

V kapitole 5 byla probrána buňka *EDUset ONE* s důrazem na strojové vidění. Jeho možnosti implementace byly důkladně zváženy. Taktéž byl navrhnut vhodný hardware pro realizaci projektu. Pro určení vhodného hardwaru byla nalezena dostupná kombinace kamery, objektivu, světla a podložky. V následující kapitole bude probrán vývoj softwarové aplikace pro robotickou buňku. Hlavní důraz bude kladen na navrhnutí a implementaci různých metod pro detekci více druhů objektů semistrukturovaně rozmístěných na ploše. Navržené metody budou porovnány a proces vyhodnocen. Vznik programu, detekující polotovary, byl stěžejní pro tuto práci. Pokud by byla v buňce použita chytrá kamera, byl by úkol jednoduší. Odpadla by velká část programování a zůstalo mnoho práce s konfigurací. Nicméně z hlediska modulového chování je lepší volba kamery společnosti *Basler AG*.

Velmi důležitým krokem bylo zvolit vhodné frameworky pro detekci rozmístěných polotovarů. Bylo stanoveno několik kritérií na jejich základě byly zvoleny. Prvním kritériem je výkon tedy, jak rychle a přesně je framework schopný detekovat objekty ve scéně. Dalším prvkem je jednoduchost použití. Jednoduchostí lze rozumět, jak snadné je detektor nasadit do softwarového projektu. Třetím důležitým kritériem je flexibilita, která udává, jak možné je upravit framework pro vlastní použití. Předposledním aspektem je podpora vývojářské komunity a posledním je dostupnost předtrénovaných modelů. Na základě zmíněných kritérií byly vybrány následující detektory: YOLOv8, SSD300 a *Faster* R-CNN. K jejich výběru přispěl i fakt, že se jedná o často srovnávané frameworky a jejich architektura je velmi dobře popsána ve zdroji [8].

Pro porovnání výkonosti detektorů je nezbytné uvést, na jaké výpočetní stanici byly natrénovány a kde probíhalo jejich testování. Veškerý vývoj probíhal na stolním počítači s jednou grafickou kartou *GeForce RTX EAGLE 12G*, s procesorem *AMD Ryzen 5 5600X* a s 32 GB RAM. Pro vývoj softwarové aplikace bylo využito prostředí *PyCharm* ve spojení s *Windows Subsystem for Linux* (WLS), což umožnilo provádění vývoje na operačním systému *Linux* distribuce *Ubuntu*. Pro správu virtuálního prostředí a správu balíčků byla použita distribuce *Anaconda*. Tato kombinace prostředí poskytla efektivní prostředí pro vývoj a správu projektu v prostředí Linuxu.

Kromě toho budou v této kapitole zkoumány nejvhodnější metody a techniky pro detekci objektů ve scéně. Každý z vybraných frameworků má své vlastní specifické výhody a omezení, a proto je důležité porovnat jejich výkonnost a efektivitu v různých situacích. Cílem je nalézt optimální kombinaci frameworků a metod, která bude nejlépe vyhovovat požadavkům a specifikacím aplikace. Tato kapitola je klíčová pro celkové pochopení procesu vývoje detekčního softwaru a pro zhodnocení jeho účinnosti a výkonnosti v praxi.

6.1 Tvorba vlastního datasetu

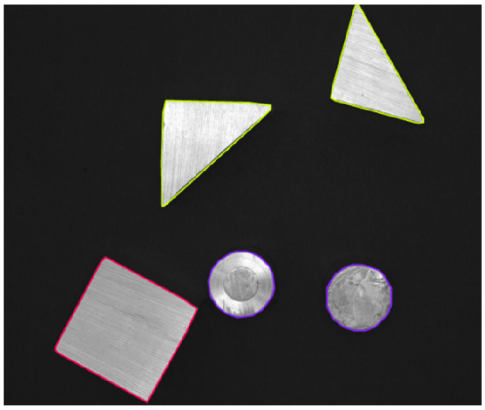
Než mohou být jednotlivé frameworky naučeny detekci polotovarů, je nutné získat množinu trénovacích dat. Jelikož se jedná o detekční úlohu, je výroba vlastního datasetu složitější než v případě klasifikační úlohy. V praxi znamená výroba datasetu pro klasifikační úlohu, pouze rozdělení stejných dat do složek [41]. Při detekční úloze je proces složitější, protože neuronová síť nepredikuje pouze, co se na obrázku nachází, ale je schopna určit, co se v obrázku nachází a na jakém místě to je[9]. Dataset se skládá z obrázků polotovarů rozmístěných na podložce. Množina dat byla získána v prostředí buňky *EDUset ONE* a všechny snímky byly provedeny na jednom místě za stejného osvětlení. Tento fakt by mohl vést k neúspěchu, protože pokud je konstrukce s kamerou přesunuta do jiného místa robotické buňky, může nastat odlišná světelná podmínka, která povede k nezdaru. Tento problém se vyskytuje často v průmyslových halách, kde je segment výrobní linky, obsahující strojové vidění, přesunut z místa pod světlíkem do temného rohu. Zmíněnému problému se lze elegantně vyhnout použitím vhodného osvětlení.

Po vytvoření 191 snímků různě rozmístěných polotovarů na podložce, bylo nutné každý obrázek správně popsat. K dispozici je nepřeberné množství anotačních aplikací, některé z nich jsou dostupné v rámci webovského rozhraní. Jako první byla vyzkoušena aplikace *Label Studio* od společnosti *HumanSignal, Inc*. Nicméně toto rozhraní nebylo přizpůsobeno pro YOLOv8. Ke spuštění aplikace bylo nutné kontaktovat tým podpory a v rámci samotného procesu označování nebyl nasazen žádný podpůrný model, tedy celý proces nabral na časové náročnosti. Důležité je zmínit, že vybrané detekční frameworky nemají kompatibilní formáty datových sad. SSD a *Faster* R-CNN využívají formát, kde je *bounding box* definován souřadnicemi svých rohů [x_{max} y_{max}, x_{min}, y_{min}]. Na rozdíl tomu YOLOv8 používá formát, kdy je souřadnicemi určen střed rámečku, který je doplněn hodnou své šířky a své výšky. Zmíněný formát může pak vypadat tímto způsobem: [x, y, šířka, výška]. Hlavní podmínkou pro výběr anotační aplikace byl požadavek, aby dataset nemusel být anotován dvakrát. Dalším kritériem byla možnost vytváření masek a posledním byla implementace modelu, který anotaci usnadní.

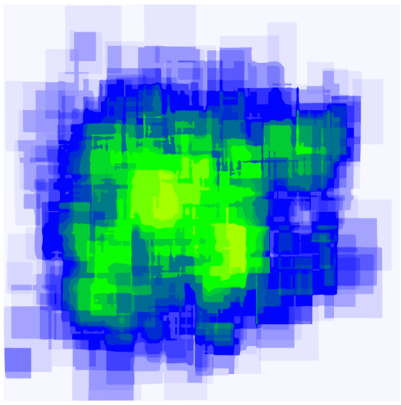
V uvedených podmínkách nejlépe obstála webovská aplikace od společnosti *Roboflow, Inc*. Proces anotování je usnadněn pomocí vestavěného modelu, který sám předikuje ohraničující rámečky. Popisy vytvořené tímto způsobem je nutné zkontrolovat, ale celkový proces to značně urychlí. Samotná aplikace je velmi intuitivní a člověku, začínajícímu s detekčními frameworky, velmi schopně napovídá. Ukázka jednoho anotovaného snímku v prostředí *Roboflow* je na obr. 22. Po anotování všech snímku je aplikace schopna rozdělit data na množinu trénovací, validační a testovací. V případě obrázků polotovaru byla vstupní množina obsahující 191 snímků rozdělena v poměru 75:15:10. Takovéto rozdělení bylo navrhnuto samotnou aplikací. Před finálním exportem je možné snímky upravit například pomocí oříznutí, převedení do odstínů šedi nebo

[9] Kromě obrázku jsou uvažována i videa, protože každé video, je určité množství snímků získaných za sebou v určité časové periodě.

změnění velikosti. Je zde možnost využití vestavěné augmentace dat, která nabízí mnohé ze základních bezpečných technik. Nicméně pro vlastní dataset této možnosti využito nebylo. Augmentační techniky, které byly použity pro trénování modelů budou probrány dále v textu. Poslední zajímavostí je, že prostředí nabízí možnost zobrazení informací o datasetu. Je možné se dozvědět, kolik prvků každé kategorie se v datové množině nachází a dokonce jakým způsobem jsou umístěné. Umístění všech prvků je zobrazeno na obr. 23, z něhož je možné dopředu odhadnout, že modely budou méně jisté v predikci s větší vzdáleností od středu.



Obr. 22: Anotace v prostředí *Roboflow*



Obr. 23: Rozmístění polotovarů v datasetu

Dostupná je aktuální verze datasetu z přiloženého odkazu[10]. V datasetu se nacházejí tři třídy objektů tak, jak tomu bylo stanoveno v kapitole o 5.6, která se věnuje volbě vhodných polotovarů určených k detekci. Tyto třídy objektů jsou pečlivě vybrány s ohledem na požadavky a specifikace dané aplikace.

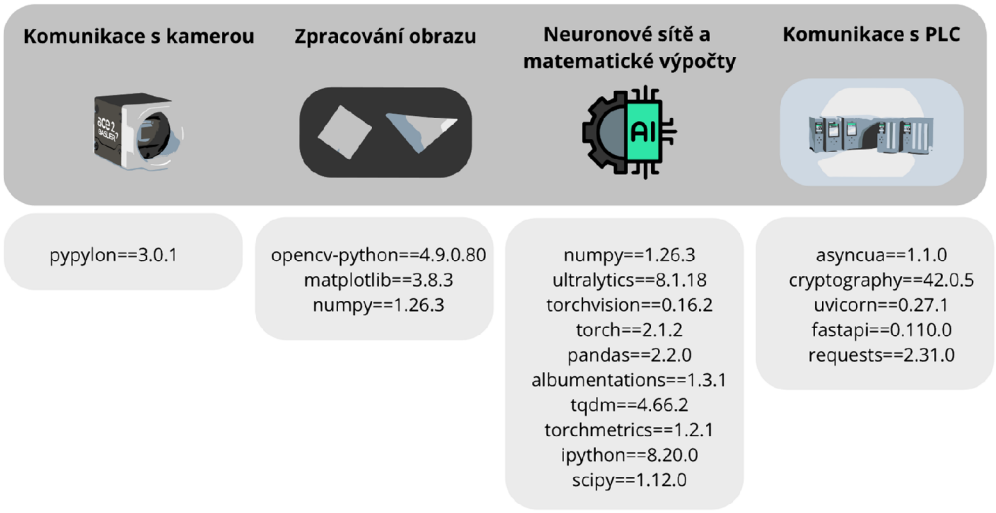
6.2 Knihovny pro implementaci v jazyku Python

V rámci celého projektu je používána verze programovacího jazyku *Python 3.11*. Samotný jazyk je velmi používaný z několika důvodů. Prvním důvodem je, že je pro začátečníky velmi dobře naučitelný. Za druhé je objektově orientovaný a v neposlední řadě podporuje velkou řadu knihoven a frameworků vytvořených uživateli. Poslední uvedené vlastnosti je využito napříč celým projektem. Nezbytné bylo zvolit vhodné balíčky, kterých bude využito. Ze zkušenosti lze tvrdit, že skoro každá programátorská výzva má připravený svůj vlastní baliček napsaný v jazyku *Python*.

Vyvíjenou softwarovou část aplikace je tedy nejprve nutné rozdělit do menších modulů. Následně ke každému modulu vybrat vhodnou knihovnu nebo framework. Rozdělení robustní aplikace na menší úkoly je zobrazeno na obr. 24. Dále je ke každému modulu uvedeno jaké balíčky byly použity. Je zřejmé, že hlavních importů je mnoho a

[10] Odkaz na aktuální verzi datasetu: https://www.kaggle.com/datasets/janzmrzly/eduset-one-dataset

budou probrány nejdůležitější, bez kterých se fungování aplikace neobejde. Uvedené balíčky *matplotlib*, *tqdm*, *ipython, pandas, torchmetrics* a *cryptography* jsou víceméně nadstavby pro lepší zobrazení nebo pro zabezpečenou komunikaci. Není nutné se jim více věnovat. Při vývoji každého modulu byly rovněž zohledněny požadavky na efektivitu a snadnou rozšiřitelnost, což bylo zajištěno volbou vhodných knihoven a frameworků.



Obr. 24: Knihovny a frameworky využité v jednotlivých modulech aplikace [32; 42]

Využití kamery od společnosti *Basler AG* se ukázalo jako velmi dobré rozhodnutí i z důvodu, že korporací je vyvíjeno *application programming interface* (API) s názvem *pypylon*. Není nutné pro komunikaci a získávání obrazových dat z kamery implementovat vlastní knihovnu. Firma také poskytuje velké množství volně dostupných ukázkových kódů. Pro komunikace vlastního softwarového řešení a kamery nebyl použit jiný import. Dalším významnými importy v odrážce pro zpracování obrazu jsou *numpy* a *opencv-python*. *Numpy* složí k usnadnění výpočtů v prostředí jazyka *Python*. Ovšem *opencv-python* je jednou z nejdůležitějších knihoven pro zpracování obrazových dat. Její funkce jsou tak rozsáhlé, že poskytuje operace od základních operací s obrázky přes hledání hran po rozpoznávání obrázků. Pro řešení projektu se, ale ukázaly nejvíce stěžejní dvě funkce: hledání nejmenšího rámečku a PCA.

Následuje velký oddíl importů, který byl využitý pro implementaci konvolučních neuronových sítí a metrik pro určení jejich výkonosti. Mezi tyto importy patří *ultralytics*, *torchvision*, *torchmetrics* a *scipy*. Velký balíček *albumentation* byl využit pro augmentaci dat. Aby mohla být aplikace kompletní, je nutné stanovit komunikaci v souladu s trendy Průmyslu 4.0. Požadavkem je, aby aplikace strojového vidění byla schopna komunikovat s programovatelným automatem, který se nachází v robotické buňce *EDUset ONE*. Nejvhodnější řešení se jeví využití komunikačního protokolu OPC UA, proto bylo nutné použít knihovnu *asyncua*. Na závěr by měl být zmíněný import *fastapi*, jehož fungování bude vysvětleno v následující kapitole, která se věnuje samotnému fungování softwaru.

V posledním odstavci této kapitoly bude vysvětleno uspořádání repositáře dostupného z přiloženého odkazu[11]. Ve zobrazení nejsou uvedeny všechny soubory, je to z toho důvodu, že souborů je poměrně velké množství a pro stručnost jsou nahrazeny výpustkou. Po dvou úvodních souborech *LICENSE* a *README.md* následuje knihovní balíček s názvem *eduset*., který tvoří jádro celého repositáře. Zároveň je i jádrem celé aplikace. Veškeré hlavní funkce jsou uvedeny ve zmíněném balíčku. Jeho složení je takové, že obsahuje podložky *fasterrcnn*, *ssd*, *yolo* a *utils*. První tři složky jsou určeny pro trénování, evaluaci a nasazení detekčních frameworků. Poslední složka *utils* je určená pro podpůrné funkce, ovšem je zde zahrnuta komunikace s kamerou a sběr obrazových dat.

```
├── LICENSE
├── README.md
├── eduset
│   ├── __init__.py
│   ├── fasterrcnn
│   │   └── ...
│   ├── ssd
│   │   └── ...
│   ├── yolo
│   │   └── ...
│   └── utils
│       └── ...
├── main
│   ├── api.py
│   ├── client.py
│   └── vision.py
├── requirements.txt
└── samples
    ├── collect_dataset.ipynb
    └── ...
```

Následující složkou je *main*, kde je již vytvořený celý software určený k detekci semi-strukturovaně rozmístěných polotovarů z plochy pomocí metod umělé inteligence. Následuje soubor *requirements.txt*, který obsahuje potřebné balíčky pro chod aplikace. Na závěr je složka *samples*, ta obsahuje ukázky ve formě *jupyter* zápisníků. Jsou zde příklady od trénování jednotlivých modelů, přes jejich validaci po sběr dat z kamery. Pro podrobnější pochopení struktury repositáře je nezbytné jej shlédnout v rámci platformy *Github*.

6.3 Implementace a porovnání detekčních frameworků

V této části kapitoly je probrána implementace vybraných detekčních frameworků. Jak bylo zmíněno budou srovnány tři přední detektory objektů: SSD, *Faster* R-CNN a *YOLOv8s*. Každý ze nich vyniká v lehce odlišných oblastech. V kapitole 4.8 bylo probráno srovnání ve dvou odlišných aplikacích. V prvním případě byla výzva založena

[11] Odkaz na aktuální verzi datasetu: https://github.com/JanZmrzly/eduset-vision

na detekci různých léků, určených k perorálnímu podání, v druhém případě bylo nutné detekovat skleníky a změny v jejich okolí [29; 30]. V obou případech byl vybrán jako nejvíce vhodný detektor z rodiny YOLO. Dostupných článků věnujících se srovnání především zmíněných detektorů je velké množství. Ve všech porovnáních rodina YOLO vždy nevyhrává. Mohou být uvedeny další dva případy, ze kterých lépe vychází framework SSD. V prvním případě je detekován tenisový míček při podání, druhý článek se zabývá rozpoznáváním typů automobilů [43; 44]. V obou článcích je uvedeno, že aplikace, na kterých byly detektory testované jsou v reálném čase. Závěry ve všech uvedených článcích jsou vyvozeny především na základě výkonosti implementace – mAP a FPS.

Mohou tedy být určena kritéria, na základě nichž bude vybrán nejvíce vhodný framework určený k detekci semi-strukturovaně rozmístěných polotovarů z plochy. Prvním kritériem, shodujícím se s články, je posouzení na základě výkonu. Druhou zkoumanou vlastností bude jednoduchost implementace. Za třetí bude brána v úvahu flexibilita – jak možné je framework přizpůsobit specifickým potřebám aplikace. Předposledním aspektem je podpora vývojářské komunity a posledním je dostupnost předtrénovaných modelů. Vybraná kritéria nejsou uvedena, jak budou probrána v rámci práce, ale na základě jejich důležitosti pro projekt.

6.3.1 Podpora vývojářské komunity a dostupnost modelů

V podkapitole jsou spojena dvě porovnávací kritéria. Prvním je podpora vývojářské komunity a druhým je dostupnost předtrénovaných modelů. Vývojáři vždy pomůže pokud má na koho se obrátit. Nutné je brát v potaz, že pokud dojde k zaseknutí na některé z implementačních překážek, nemusí programátor úkol vůbec dokončit. V případě YOLOv8 existuje aktivní *Discord* server provozovaný přímo firmou *Ultralytics*. Podpora vývojářské komunity je pak nesrovnatelně lepší oproti zbylým frameworkům.

K trénování bylo přistoupeno metodou *transfer learning*, tedy síť nemá inicializované počáteční váhy na náhodné hodnoty, ale byla předtrénována na jedné z velkých datových množin. K těmto krokům je nezbytné mít k dispozici model konvoluční neuronové sítě a váhy, které slouží ke zmíněné inicializaci. Dostupnost všech detektorů je velmi dobrá, tudíž stejná.

6.3.2 Jednoduchost implementace a flexibilita

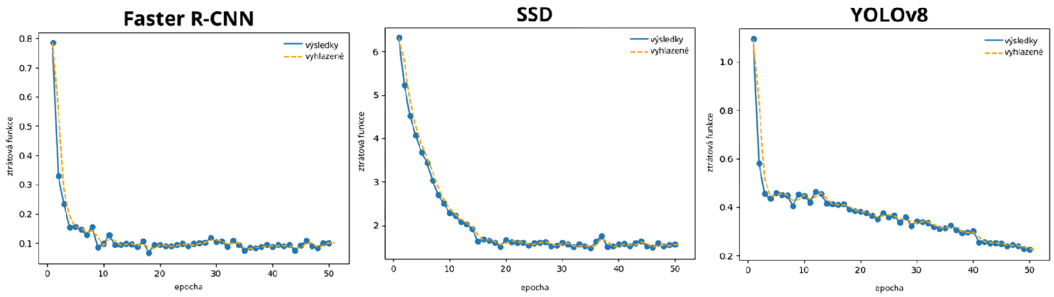
V případě jednoduchosti realizace lze velmi dobře určit, že nejsnáze implementovatelný je model *YOLOv8s*. Samotné provedení nezabere více než několik řádků. Možnost takové jednoduchosti je zapříčiněná dostupností knihovny *ultralytics.* Opravdu lze tvrdit, že natrénovat a následně nasadit framework YOLOv8 zvládne i osoba, která nemá větší zkušenosti s programováním. Před samotným trénováním sítě dokonce není nutné data rozšiřovat pomocí knihovny *albumentation*, data jsou augmentovaná v rámci zavolání trénovací funkce. Realizace SSD a *Faster* R-CNN je daleko komplikovanější. Modely jsou dostupné, ale ve velice syrovém provedení. Pro jejich implementaci je nutná pokročilá znalost programovacího jazyka *Python* a knihovny *PyTorch*. Pro správné

fungování musí být model upraven odstraněním klasifikační hlavy a přidáním nové. Pro načítání dat musí být vytvořen nový objekt a další pro sledování metrik sítě. Co bylo zvládnutelné v případě YOLOv8 na několika řádcích zdrojového kódu, je nutné napsat nižší stovky řádků zdrojového kódu. Nevyhnutelná augmentace trénovacích a validačních dat je proveditelná pomocí knihovny *albumentation*. Výhodou je, že oba frameworky je možné realizovat velmi podobnou cestou.

V případě flexibility se může zdát, že větší výhody mají frameworky SSD a *Faster* R-CNN. Jelikož jsou modely surové, lze mít dobrou kontrolu nad zdrojovým kódem. Jednotlivé implementace jsou skvěle škálovatelné. Nicméně YOLOv8 je taktéž flexibilní a nabízí možnost značné kontroly. Pokud je uvažována flexibilita z pohledu přechodu na konvoluční neuronovou síť segmentující obraz, lze jednoznačně vyzdvihnout YOLOv8[12]. Segmentační úloha se liší od detekční a jejím hlavním úkolem je určit, jaký objekt se v obraze nachází, kde se nachází a jaká je jeho maska.

6.3.3 Výkony frameworků na vlastním datasetu

Nejčastěji jsou modely srovnávány na základě jejich výkonosti. Je to také hlavní kritérium pro výběr modelu v rámci práce. Pro porovnání výsledků rozpoznávání polotovarů byl vyhodnocen výkon jednotlivých frameworků. Po natrénování modelu mohl být vybrán nejlepší model s nejvyšší přesností a nejrychlejším zpracováním. Na následujícím obr. 25 je uvedena trénovací ztrátová funkce vybraných modelů v závislosti na tréninkové epoše. Může být konstatováno, že přes vlastní realizaci se povedlo implementovat všechny modely tak, aby ztrátová funkce byla klesající.



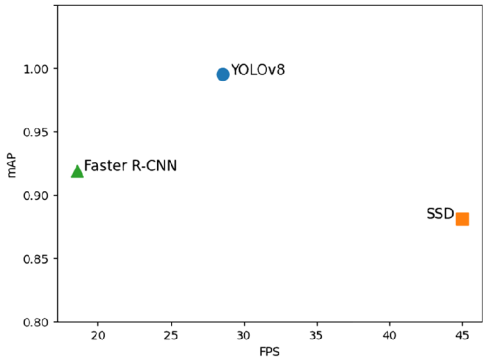
Obr. 25: Srovnání vybraných modelů na základě ztrátové funkce při trénování

V případě *Faster* R-CNN stačilo několik epoch ke značnému snížení ztrátové funkce. Nejmenší hodnota se tak zastavila na čísle 0,065 v 18. epoše. K natrénování modelu by mohlo postačit méně epoch, ovšem doba trénování všech epoch byla 14 minut a 43 sekund. Druhým nasazený modelem byl SSD. Jeho ztrátová funkce dosáhla nejlepší hodnoty 1,47 a to ve 35. epoše. Trénování detektoru SSD trvalo 3 minuty a 12 sekund

[12] Segmentační úloha v oblasti počítačového vidění se zabývá rozdělováním obrazu na jednotlivé segmenty nebo regiony, které mají podobné vlastnosti nebo charakteristiky. Cílem segmentace je identifikovat a oddělit jednotlivé objekty nebo části objektů v obraze. To může zahrnovat rozlišování mezi objekty a pozadím, extrakci hranic objektů nebo rozdělení obrazu na jednotlivé složky na základě různých vlastností, jako jsou barva, jas, textura nebo tvar.

Posledním nasazeným frameworkem byl YOLOv8. Trénovací ztrátová funkce se zastavila na nejmenší hodnotě 0,225 a to v poslední 50. epoše. Na základě grafu lze konstatovat, že již od 3. epochy se ztrátová funkce pohybuje okolo 0,450. Model *YOLOv8s* byl trénován 5 minut a 21 sekund.

Další výkonnostní srovnání je zobrazeno v obr. 26. V tomto grafu jsou prezentovány výsledky detekčních modelů v závislosti na dvou klíčových metrikách: mAP na ose y a FPS na ose x. Každý bod na grafu reprezentuje jeden implementovaný framework. Bod s vyšším mAP a vyšším FPS je považován za přínosnější, neboť dosahuje lepší přesnosti detekce a zároveň je schopen zpracovat větší množství snímků za sekundu. Ve snaze nalézt optimální kompromis mezi přesností a rychlostí detekce jsou hodnoceny jednotlivé modely. *YOLOv8s* dosáhl mAP 0,995 při 28,55 FPS. Nejnižší hodnoty byly zaznamenány u detektoru *Faster* R-CNN s mAP 0,919 a 18,54 FPS. Framework SSD dosáhl nejvyšší rychlosti 45,00 FPS, avšak jeho mAP dosáhla nižší hodnoty 0,881. V poslední řadě lze konstatovat, že hodnoty FPS a mAP korelují s dobou tréninku a ztrátovou funkcí při trénování.



Obr. 26: Výkonost konvolučních neuronových sítí při detekci polotovarů

6.3.4 Shrnutí porovnání detekčních frameworků

V předcházející části kapitoly byly probrány výhody a nevýhody vybraných frameworků. V této kapitole je nezbytné vybrat jeden nejvíce vhodný pro projekt. Před samotným výběrem jsou shrnuty zkoumané vlastnosti zvolených detektorů. Pro dobrou přehlednost byla všechna zkoumaná kritéria uvedena v tab. 4. Detekční modely byly srovnány na základě zkoumaných kritérii. Z hlediska důležitosti pro projekt jsou seřazená v tabulce od nejvíce důležitého po nejméně.

Na základě výkonosti je jako nejvíce vhodný detektor YOLOv8. Jeho výběrem je dosaženo kompromisu mezi rychlostí a přesností, která je pro projekt vyžadována. V ohledu výkonosti může modelu konkurovat pouze SSD, které dokáže detekovat polotovary vyšší rychlostí. Nicméně přesnost SSD je nižší a výběr není založen pouze na rychlosti. Pokud by byla hlavním kritériem volby rychlost detekce, bylo by vybráno SSD. Rychlost *YOLOv8s* je pro projekt dostatečná a je podpořena přesnou detekcí. Dalším důležitým kritériem je flexibilita. Z hlediska flexibility bylo jako nejvíce vhodné opět

vybráno *YOLOv8s*. Hlavním důvodem je možnost rychlého přechodu z detekční úlohy na úlohu segmentační[13]. Pro zmíněnou změnu není nutné přepisovat velké množství zdrojového kódu, stačí pouze změnit typ výchozí sítě při tvorbě vlastního modelu. Tento fakt úzce souvisí s jednoduchostí implementace. Při využití knihovny *ultralytics* se realizace značně usnadní a urychlí. Pro implementaci frameworku SSD a *Faster* R-CNN je nezbytná pokročilá znalost jazyka *Python* a knihovny *PyTorch.* Předtrénované modely jsou pro všechny detektory dostupné, nelze tedy jednoznačně rozhodnou, který v rámci kritéria vyhrává. Ovšem zcela jasně lze tvrdit, že nejlepší komunitní podporu má YOLOv8.

V technickém kontextu by to mohlo působit tak, že SSD a *Faster* R-CNN nejsou schopny konkurovat modernímu frameworku YOLOv8. Nicméně, každý z těchto přístupů má své pozitivní vlastnosti a rozhodnutí o výběru závisí na konkrétních požadavcích a výzvách daného projektu. SSD vyniká vysokou rychlostí zpracování obrázků, což je užitečné pro situace, kde je prioritou efektivita v reálném čase. *Faster* R-CNN naopak exceluje v přesnosti detekce, což může být klíčové pro úlohy, kde je důležitá maximální spolehlivost. Avšak, *YOLOv8s* nabízí výhodu vyvážené kombinace mezi rychlostí a přesností, což ho činí atraktivním řešením pro mnoho aplikací. Celkově lze tvrdit, že volba rodiny YOLO není chybou. Tento framework nabízí jednoduchou implementaci, vysokou podporu a vyvážený poměr mezi rychlostí a přesností detekce, což je pro mnohé aplikace ideální kombinace vlastností.

Tab. 4: Porovnání vybraných vlastností detekčních frameworků

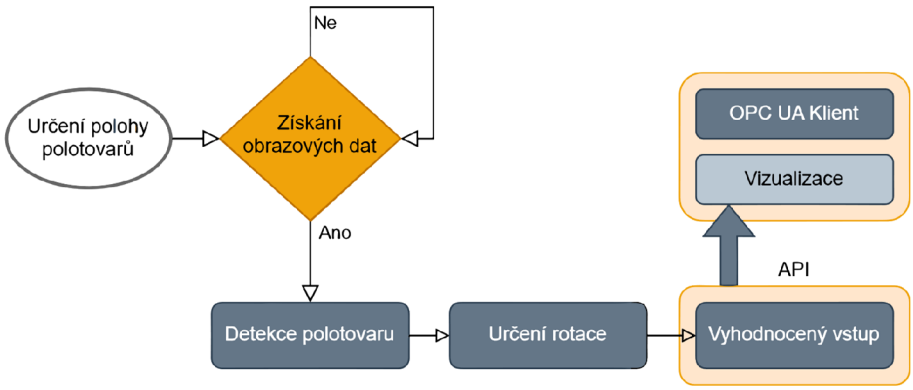
Kritérium	**SSD**	**YOLOv8s**	***Faster* R-CNN**
mAP	0,881	0,995	0,919
FPS	45,00	28,55	18,54
Flexibilita	Dobrá	Výborná	Dobrá
Implementace	Dobrá	Výborná	Dobrá
Podpora komunity	Dostatečná	Výborná	Dostatečná
Dostupnost modelů	Výborná	Výborná	Výborná

6.4 Strategie určení polohy polotovaru

V předchozí části diplomové práce byly porovnány detekční modely objektů. Z vybraných detektorů byl zvolen vhodný framework. Nyní je nutné spojit všechny jednotlivé části práce a vytvořit funkční software určený k detekci semi-strukturovaně rozmístěných polotovarů z plochy. Celý proces bude fungovat dle následující rozvahy.

[13] Využití segmentace se pozitivně projeví pro určení rotace polotovaru.

Řídícím členem robotické buňky je PLC *SIMATIC S7-1500F,* všechny ostatní členy by měly s programovatelným automatem komunikovat – měly by být řízeny. Vstupním prvkem do tvořeného programu je signál vyslaný z PLC. Tento signál bude náběžná hrana typu *bool*. Po dobu trvání signálu se bude vlastní program cyklicky vykonávát, po jeho odeznění se dokončí rutina a program přestane pracovat. Je-li programem zachycen signál, znamená to, že je vyslán požadavek pro určení polohy polotovarů. Tento stav je startovním pro vlastní implementaci programu. Jako první krok je získání obrazových dat z kamery[14]. Dokud nejsou získána obrazová data probíhá rutina určená k jejich získání. Jakmile je získán obrázek, je poskytnut dále k detekci polotovarů. Rozpoznání provádí natrénovaný model vybraného frameworku YOLOv8. Údaje o nalezených polotovarech jsou předány dále funkci pro určení rotace. Výsledek je uspořádán jako slovník s je možné k němu přistoupit přes API. Dotazy přes API využívá OPC UA klient, který komunikuje s vestavěným OPC UA serverem na řídícím PLC. Návrh fungování je představen na následujícím obr. 27.



Obr. 27: Architektura vlastního softwaru pro detekci polotovarů

V rámci implementace nastalo několik zásadních momentů, které je třeba pro lepší pochopení fungování vlastního softwaru okomentovat. V rámci detekce polotovarů bylo vhodné zvolit výchozí neuronovou síť *YOLOv8s-seg*. Bylo nutné změnit úlohu z detekční na segmentační z toho důvodu, že z predikovaných masek se lépe určuje rotace objektu. Pokud jsou polotovary v těsném shluku, pak nejsou jednoznačně rozlišitelné jeho hranice. Při použití různých detektorů hran nastává problém, že objekty splynou do jednoho. Pro učení rotace je lepší využít masku objektu.

Vstupem pro určení rotace objektu je maska polotovaru. Predikovaná maska, často není dokonalá a je potřeba ji vyhladit. Maska je v podobě souřadnic bodů a tyto body lze proložit vhodným polynomem, tímto procesem se dosáhne hladších hran masky. Polotovary jsou rozděleny do dvou skupin. První skupinou jsou polotovary jako kruh, čtverec. Ty vykazují symetrii podle dvou os a jejich body jsou rovnoměrně distribuovány

[14] K dispozici je několik scénářů, jak dat z kamery snímat. Nejběžnějším případem je, že se provede jeden snímek a ten je poskytnut dále. Jiným případem je, že snímků je uděláno několik a poslední platný je poskytnut. Tímto způsobem se předejde vypadnutí dat pří špatné komunikaci.

od těžiště. Druhou skupinou je tvar – trojúhelník, který je symetrický podle jedné osy a jeho body nejsou tak dobře distribuovány vzhledem k těžišti. Pro určení rotace první skupiny je použita funkce ohraničení nejmenším obdélníkem a pro druhou skupinu je rotace stanovena pomocí PCA. Výsledkem je datová struktura slovník poskytující informace o typu objektu, souřadnicích jeho středu a rotaci.

Posledním probranou částí zdrojového kódu je vysvětlení komunikace. Získané informace jsou v datové struktuře slovník, aby mohly být dále poskytovány prostřednictvím API. Řešení také mohlo být provedeno pomocí dočasných souborů, do kterých by byla data zapisována a následně z nich čtena. Vzhledem k záměru robustnosti aplikace bylo přistoupeno k vytvoření příslušného koncového bodu. Takové řešení umožňuje následné rozšíření o vizualizaci, případně vzdálenou konfiguraci programu.

6.5 Komunikace prostřednictvím OPC UA

Posledním technickým aspektem při realizaci vlastního detekčního softwaru je stanovení komunikace mezi aplikací strojového vidění a programovatelným automatem *SIMATIC S7-1500F*. PLC je řídícím členem celé robotické buňky, a tedy je důležité, aby existovala možnost vlastní detekční software ovládat. Nejvhodnějším komunikačním protokolem pro aplikaci byl zvolen OPC UA. Jeho použití je v souladu s trendy Průmyslu 4.0. Pro implementaci je dostupných několik knihoven vytvořených v různých programovacích jazycích. Z hlediska ucelenosti projektu bylo rozhodnuto, že bude využit objektově orientovaný programovací jazyk *Python* a knihovna *asyncua*.

PLC, které se nachází v buňce *EDUset ONE*, disponuje vestavěným OPC UA serverem. V rámci vlastního řešení bude vytvořen klient, který se připojí k serveru. Provedení klienta bude přizpůsobeno pouze tomuto projektu. Nebude se jednat o robustní modulární aplikaci, ale pouze o jednoúčelový skript. Myšlenka je taková – v rámci PLC je vytvořena datová struktura – obr. 28. Datová struktura odpovídá přesně jednomu detekovanému prvku, tedy její proměnných je součástí název, souřadnice x a y těžiště a určená rotace v radiánech. Následně je v rámci proměnných PLC vytvořeno pole skládající se z vlastních datových struktur o nespecifikované velikosti[15]. K dispozici je přepisovatelný zásobník *buffer,* který je měněn na základě klientského skriptu.

EdusetOneVision

	Name	Data type	Default value	Accessible f	Writa...	Visible in	Setpoint
1	name	String	''	☑	☑	☑	☐
2	x	UInt	0	☑	☑	☑	☐
3	y	UInt	0	☑	☑	☑	☐
4	angle	Real	0.0	☑	☑	☑	☐

Obr. 28: Datová struktura pro komunikaci prostřednictvím OPC UA

[15] Velikost pole závisí pouze na vývojáři. Prvky v poli se přepisují na základě detekovaných prvků. Je tedy ponecháno pouze na něm, o kolika polotovarech informaci potřebuje.

Nyní je v programovatelném automatu dostupná proměnná typu pole, která bude sdílena prostřednictvím OPC UA protokolu. Pro nastavení serveru existuje množství návodů, proto nebude uvedeno krok po korku, jak OPC UA server v PLC nastavit. Důležité je pouze, aby byla nastavena proměnná typu pole ke komunikaci. Je-li konfigurace provedena správným způsobem, neměl by být problém získat informace o uzlu, kterým je zásobník. Informace musí být o místě, kde se v adresním prostoru uzel nachází. Její znalost, je nezbytná pro správné nastavení klientské strany.

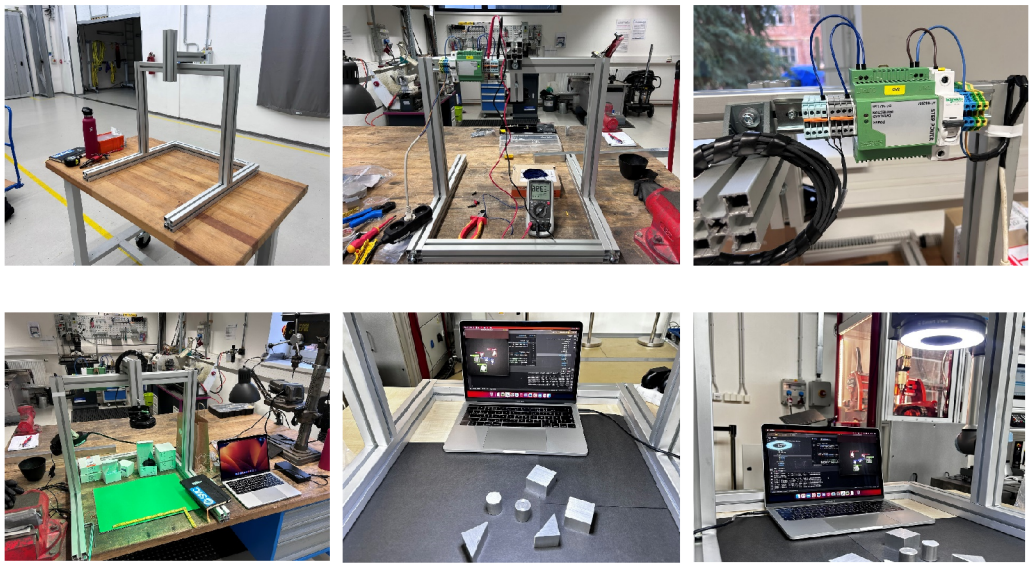
Klient dokáže přepisovat informace o detekovaných polotovarech přímo v paměti PLC. Nejprve jsou tedy získány zprávy o polotovarech pomocí detekčního programu, které jsou dále šířeny komunikací mezi OPC UA serverem a klientem. Velmi podobné nastaveni lze provést pro vstupní signál, kterým počíná činnost detekčního programu. Změna je pouze taková, že klient vstupní signál nemění a jeho hodnota je pouze na řídícím PLC. Klient musí jen znát, zda je signál aktivní či nikoliv. Provedení je tedy jednodušší než v případě *bufferu*. Klíčovou podmínkou je správné nastavení na straně klienta a získání informací o uzlu v adresním prostoru.

Velkým benefitem je, že ve vlastním provedení jde nastavit vyšší stupeň zabezpečení komunikace mezi serverem a klientem. Aplikace by fungovala, pokud by zabezpečení nebylo žádné, ale v moderních továrnách se velmi dbá na informační bezpečnost, a proto je možnost šifrování vítaná. V adresáři repositáře jsou výchozí sobory, sloužící k zabezpečené komunikaci. Nicméně je možnost vygenerovat a nasadit vlastní.

7 ZHODNOCENÍ A PŘIPOMÍNKY KE ZLEPŠENÍ

Kapitola uvádí zhodnocení vlastní práce v kontextu nastudované teorie a elektronických článků. Dále jsou uvedeny příklady, který by mohly vést ke zlepšení práce. Zmíněné případy mohou sloužit jako návod, jakým způsobem by mohlo být na práci navázáno.

Posledním krokem realizace bylo otestování funkčnosti s reálnými polotovary a s fyzickým hardwarem. Všechny testy proběhly úspěšně v prostředí robotické buňky *EDUset ONE*. Celý proces vývoje ze zachycen na několika snímcích v obr. 29. Uspořádání je chronologické. Obrázky naznačují vývoj aplikace strojového vidění pro buňku *EDUset ONE*. Prvním počinem bylo vytvoření hliníkové konstrukce z profilů od společnosti *MiniTech*. Následovalo přidání 24 V zdroje a jističe. Dalším krokem bylo správné nastavení scény. To obsahovalo vhodně zvolit kameru, osvětlení a pozadí. Předposlední snímek naznačuje tvorbu vlastního datasetu a poslední již finální testování v rámci robotické buňky. Při testování se povedlo detekovat předložené polotovary a jejich poloha byla odeslána prostřednictvím protokolu OPC UA do řídícího programovatelného automatu.



Obr. 29: Proces vývoje aplikace strojového vidění v robotické buňce *EDUset ONE*

V rámci práce byly také porovnány tři detekční frameworky: YOLOv8, *Faster* R-CNN a SDD. Z nastudované literatury bylo jasné, že detektor z rodiny R-CNN nezvládne z povahy zadání konkurovat zbylým frameworkům. V porovnání SSD a YOLOv8 vyšlo lépe YOLOv8, které je pomalejší při detekci než SSD, ale nabízí větší přesnost. Zvolené bylo také z důvodu, že přechod od detekční úlohy z segmentační je jednodušší než v rámci uvedených detektorů. Segmentace polotovarů napomohla lepší určení rotace a polohy, jelikož se blízko umístěné polotovary méně ovlivnily. Trénink modelů byl taktéž ovlivněn velikostí datasetu. Větší přesnosti by bylo možné dosáhnout vytvořením větší

datové množiny. Proces ale vyžaduje značnou časovou dotaci, jelikož je nutné polotovary připravit, nasnímat a na závěr oštítkovat. Pro větší robustnost aplikace by při tvorbě datasetu mohla být použita různá pozadí a osvětlení. Práce je také omezena dostupností hardwaru, na kterém probíhá trénování modelu. S využitím většího výpočetního výkonu by mohl být natrénován větší model, který by přinesl zlepšení predikce a značné zrychlení.

K celému projektu bylo přistupováno jako k pilotní verzi, tedy MVP. Vždy nebylo voleno nejlepší řešení, ale dostatečné. Například výběr osvětlení by napomohl přesnějším výsledkům detekce polotovarů, použití telecentrického objektivu by vylepšilo navádění robotického ramene. Uvedené jsou jen dva příklady ze série rozhodnutí, které vedly ke snížení výsledků. Nezbytné je konstatovat, že přes tato rozhodnutí bylo dosaženo vysoké přesnosti detekce a výsledky dokázaly překonat očekávání. Další kroky, kterými by bylo možné navázat na páci jsou uvedené v následujícím textu.

7.1 Možnosti navázání na práci

První možností je vylepšení volby hardwaru. Záměna kamery od společnosti *Basler AG* za chytrou kameru, výměna osvětlení, instalace telecentrického objektivu. Realizace se značně odvíjí od dostupnosti komponent, a proto by následná práce mohla být spojena s instalací kamery na robotické rameni, které je součástí buňky. Přesunutím kamery na manipulátor je možné dosáhnout zvětšení snímané plochy, také snímaní polotovarů z různým míst. V případě nejasné detekce může být změněna poloha kamery vůči polotovaru a určení může být bližší. V potaz je nutné vzít, že polotovary budou pomocí manipulátoru upínány do obráběcího stroje a instalace musí být přizpůsobena zástavbě obráběcího centra.

Další možností navázání na práci je vytvoření funkčního bloku v prostředí *TIA Portal*. Vysvětlení je takové, že PLC programování může být realizováno v různých programovacích jazycích, velmi rozšířený přístup je pomocí grafického programovacího jazyka ve funkčních blocích. Implementace v blocích je výhodná, jelikož program je dobře čitelný pro techniky, elektrikáře, případně údržbáře. V navazující práci by mohl být vytvořen funkční blok *EDUset ONE Vision*, který by sloužil jako knihovní prvek pro řízení z PLC. Spojení programovatelného automatu a vlastního softwaru v jazyku *Python* by se tak usnadnilo a zpřehlednilo. Následné navádění manipulačního robotu z PLC by pak bylo více přehledné a intuitivní.

Třetím příkladem, je rozšíření práce o vhodnou vizualizaci. Provedení by mohlo být realizováno například na panelu člověk-stroj, který je přímo připojený k PLC. Znamenalo by to přizpůsobit vytvořený funkční blok a rozšířit sdílené proměnné, které jsou komunikovány prostřednictvím OPC UA. Další provedení by mohlo být prostřednictvím vytvořeného API. Rozhraní by bylo nutné upravit pro výchozí možnosti, dále by bylo nezbytné vyvinout *frontend* aplikaci. Výhodou je, že by mohl být program strojového vidění konfigurován nejen z panelu člověk-stroj. Nicméně tento přístup přináší určité bezpečnostní riziko, které by muselo být eliminováno vhodnou autorizací.

Obecně lze uvést, že práce v oblasti strojového vidění v robotické buňce otevírá řadu možností pro další rozvoj a aplikace. Jedním z možných směrů je rozšíření aplikace o další funkcionality či zdokonalení stávajících procesů v robotické buňce na základě zjištěných výsledků. To může zahrnovat vylepšení detekce objektů, optimalizaci trasy robotické manipulace nebo implementaci nových technik manipulace s objekty. Dále je možné rozšířit práci o integraci dalších senzorů nebo zařízení, které mohou zlepšit schopnosti robotické buňky, a tím zvýšit úroveň autonomie a spolehlivosti robotických operací. Optimalizace procesů v robotické buňce na základě analýzy dat získaných pomocí strojového vidění představuje další perspektivní směr. V konečném důsledku, s velkými ambicemi, může tato práce posloužit jako základ pro další zkoumání nových technologií v oblastech robotiky a strojového vidění, což by mohlo vést k inovacím a pokrokům v automatizaci a průmyslových robotických systémech.

8 ZÁVĚR

V práci byla zkoumána a analyzována aplikace strojového vidění. Hlavní důraz byl kladen na provázání s populárním tématem – Průmysl 4.0. V úvodu práce byly nastíněny základní principy a technologie spojené s průmyslovým viděním a Průmyslem 4.0. Dále byly prozkoumány konkrétní algoritmy a metody používané ve strojovém vidění, s důrazem na přechod od klasických metod k moderním technikám založeným na hlubokém učení. Zmíněná část práce obsáhla analýzu reprezentace obrázků v počítači, detekci hran, extrakci příznaků a další důležité aspekty nezbytné k tvorbě detekčního softwaru.

Poté byla zavedena aplikace strojového vidění do robotické buňky *EDUset ONE*. Implementace byla zaměřena na vhodnou volbu hardwarových komponent ve výrobním prostředí, včetně výběru senzorů, umístění kamer a strategií pro efektivní strojové vidění. Byly důkladně prozkoumány různé faktory ovlivňující úspěšnost detekce objektů v reálném prostředí průmyslového zařízení. Nutné je zmínit, že velmi často nebylo vybráno nejlepší řešení, ale to dostatečné. Uvedená skutečnost otevírá dveře dalším příležitostem k vylepšení implementace.

Další fází diplomové práce bylo provedení srovnání předních detekčních frameworků využívajících neuronové sítě, jako jsou SSD, R-CNN a YOLO. Byla provedena detailní analýza výkonnosti těchto frameworků a jejich vhodnosti pro konkrétní aplikaci v průmyslovém prostředí. Na základě srovnávacích kritérií byl vybrán vhodný detekční framework, jenž se stal základním kamenem pro vlastní detekční software.

Na závěr bylo provedeno zhodnocení dosažených výsledků a bylo diskutováno o možnostech dalšího vývoje detekčního softwaru v průmyslovém prostředí. Byla navržena také možná zlepšení v implementaci a využití těchto technologií, aby byla otevřena cesta pro další pokrok v oblasti implementace strojového vidění do robotické buňky *EDUset ONE*. Tímto bylo přispěno k lepšímu porozumění této důležité oblasti a byly posíleny znalosti a možnosti využití v praxi.

SEZNAM POUŽITÉ LITERATURY

[1] SMITH, Melvyn L.; SMITH, Lyndon N. a HANSEN, Mark F. The quiet revolution in machine vision - a state-of-the-art survey paper, including historical review, perspectives, and future directions. Online. *Computers in industry*. 2021, roč. 130, s. 103472. ISSN 0166-3615. Dostupné z: https://doi.org/10.1016/j.compind.2021.103472. [cit. 2024-03-23].

[2] GOLNABI, H. a ASADPOUR, A. Design and application of industrial machine vision systems. Online. *Robotics and computer-integrated manufacturing*. 2007, roč. 23, č. 6, s. 630-637. ISSN 0736-5845. Dostupné z: https://doi.org/10.1016/j.rcim.2007.02.005. [cit. 2024-04-03].

[3] JAVAID, Mohd; HALEEM, Abid; SINGH, Ravi Pratap; RAB, Shanay a SUMAN, Rajiv. Exploring impact and features of machine vision for progressive industry 4.0 culture. Online. *Sensors international*. 2022, roč. 3, s. 100132. ISSN 2666-3511. Dostupné z: https://doi.org/10.1016/j.sintl.2021.100132. [cit. 2024-03-23].

[4] *Průmysl 4.0*. Online. In: Texas Instruments. 2020. Dostupné z: https://www.ti.com/applications/industrial/industry-4-0.html. [cit. 2024-03-23].

[5] DRAHOS, Peter; KUCERA, Erik; HAFFNER, Oto a KLIMO, Ivan. Trends in industrial communication and OPC UA. Online. 2018, s. 1-5. ISBN 978-1-5386-4419-5. Dostupné z: https://doi.org/10.1109/CYBERI.2018.8337560. [cit. 2024-03-23].

[6] SIMON, Jean-François. *OPC UA: Could it be the holy grail of your Digital Transformation?* Online. In: Ewon. 2020. Dostupné z: https://www.ewon.biz/de/all-resources/ewon-iiot-blog/opc-ua-could-it-be-the-holy-grail-of-your-digital-transformation. [cit. 2024-03-23].

[7] VOJÁČEK, Antonín. Průmyslová komunikace OPC UA - 1.díl - popis protokolu. Online. *Automatizace.hw.cz*. 2020, s. 1. Dostupné z: https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html. [cit. 2022-03-16].

[8] ELGENDY, Mohamed. *Deep Learning for Vision Systems*. 1. United States of America: MANNING SHELTER ISLAND, 2020. ISBN 9781617296192.

[9] *Ama Dablam Summit push*. Online. In: Namas Adventure. 2021. Dostupné z: https://www.namasadventure.com/7000m-8000mexpeditions-blog-posts/experiences-and-skills-required-to-climb-ama-dablam-expedition. [cit. 2024-03-23].

[10] DAVIES, E. R. (E. Roy). *Computer and machine vision: theory, algorithms, practicalities*. 4th ed. Amsterdam: Academic Press, 2012. ISBN 978-0-12-386908-1.

[11] *Bagr*. Online. In: Veacom. 2023. Dostupné z: https://www.veacom.cz/cs/blog/znacky-bagru-21. [cit. 2024-03-23].

[12] TORRE, Vincent a POGGIO, Tomaso A. On Edge Detection. Online. *IEEE transactions on pattern analysis and machine intelligence*. 1986, roč. PAMI-8, č. 2, s. 147-163. ISSN 0162-8828. Dostupné z: https://doi.org/10.1109/TPAMI.1986.4767769. [cit. 2024-03-23].

[13] SUN, Rui; LEI, Tao; CHEN, Qi; WANG, Zexuan; DU, Xiaogang et al. Survey of Image Edge Detection. Online. *Frontiers in signal processing (Lausanne)*. 2022, roč. 2. ISSN 2673-8198. Dostupné z: https://doi.org/10.3389/frsip.2022.826967. [cit. 2024-03-23].

[14] DE LA TORRE, F. a BLACK, M.J. Robust principal component analysis for computer vision. Online. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. IEEE, 2001, 362-369 vol.1. ISBN 0769511430. Dostupné z: https://doi.org/10.1109/ICCV.2001.937541. [cit. 2024-03-23].

[15] SEBERA, Martin. *Statistika - vícerozměrné metody*. Brno: Masarykova univerzita, 2014. ISBN 978-80-210-6692-2.

[16] LENDAVE, Vijaysinh. *Detecting Orientation of Objects in Image using PCA and OpenCV*. Online. 1. 2021. Dostupné také z: https://analyticsindiamag.com/detecting-orientation-of-objects-in-image-using-pca-and-opencv/.

[17] *Dokumentace OpenCV: Introduction to Principal Component Analysis (PCA)*. Online. 4.5.3. Dostupné také z: https://docs.opencv.org/4.5.3/d1/dee/tutorial_introduction_to_pca.html.

[18] *TheAILearner: OpenCV Minimum Area Rectangle*. Online. 2024. Dostupné z: https://theailearner.com/tag/cv2-minarearect/. [cit. 2024-03-23].

[19] *Dokumentace OpenCV: Contour Features*. Online. 4.9.0. Dostupné také z: https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html.

[20] MCMULLEN, P. CONVEX SETS AND THEIR APPLICATIONS. Online. *Bulletin of the London Mathematical Society*. 1983, roč. 15, č. 3, s. 270-271. ISSN 0024-6093. Dostupné z: https://doi.org/10.1112/blms/15.3.270. [cit. 2024-03-23].

[21] *Buying Skis for Beginners*. Online. In: Skatepro. 2020. Dostupné z: https://www.skatepro.com/en-us/a806.htm. [cit. 2024-03-23].

[22] SHAMOS, MICHAEL IAN. *Computational Geometry*. 1978. ISBN 108264837X.

[23] *OpenCV: minAreaRect*. Online. 4.9.0. Dostupné také z: https://github.com/opencv/opencv/blob/master/modules/imgproc/src/rotcalipers.cpp.

[24] O'SHEA, Keiron a NASH, Ryan. An Introduction to Convolutional Neural Networks. Online. *ArXiv.org*. 2015. Dostupné z: https://doi.org/10.48550/arxiv.1511.08458. [cit. 2024-03-23].

[25] 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. 2024-03-23].

[26] MUKHLIF, Abdulrahman Abbas; AL-KHATEEB, Belal a MOHAMMED, Mazin Abed. Incorporating a Novel Dual Transfer Learning Approach for Medical Images. Online. *Sensors (Basel, Switzerland)*. 2023, roč. 23, č. 2, s. 570. ISSN 1424-8220. Dostupné z: https://doi.org/10.3390/s23020570. [cit. 2024-03-23].

[27] POTRIMBA, Petru. *What is R-CNN?* Roboflow Blog, 2023. Dostupné také z: https://blog.roboflow.com/what-is-r-cnn/.

[28] SOLAWETZ, Jacob. *What is YOLOv8? The Ultimate Guide*. Roboflow Blog, 2023. Dostupné také z: https://blog.roboflow.com/whats-new-in-yolov8/.

[29] TAN, Lu; HUANGFU, Tianran; WU, Liyao a CHEN, Wenying. Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. Online. *BMC medical informatics and decision making*. 2021, roč. 21, č. 1, s. 324-324. ISSN 1472-6947. Dostupné z: https://doi.org/10.1186/s12911-021-01691-8. [cit. 2024-04-06].

[30] LI, Min; ZHANG, Zhijie; LEI, Liping; WANG, Xiaofan a GUO, Xudong. Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster R-CNN, YOLO v3 and SSD. Online. *Sensors (Basel, Switzerland)*. 2020, roč. 20, č. 17, s. 1-14. ISSN 1424-8220. Dostupné z: https://doi.org/10.3390/s20174938. [cit. 2024-03-24].

[31] *B&R Chytrá kamera*. Online. Br-automation.com. 2024. Dostupné z: https://www.br-automation.com/en/products/vision-systems/smart-camera/. [cit. 2024-04-03].

[32] *Aca2440-20gc*. Online. Baslerweb.com. 2024. Dostupné z: https://www.baslerweb.com/en/shop/aca2440-20gc/. [cit. 2024-04-03].

[33] *Aca2500-20gc*. Online. Baslerweb.com. 2024. Dostupné z: https://www.baslerweb.com/en/shop/aca2500-20gc/. [cit. 2024-04-03].

[34] *Aca2500-14gm*. Online. Baslerweb.com. 2024. Dostupné z: https://www.baslerweb.com/en/shop/aca2500-14gm/. [cit. 2024-04-03].

[35] *C125-0418-5m-p*. Online. Baslerweb.com. 2024. Dostupné z: https://www.baslerweb.com/en/shop/basler-lens-c125-0418-5m-p-f4mm. [cit. 2024-04-03].

[36] *C10-1214-2m-s*. Online. Baslerweb.com. 2024. Dostupné z: https://www.baslerweb.com/en-us/shop/basler-lens-c10-1214-2m-s-f12mm. [cit. 2024-04-03].

[37] *C23-0824-5m-p*. Online. Baslerweb.com. 2024. Dostupné z: https://www.baslerweb.com/en/shop/basler-lens-c23-0824-5m-p-f8mm. [cit. 2024-04-03].

[38] *Kupolovitá světla*. Online. Smartview.cz. 2022. Dostupné z: https://www.smartview.cz/p-kupolovita-svetla-dl. [cit. 2024-04-03].

[39] *Výkonná plochá světla*. Online. Smartview.cz. 2022. Dostupné z: https://www.smartview.cz/p-vykonna-plocha-svetla. [cit. 2024-04-03].

[40] *Kruhová světla*. Online. Smartview.cz. 2022. Dostupné z: https://www.smartview.cz/p-kruhova-svetla. [cit. 2024-04-03].

[41] ŠKRABÁNEK, Pavel. *Strojové vidění*. Lekce. Fakulta strojního inženýrství, Ústav automatizace a informatiky, 2023.

[42] *SIMATIC S7-1500F*. Online. Siemens.com. C1996-2024. Dostupné z: https://www.siemens.com/cz/cs/products/automation/systems/industrial/plc/simatic-s7-1500.html. [cit. 2024-04-06].

[43] DEEPA, R; TAMILSELVAN, E; ABRAR, E.S. a SAMPATH, Shrinivas. Comparison of Yolo, SSD, Faster RCNN for Real Time Tennis Ball Tracking for Action Decision Networks. Online. *2019 International Conference on Advances in Computing and Communication Engineering (ICACCE)*. 2019, s. 1-4. ISBN 978-1-7281-3250-1. Dostupné z: https://doi.org/10.1109/ICACCE46606.2019.9079965. [cit. 2024-03-24].

[44] KIM, Jeong-ah; SUNG, Ju-Yeong a PARK, Se-ho. Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. Online. *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*. 2020, s. 1-4. ISBN 978-1-7281-6164-8. Dostupné z: https://doi.org/10.1109/ICCE-Asia49877.2020.9277040. [cit. 2024-04-06].

SEZNAM ZKRATEK A SYMBOLŮ

API	Application Programming Interface
CASENet	Convolutional Aggregation of Superpixels for Semantic Segmentation
CEDN	Context Encoding for Semantic Segmentation
CNN	Convolutional Neural Network
CMOS	Complementary Metal-Oxide-Semiconductor
FPS	Frames Per Second
IoT	Internet of Things
KLT	Karhunenova-Loèvova transformace
mAP	Mean Average Precision
MVP	Minimum Viable Product
NMS	Non-Maximum Suppression
OPC UA	OPC Unified Architecture
PCA	Principal Component Analysis
PLC	Programmable Logic Controller
ResNet	Residual Neural Network
R-CNN	Region-based Convolutional Neural Network
RINDNet	Recurrent Inference of Discontinuous Entities Network
SVM	Support Vector Machine
TIA	Totally Integrated Automation
VGG16	Visual Geometry Group 16
VGGNet	Visual Geometry Group Network
YOLO	You Only Look Once

SEZNAM OBRÁZKŮ

Obr. 1: Průmysl 4.0 v provedení moderního výrobního závodu 18
Obr. 2: Průmyslová standardizace OPC UA v malém podniku..... 19
Obr. 3: Výstup na Ama Dablam 21
Obr. 4: Lyžař v horách při sjezdu 21
Obr. 5: Obrázek v počítači zobrazen jako 3D matice s kanály RGB..... 23
Obr. 6: Detekované hrany na obrázky lyžaře pomocí Sobelovy masky 24
Obr. 7: Architektura konvoluční sítě CEDN využívaná pro detekci hran 25
Obr. 8: Extrakce rysů z obrázku bagru 26
Obr. 9: Vizualizace mraku dat v 2D grafu, určení os dat pomocí PCA..... 27
Obr. 10: Lyžař ve dvou rámečcích, červený box má minimální plochu 28
Obr. 11: Konvexní obálka okolo bodů, které objímá elastická gumička..... 29
Obr. 12: Architektura konvoluční neuronové sítě rozdělená podle vrstev 32
Obr. 13: Techniky augmentace obrazu vytváří nová vstupní data pro konvoluční síť... 33
Obr. 14: Transfer learning se zachovanými konvolučními vrstvami a novým klasifik.. 34
Obr. 15: Architektura vyspělejší sítě Fast R-CNN 36
Obr. 16: Architektura frameworku Single Shot Detector 38
Obr. 17: Architektura sítí YOLO..... 39
Obr. 18: Porovnání kamery firmy *Basler AG* a chytré kamery *B&R* 44
Obr. 19: Konstrukce pro aplikaci strojového vidění v buňce *EDUset ONE*..... 46
Obr. 20: Zvolené tvary polotovarů pro aplikaci strojového vidění..... 48
Obr. 21: Hledání vhodné kombinace světla a pozadí 51
Obr. 22: Anotace v prostředí *Roboflow* 55
Obr. 23: Rozmístění polotovarů v datasetu 55
Obr. 24: Knihovny a frameworky využité v jednotlivých modulech aplikace..... 56
Obr. 25: Srovnání vybraných modelů na základě ztrátové funkce při trénování 59
Obr. 26: Výkonost konvolučních neuronových sítí při detekci polotovarů..... 60
Obr. 27: Architektura vlastního softwaru pro detekci polotovarů 62
Obr. 28: Datová struktura pro komunikaci prostřednictvím OPC UA 63
Obr. 29: Proces vývoje aplikace strojového vidění v robotické buňce *EDUset ONE*.... 65

SEZNAM TABULEK

Tab. 1: Výsledky frameworků po trénování na datasetu medikamentů 40
Tab. 2: Výsledky frameworků pro detekci skleníku ze satelitních snímků 40
Tab. 3: Srovnání dostupných sestav (kamer + objektiv) od firmy *Basler AG* 49
Tab. 4: Porovnání vybraných vlastností detekčních frameworků 61

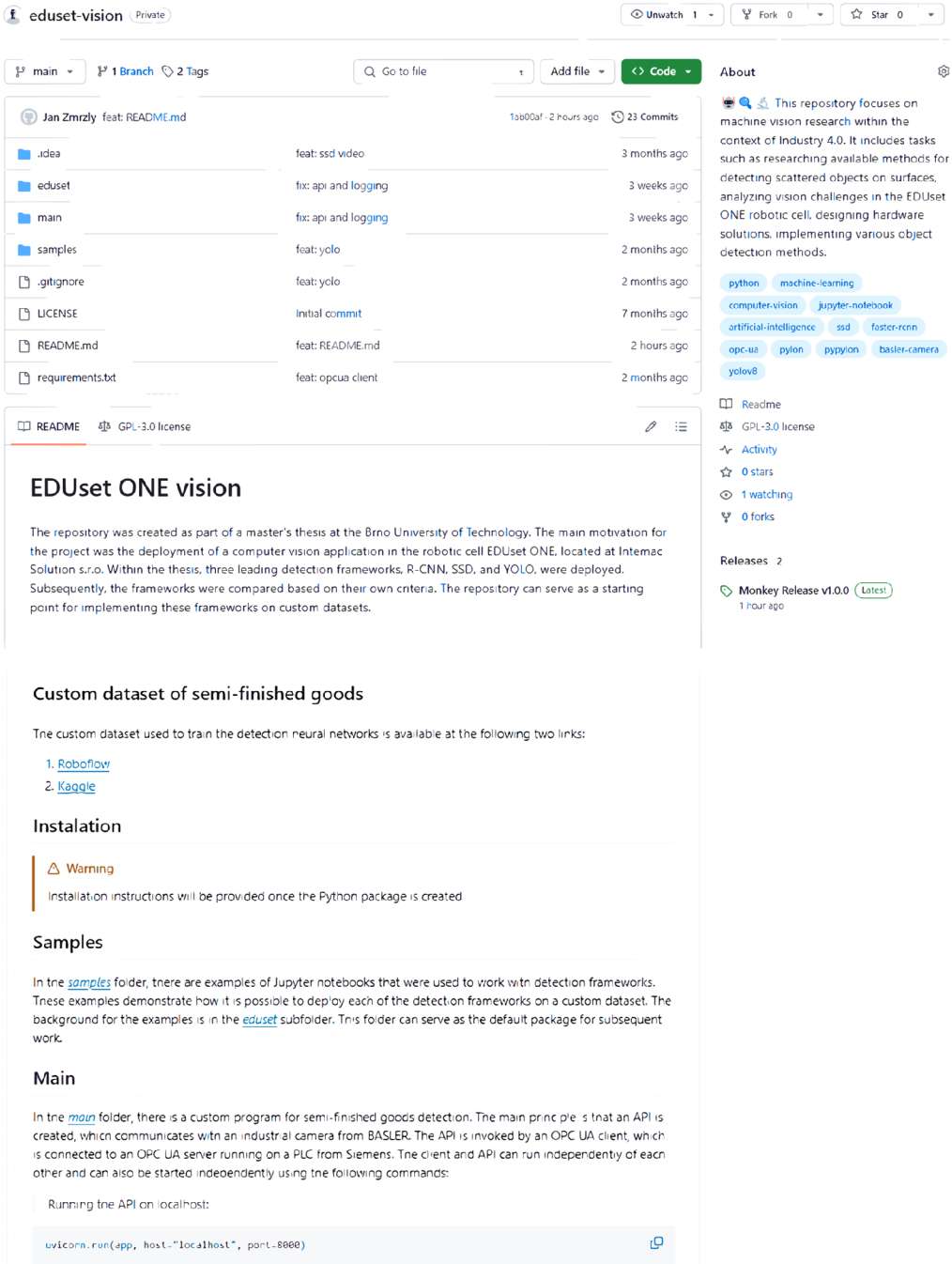
SEZNAM PŘÍLOH

PŘÍLOHA A **Struktura repositáře *eduset-vision* na platformě *Github***
PŘÍLOHA B **Ukázka komunikace detekčního softwaru a PLC**
PŘÍLOHA C **Vizualizace vlastního detekčního softwaru**

Elektronické přílohy:
Export repositáře *eduset-vision Monkey v1.0.0*

PŘÍLOHY

A Struktura repositáře *eduset-vision* na platformě *Github*



Running the OPC UA client:

```
url = "opc.tcp://192.168.0.10:4840"

api_endpoint = "http://localhost:8000/placement"

control_signal = {"ns": "http://EdusetONE", "i": 75}
placement = {"ns": "http://EdusetONE", "i": 19}

client = OPCUAClient(url=url)

try:
    await client.set_encryption() if client.encryption is True else None
    await client.run(control_signal, placement, line=2, api_endpoint=api_endpoint)
except KeyboardInterrupt:
    await client.disconnect()
```

License:

License: GPLv3

The project is licensed under the GNU General Public License version 3 (GPLv3)

Contact:

For inquiries and support, please contact zmrzlyjan@gmail.com for more information.

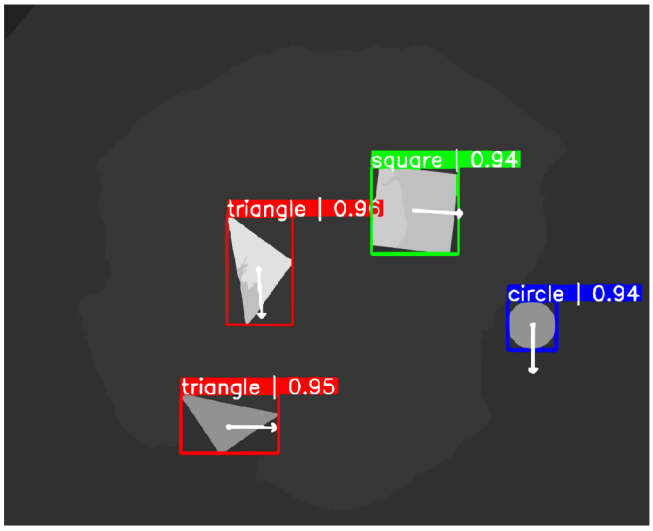
Obr. 1: Repositář *eduset-vision* dostupný na platformě *Github*[16]

16 Odkaz na projekt *eduset-vision*: https://github.com/JanZmrzly/eduset-vision

B Ukázka komunikace detekčního softwaru a PLC

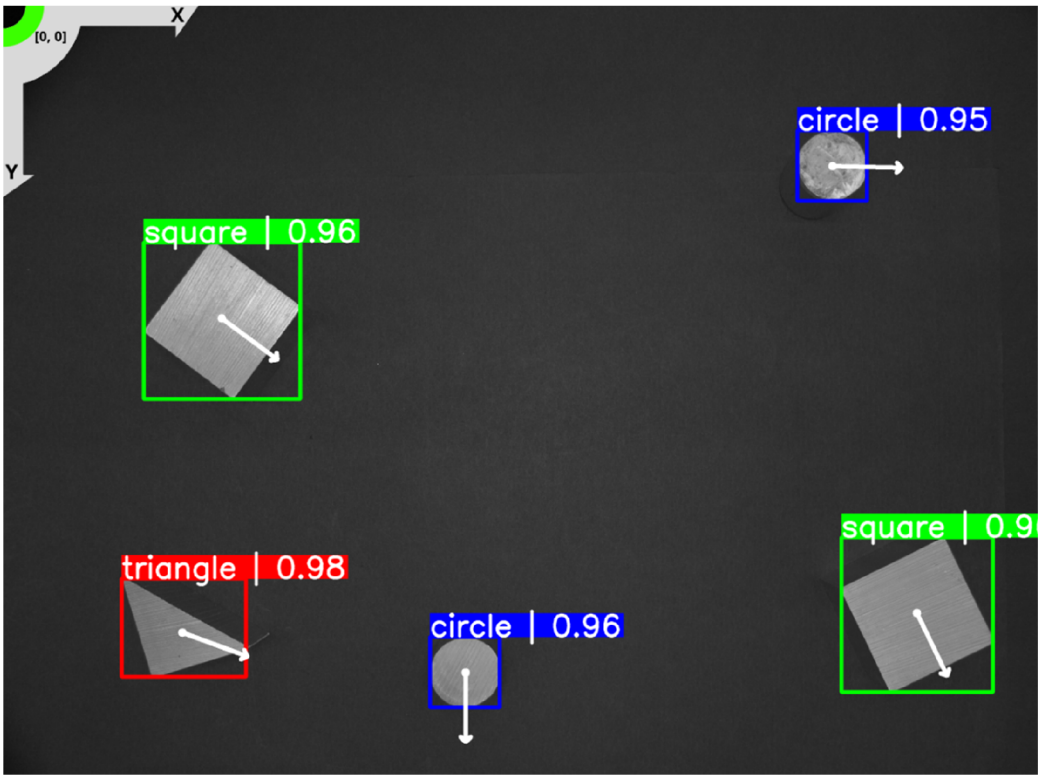
OPCData

	Name	Data type	Start value	Monitor value
16	Placement	Array[0..10] of "Ed...		
17	Placement[0]	"EdusetOneVision"		
18	name	String	''	'triangle'
19	x	UInt	0	1052
20	y	UInt	0	959
21	angle	Real	0.0	1.493902
22	Placement[1]	"EdusetOneVision"		
23	name	String	''	'triangle'
24	x	UInt	0	949
25	y	UInt	0	1496
26	angle	Real	0.0	0.01342439
27	Placement[2]	"EdusetOneVision"		
28	name	String	''	'circle'
29	x	UInt	0	1990
30	y	UInt	0	1148
31	angle	Real	0.0	1.570796
32	Placement[3]	"EdusetOneVision"		
33	name	String	''	'square'
34	x	UInt	0	1585
35	y	UInt	0	757
36	angle	Real	0.0	0.06854341



Obr. 2: Ukázka komunikace detekčního softwaru prostřednictvím protokolu OPC UA

C Vizualizace vlastního detekčního softwaru



Obr. 3: Vizualizace vlastního detekčního softwaru dostupná přes API