

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

**VÝVOJ SYSTÉMU DETEKCE VAD NA OSAZENÝCH PCB
S VYUŽITÍM KONVOLUČNÍCH NEURONOVÝCH SÍTÍ**

Bc. Kamil Špicar

Diplomová práce

2024

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Kamil Špicar**
Osobní číslo: **I22199**
Studijní program: **N0714A150005 Automatické řízení**
Téma práce: **Vývoj systému detekce vad na osazených PCB s využitím konvolučních neuronových sítí**
Zadávající katedra: **Katedra řízení procesů**

Zásady pro vypracování

Cílem práce je vývoj systému detekce vad na osazených deskách plošných spojů (PCB) pomocí umělých neuronových sítí. V rámci teoretické části práce se student zaměří na problematiku detekce vad na osazených PCB a význam automatizovaného řešení tohoto problému. Klíčovým bodem teoretické části bude řešení architektury konvolučních neuronových sítí vhodných pro řešení detekce vad. Praktická část bude zahrnovat identifikaci a roztřídění defektů z obrazových dat. Student navrhne vhodné metody detekce různých typů defektů s využitím konvolučních neuronových sítí a vytvoří software pro trénink a testování těchto modelů s využitím existujících platform. Dále bude provedeno hodnocení vzniklých modelů na testovacích obrazových datech. Závěrem praktické části bude návrh metod pro dodatečné zpracování výsledků s cílem detekce a kategorizace vad.

Rozsah pracovní zprávy: **cca 60 stran**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

RUSS, John C. *The image processing handbook*. 6th ed. Boca Raton: CRC Press, c2011. ISBN 978-1-4398-4045-0
GONZALEZ, Rafael C. a Richard E. WOODS. *Digital image processing*. Fourth edition. New York: Pearson, [2018]. ISBN 978-013-3356-724.

Vedoucí diplomové práce: **Ing. Dominik Štursa**
Katedra řízení procesů

Datum zadání diplomové práce: **8. listopadu 2023**
Termín odevzdání diplomové práce: **17. května 2024**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Daniel Honc, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 14. listopadu 2023

Prohlášení

Prohlašuji:

Práci s názvem Vývoj systému detekce vad na osazených PCB s využitím konvolučních neuronových sítí jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 17. 5. 2024

Bc. Kamil Špicar v. r.

Poděkování

Děkuji své rodině a přítelkyni za podporu nejen při tvorbě této diplomové práce, ale i v průběhu celého mého studia. Dále děkuji Ing. Adamovi Šolcovi z firmy Vitesco Technologies za zprostředkování tématu této práce a cenné rady.

V Pardubicích dne 17. 5. 2024

Bc. Kamil Špicar v. r.

ANOTACE

Cílem práce je návrh systému detekce vad na osazených deskách plošných spojů s využitím konvolučních neuronových sítí. Navržený systém je komplexní množinou detekčních metod zaměřených na vady identifikované z historických obrazových dat, jejichž kombinací lze klasifikovat celé inspekční fotografie osazených desek plošných spojů. Detekční metody zahrnují ověřené architektury konvolučních neuronových sítí a výpočty parametrů pro dílčí klasifikace z jejich výstupů. Byl vytvořen software pro testování celého navrženého systému a trénování jednotlivých modelů.

KLÍČOVÁ SLOVA

AOI, PCB, konvoluční neuronové sítě, vizuální inspekce, detekce vad

TITLE

DEVELOPMENT OF AN ASSEMBLED PCB DEFECT DETECTION SYSTEM USING CONVOLUTIONAL NEURAL NETWORKS

ANNOTATION

This thesis aims to design a defect detection system for assembled printed circuit boards using convolutional neural networks. The proposed system is a complex set of detection methods that focus on defects identified from historical image data, the combination of which can be used to classify entire inspection photos of assembled printed circuit boards. The detection methods include verified convolutional neural network architectures and parameter calculations for partial classifications based on their outputs. Software was developed to test the entire proposed system and to train individual models.

KEYWORDS

AOI, PCB, Convolutional neural networks, Visual inspection, Defect detection

OBSAH

Seznam zkratk a značek	12
Seznam ilustrací	13
Seznam tabulek	15
Úvod	16
1 Vizualní inspekce osazených pcb	17
1.1 Vady osazených pcb	17
1.1.1 Vady plošných spojů	17
1.1.2 Vady pájení	17
1.1.3 Vady komponent	18
1.1.4 Kontaminace	18
1.2 Vady bondování	18
1.3 Klíčové vlastnosti pořizovaného obrazu	19
1.3.1 Osvětlení	20
1.3.2 Rozlišení a detail	21
1.3.3 Barva a kontrast	21
2 Architektury CNN	24
2.1 Faster R-CNN	24
2.1.1 ResNet	25
2.1.2 FPN	28
2.1.3 RPN	29
2.1.4 RoI Align	31
2.1.5 Finální vrstvy Faster R-CNN	32
2.2 Mask R-CNN	34
2.2.1 Finální vrstvy Mask R-CNN	34
2.2.2 PointRend	37
2.3 EfficientNet	42
2.3.2 MBConv blok	44
2.3.3 Finální vrstvy architektury EfficientNet	45
3 Definice detekční úlohy	47
3.1 Identifikace a kategorizace defektů	47
3.1.1 Deformace bondovaných drátků	48
3.1.2 Chybějící komponenty	48

3.1.3	Chyba pozice nebo natočení komponent	49
3.1.4	Nadměrné množství nebo absence gelu na ASIC	49
3.1.5	Přítomnost gelu na komponentech	50
3.1.6	Absence gelu na portu	51
3.1.7	Kontaminace vlákny	51
3.1.8	ECA kontaminace	52
3.1.9	Nadměrné množství a kontaminace silikonem	52
3.1.10	Kontaminace částicemi	53
3.1.11	Shrnutí identifikovaných kategorií defektů	53
4	Návrh detekčního systému	55
4.1	Modely	57
4.1.1	Locator	57
4.1.2	Wires	59
4.1.3	Port	60
4.1.4	Anomalies	61
4.1.5	Tvorba datasetů	62
4.2	Navržené metody měření	63
4.2.1	Měření pozice komponent	63
4.2.2	Měření rotace komponent	64
4.2.3	Měření tvaru komponent	66
4.2.4	Měření gelu na ASIC	66
5	Software pro trénování a testování systému	67
5.1	Trénování modelů	68
5.2	Testování navrženého systému	68
6	Hodnocení modelů	71
6.1	Locator	71
6.1.1	Test nalezení všech objektů	74
6.2	Wires	75
6.2.1	Test přesnosti klasifikace	77
6.3	Port	78
6.3.1	Test přesnosti klasifikace	80
6.4	Anomalies	81
6.4.1	Test nalezení všech objektů	82
7	Závěr	84

Použitá literatura	86
Přílohy	89

SEZNAM ZKRATEK A ZNAČEK

AOI	automatická optická inspekce
ASIC	integrovaný obvod pro specifické aplikace
CNN	konvoluční neuronová síť
ECA	elektricky vodivé lepidlo
FPN	příznaková pyramidální síť
MBCConv	mobilní invertovaný zúžený konvoluční (blok)
PCB	deska plošných spojů
R-CNN	oblastní konvoluční neuronová síť
ReLU	rektifikovaná lineární jednotka (aktivační funkce)
RoI	oblast zájmu
RPN	síť navrhuující oblasti
ResNet	residuální síť
SE	stlačení a excitace (vrstva)
SiLU	swish aktivační funkce

SEZNAM ILUSTRACÍ

Obr. 1.1 – Zkrat mezi sousedními piny (PCBONLINE, 2021)	17
Obr. 1.2 – Posun komponentu (DigiSource, 2020).....	18
Obr. 1.3 – Ilustrace vad bondování.....	19
Obr. 1.4 – Rozdíl mezi přímým a difuzním osvětlením PCB (Advanced illumination, 2021)	20
Obr. 1.5 – Rozdíl mezi zeleným a modrým osvětlením PCB (Advanced illumination, 2021)	21
Obr. 2.1 – Blokové schéma Faster R-CNN ResNet FPN architektury	24
Obr. 2.2 – Struktura residuálního bloku ResNet50 dle Detectron2 implementace.....	26
Obr. 2.3 – Blokové schéma ResNet50 dle Detectron2 implementace.....	27
Obr. 2.4 – Schéma procesu FPN.....	28
Obr. 2.5 – Blokové schéma RPN dle Detectron2 implementace.....	30
Obr. 2.6 – Blokové schéma modulu RoI Align dle Detectron2 implementace	31
Obr. 2.7 – Blokové schéma finálních vrstev dle Detectron2 implementace.....	33
Obr. 2.8 – Blokové schéma procesu finálních vrstev Mask R-CNN při tréninku	35
Obr. 2.9 – Blokové schéma procesu finálních vrstev Mask R-CNN při inferenci	36
Obr. 2.10 – Vliv PointRend modulu na přesnost segmentace Mask R-CNN (Kirillov, 2020).37	
Obr. 2.11 – Blokové schéma procesu predikce hrubé masky modulem PointRend.....	38
Obr. 2.12 – Blokové schéma modulu PointRend při tréninku.....	39
Obr. 2.13 – Blokové schéma finálních vrstev Mask R-CNN PointRend při inferenci.....	40
Obr. 2.14 – Proces zpřesnění segmentační masky modulem PointRend při inferenci	41
Obr. 2.15 – Blokové schéma obecné EfficientNet architektury	43
Obr. 2.16 – Blokové schéma EfficientNet Stem bloku dle pycls implementace.....	43
Obr. 2.17 – Blokové schéma EfficientNet MBCConv bloku dle pycls implementace	45
Obr. 2.18 – Finální vrstvy EfficientNet architektury dle pycls implementace	46
Obr. 3.1 – Popis ukázkové inspekční fotografie.....	47
Obr. 3.2 – Ukázky různých typů defektů bondovaných drátků	48
Obr. 3.3 – Ukázky inspekčních fotografií s chybějícími komponenty	49
Obr. 3.4 – Ukázky inspekčních fotografií s komponenty mimo pozici a špatným natočením	49
Obr. 3.6 – Ukázky vzhledů nadměrného množství ochranného gelu na ASIC	50
Obr. 3.5 – Ukázky vzhledů absence ochranného gelu na ASIC	50
Obr. 3.7 – Ukázky ochranného gelu na komponentech.....	50
Obr. 3.8 – Ukázka inspekční fotografie s chybějícím gelem na portu.....	51
Obr. 3.9 – Ukázky inspekčních fotografií s kontaminací vláknou	51

Obr. 3.10 – Ukázky ECA kontaminací	52
Obr. 3.11 – Ukázky nadměrného množství a kontaminace silikonem	52
Obr. 3.12 – Ukázky inspekčních fotografií se silnou kontaminací částicemi.....	53
Obr. 4.1 – Blokové schéma návaznosti modelů navrhovaného detekčního systému	57
Obr. 4.2 – Vypočtené osy souřadnicového systému PCB	64
Obr. 4.3 – Vypočtený úhel natočení PCB.....	65
Obr. 5.1 – Stromové schéma repozitáře vytvořeného softwaru.....	67
Obr. 5.2 – Kód pro konfiguraci a spuštění celého systému	69
Obr. 5.3 – Vizualizace výsledků vytvořeným softwarem.....	70
Obr. 6.1 – Průběh klasifikační ztráty modelu Locator během tréninku.....	72
Obr. 6.3 – Průběh segmentační ztráty PointRend modelu Locator během tréninku	73
Obr. 6.2 – Průběh segmentační ztráty Mask R-CNN modelu Locator během tréninku	73
Obr. 6.4 – Vizualizace výstupu modelu Locator	74
Obr. 6.5 – Průběh přesnosti klasifikace objektů modelu Wires během tréninku.....	76
Obr. 6.6 – Vizualizace výstupu modelu Wires na vstupní fotografii	78
Obr. 6.7 – Průběh přesnosti klasifikace modelu Port během tréninku	79
Obr. 6.8 – Průběh přesnosti klasifikace objektů modelu Anomalies během tréninku.....	82

SEZNAM TABULEK

Tab. 3.1 – Identifikované kategorie defektů orientačně seřazené dle priority	54
Tab. 4.1 – Přehled rozřazení kategorií defektů do skupin dle podobnosti.....	56
Tab. 4.2 – Třídy modelu Locator	58
Tab. 4.3 – Třídy modelu Wires	60
Tab. 4.4 – Třídy modelu Port.....	61
Tab. 4.5 – Třídy modelu Anomalies	62
Tab. 6.1 – Základní konfigurované parametry modelu Locator	71
Tab. 6.2 – Výsledek testu nalezení všech objektů pro model Locator	75
Tab. 6.5 – Základní konfigurované parametry modelu Wires	76
Tab. 6.6 – Matice záměn z výsledků testu přesnosti klasifikace modelu Wires.....	77
Tab. 6.7 – Procentuální přesnosti klasifikace jednotlivých tříd modelu Wires	77
Tab. 6.8 – Základní konfigurované parametry modelu Port.....	79
Tab. 6.9 – Matice záměn z výsledků testu přesnosti klasifikace modelu Port	80
Tab. 6.10 – Procentuální přesnosti klasifikace jednotlivých tříd modelu Port.....	80
Tab. 6.11 – Základní konfigurované parametry modelu Anomalies	81
Tab. 6.12 – Výsledek testu nalezení všech objektů pro model Anomalies	82

ÚVOD

V oblasti průmyslové výroby osazených desek plošných spojů (PCB) je vizuální inspekce kritickým krokem zajišťujícím jakost finálních produktů. S rostoucí komplexitou vyráběných PCB roste i složitost jejich automatické vizuální inspekce, která tak může překročit možnosti tradičních metod a musí být prováděna lidskými operátory. Kontrola je tedy ve výsledku časově náročná a náchylná k chybám, což vede k potřebě vyvinout pokročilé automatizované systémy schopné v této roli člověka nahradit. Velký potenciál mají v tomto případě metody umělé inteligence, především umělé neuronové sítě, které v posledních letech zaznamenaly rapidní vývoj.

Tato diplomová práce je tvořena ve spolupráci s firmou Vitesco Technologies v Trutnově, která poskytuje prostředky pro realizaci praktické části. Předpokládaným výsledkem práce je návrh systému automatické vizuální inspekce využívající konvoluční neuronové sítě, ze kterého by měla vycházet následná implementace přímo ve výrobě. Navržena bude sekvence kontrolních kroků vedoucí k finální klasifikaci jakosti vyrobeného produktu. Součástí návrhu bude ukázková aplikace takového systému vycházející z obrazových dat poskytnutých firmou Vitesco Technologies.

1 VIZUÁLNÍ INSPEKCE OSAZENÝCH PCB

Vizuální inspekce je zásadním krokem procesu kontroly jakosti osazených desek plošných spojů (PCB), jelikož mnoho kritických vad, které mohou v průběhu výrobního procesu vzniknout, má vizuální charakter.

1.1 VADY OSAZENÝCH PCB

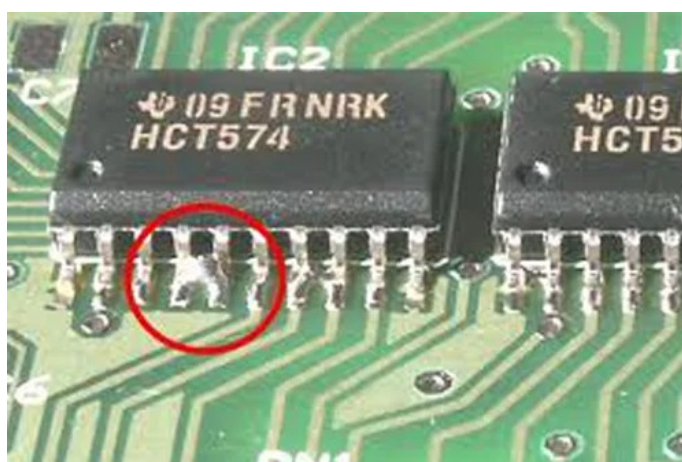
Vady PCB, které mají vizuální charakter a jsou tudíž detekovatelné vizuální inspekcí lze dále rozdělit na vady plošných spojů, vady pájení, vady komponent a kontaminace.

1.1.1 Vady plošných spojů

Obecně může jít o narušení nebo propojení vodivých stop na povrchu desky. Narušení vodivé stopy může vzniknout v důsledku prasklin, škrábanců či jiných mechanických poškození. K propojení dochází vlivem cizího vodivého materiálu, který se mohl na desku dostat například při pájení komponent (MorePCB, 2023).

1.1.2 Vady pájení

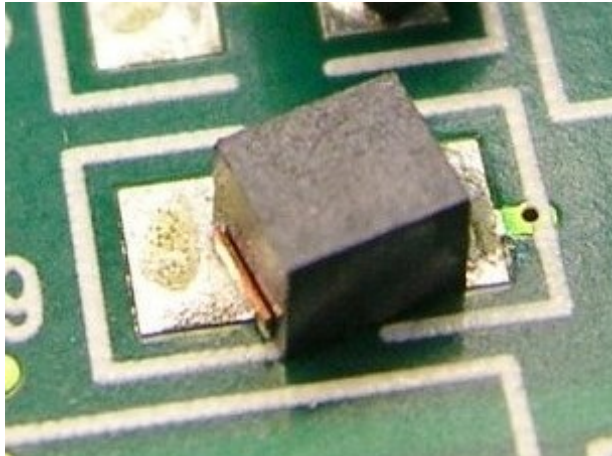
Vady pájení lze obecně rozdělit na nedostatečné a nadměrné pájení. Nedostatečné pájení může zanechat komponenty připojené pouze částečně a nadměrné pájení může vést ke zkratům mezi sousedními piny nebo stopami (PCBONLINE, 2021).



Obr. 1.1 – Zkrat mezi sousedními piny (PCBONLINE, 2021)

1.1.3 Vady komponent

Komponenty mohou být během osazování umístěny na špatnou pozici, mít špatnou orientaci nebo chybět úplně. Ke změně jejich pozice může dojít i v průběhu tuhnutí pájky nebo vodivého lepidla (DigiSource, 2020). Dále může během výroby docházet i k jejich mechanickému poškození, jehož výsledkem mohou být praskliny, odlomky nebo deformace (MorePCB, 2023).



Obr. 1.2 – Posun komponentu (DigiSource, 2020)

1.1.4 Kontaminace

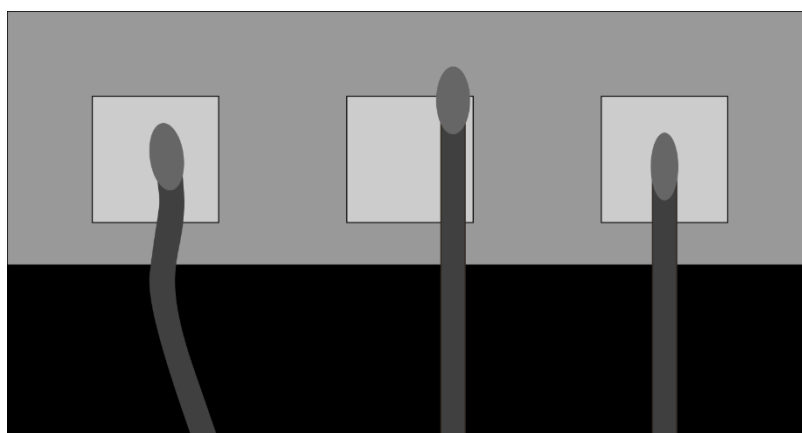
Během výrobního procesu se může na PCB dostat prach, vlákna, otisky prstů, nebo jiné cizí materiály, což může vést k elektrickým i mechanickým problémům finálního produktu (MorePCB, 2023).

1.2 VADY BONDOVÁNÍ

Bondování (Wire Bonding) je efektivní propojovací metoda, která se používá převážně při výrobě polovodičových součástek pro připojení čipu k pinům na jeho obalu pomocí jemných drátků. Využívá se ale i pro přímé propojení integrovaného obvodu nebo PCB s jiným elektronickým prvkem, jako je například čidlo (Valenta, 2015). Proto může být i součástí vizuální inspekce osazených PCB. Existuje více bondovacích metod, které se liší způsobem vytvoření bondu a tím pádem i jeho tvarem, což ovlivňuje i tvar zakřivení drátků. Mezi hlavní metody patří kuličkové a klínové bondování. Volba vhodné bondovací metody závisí na konkrétní aplikaci. Pro účely propojení drobného PCB s jiným elektronickým prvkem je obvykle lepší variantou klínové bondování, protože vyžaduje menší prostor a bondovací plošky

tak mohou být blíže k sobě (Green, 2017). Právě tato bondovací metoda byla využita při výrobě produktu, pro který je v rámci této práce navrhován systém detekce vad.

Vady, které mohou nastat během procesu klínového bondování, jsou v převážné většině odhalitelné vizuální inspekcí. Mezi tyto vady obecně patří deformace drátků, špatné umístění bondu a nedostatečný spoj v místě bondu. Na tvar a umístění bondu se obvykle zavádí limity, které se odvíjí od materiálu drátků a bondovací plošky. V závislosti na konkrétní aplikaci se ale rozsah těchto limitů může lišit. Zmíněné kategorie vad spolu úzce souvisí, tedy pokud je například tvar drátku viditelně deformovaný, dá se předpokládat, že síla, která deformaci způsobila, narušila také bond. Ilustrace kategorií vad klínového bondování je na obr. 1.3.



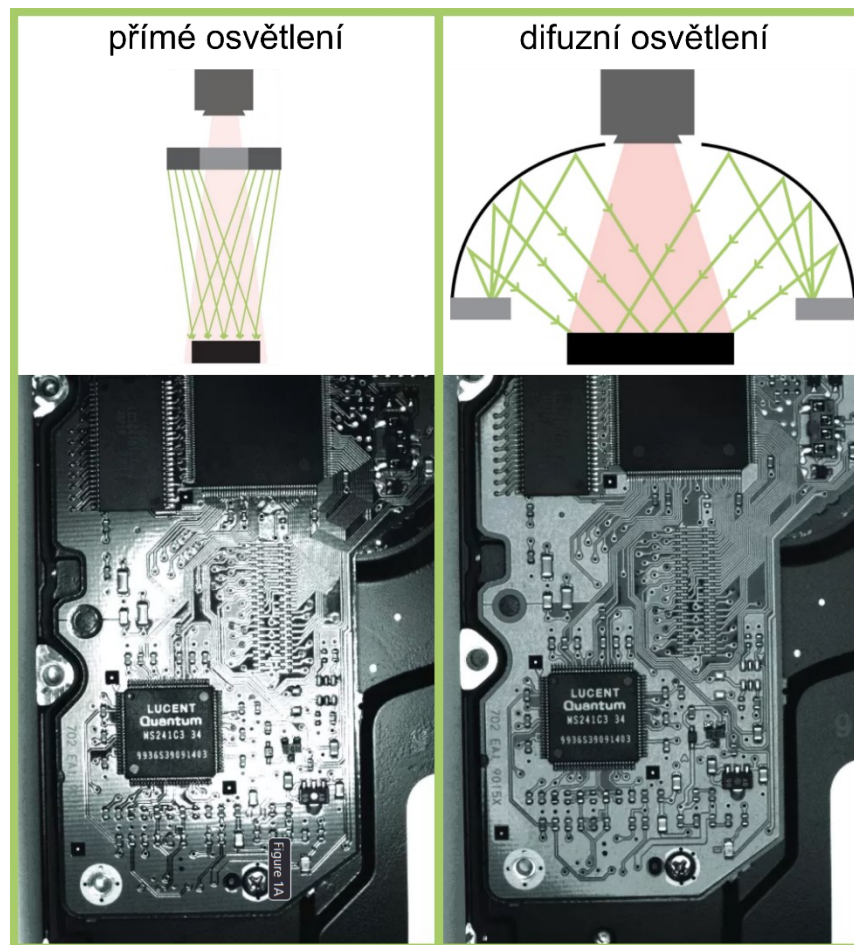
Obr. 1.3 – Ilustrace vad bondování

1.3 KLÍČOVÉ VLASTNOSTI POŘIZOVANÉHO OBRAZU

V současné době se ještě stále můžeme setkat s manuální vizuální inspekcí prováděnou přímo lidským operátorem bez použití dalších pomůcek usnadňujících inspekci, snad kromě mikroskopu (MorePCB, 2023). Tento přístup k vizuální inspekci ovšem neumožňuje sběr dat o vyrobených produktech a v případě reklamace může být obtížné zpětně odhalit přesný důvod selhání konkrétního produktu. Proto se do vizuální inspekce zavádějí kamery a obrazová data všech vyrobených produktů jsou ukládána do databáze. Operátor pak provádí inspekci nepřímou na pořízených obrazových datech. Výhodou tohoto přístupu je možnost zpracování obrazu tak, aby byly zvýrazněny relevantní detaily a oblasti zájmu, což může vést ke zvýšení úspěšnosti lidských operátorů. Při pořizování obrazu pro vizuální inspekci je nutné jeho odladění na základě vizuálních charakteristik a míst výskytu všech předpokládaných vad, které je nutné detekovat.

1.3.1 Osvětlení

Správné osvětlení je základem pro získání kvalitního obrazu osazeného PCB. Čistě pro kontrolu přítomnosti a správné pozice všech komponent je vhodné využití difuzního osvětlení. Difuzní osvětlení osvětluje PCB rovnoměrně a pomáhá minimalizovat stíny a odlesky, které by mohly zakrývat důležité detaily, protože se jeho paprsky odražené od PCB neodráží přímo do kamery. Pro kontrolu pájených spojů může být naopak přínosnější směrové osvětlení, protože vyzdvihuje výškové rozdíly a kombinací více snímků je tak možné do jisté míry odhadnout i prostorové rozměry. Je-li směrové osvětlení dostatečně stabilní, dá se využít k detekci změny natočení osvětlovaných objektů. V případě silně odrazivých povrchů lze využít polarizační filtry. Intenzita osvětlení a úhly směrového osvětlení se ladí na základě vlastností konkrétní scény (Advanced illumination, 2021). Pro automatickou vizuální inspekci je velmi důležitá stabilita osvětlení, protože výraznější změny v osvětlení mohou přesáhnout její schopnosti generalizace. Na obr. 1.4 je znázorněn rozdíl mezi přímým a difuzním osvětlením PCB s příkladem základního umístění kamery a světelného kroužku pro oba případy.



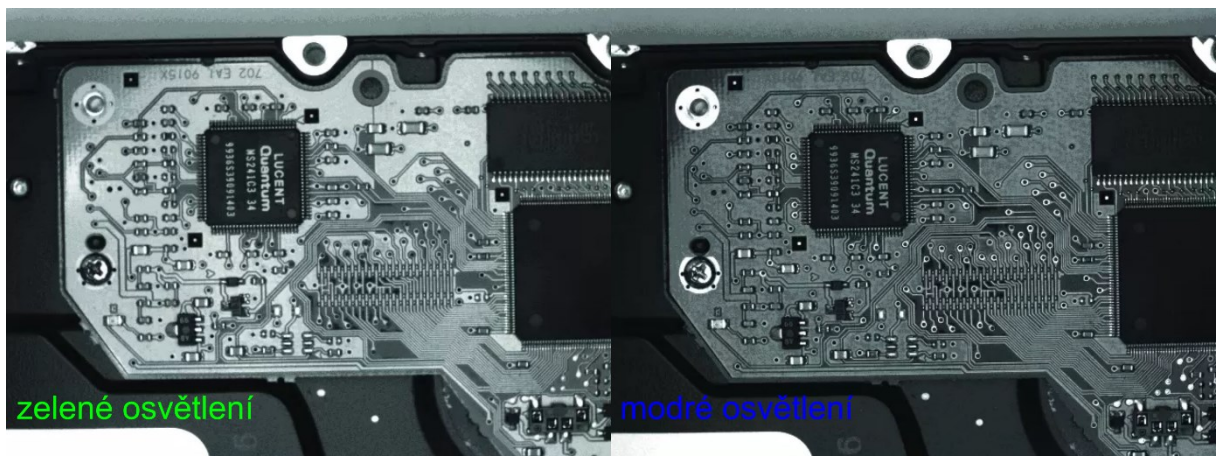
Obr. 1.4 – Rozdíl mezi přímým a difuzním osvětlením PCB (Advanced illumination, 2021)

1.3.2 Rozlišení a detail

Rozlišení kamery by mělo být tak vysoké, aby bylo možné rozlišit i ty nejmenší kontrolované detaily. Aby tedy kamera byla schopna zachytit všechny relevantní informace. V případě kontrol na výrobních linkách schopných vyrábět vícero provedení PCB, se může vyplatit pořízení kamery s lehce vyšším rozlišením, pro případ, že se objeví nové provedení s menšími prvky. Pořizovaný obraz se obvykle ukládá do databáze, přičemž jeho ztrátová komprese není vždy žádoucí, protože by mohlo dojít ke ztrátě informací, které sice člověk nevnímá, ale algoritmy AOI ano. V takovém případě je nutné vzít v úvahu i velikost pořizovaného obrazu, z čehož může vyplynout určitá horní hranice rozlišení kamery. Dostatečné rozlišení nepomůže, není-li kamera zaostřena na relevantní detaily. Přestože bývají produkty během výroby pevně uchyceny, nelze zabránit jejich jemnému pohybu v rámci jejich uchycení způsobeného vibracemi a nárazy. Je tedy vhodné zavést systém automatického ostření, který je buď součástí kamery nebo externího objektivu, nebo jiným způsobem zajistit skutečně pevné uchycení kontrolovaného produktu (JAI, 2021).

1.3.3 Barva a kontrast

Nastavení barvy a kontrastu je důležité pro zvýraznění rozdílů mezi různými materiály a komponenty na desce. Pořizovaný obraz ovšem nemusí být nutně barevný. Černobílý obraz umožňuje zavedení barevného osvětlení scény. Různé barvy osvětlení mají různou odrazivost od různých materiálů. Této interakce je možné využít k dalšímu zvýraznění relevantních prvků v obraze a k utlumení těch nepodstatných. Také lze pořídit obrazy pod různými barevnými konfiguracemi osvětlení, kdy každé osvětlení zvýrazňuje jiné prvky, a následně je zkombinovat do jednoho (Advanced illumination, 2021).



Obr. 1.5 – Rozdíl mezi zeleným a modrým osvětlením PCB (Advanced illumination, 2021)

1.4 LIDSKÝ ZRAK A VIZUÁLNÍ INSPEKCE

Ze všech lidských smyslů je pro člověka zrak největším zdrojem informací o okolním světě. Parametry lidského oka přitom nejsou v porovnání s ostatními tvory na Zemi nijak zvlášť výjimečné. Surové informace získané z fotonů dopadajících na sítnici, jsou v porovnání s tím, jaký zorný úhel člověk vnímá, že vidí, omezené. Jak je tedy možné, že toho „vidíme“ tolik? Člověk si vyvinul schopnost obraz zpracovávat předtím, než se dostane do vědomí. Lidské oko je schopné rychle zmapovat poměrně široký zorný úhel, ačkoliv skutečně detailní a zaostřené je pouze místo, kam se dívá. Dělá tak skládáním jednotlivých obrazů, které byly postupně nasbírány během mapování okolí a které si člověk podvědomě pamatuje. V této fázi je obraz modifikován na základě evolučních aspektů ale i předchozích zkušeností. Člověk ve svém zorném poli tedy reálně vnímá jen omezené množství objektů. Zbytek si jen myslí, že vnímá, protože mu to podvědomý systém zpracovávající signál ze zrakového ústrojí řekl (Russ, 2016). Pokud je ale velká část obrazu, který nakonec vidíme, výsledkem tohoto procesu, můžeme se na svůj zrak spolehnout při úloze vizuální inspekce?

Vizuální inspekce je v podstatě detekcí určitých objektů. Některé objekty vyhledáváme, protože se v obraze musí nacházet a jiné, protože se v něm nacházet nesmí. Zrak člověka je silně provázaný s jeho myslí. Když se člověk dívá kolem sebe, podvědomě všemu, co vnímá jako objekt, přiřazuje určitý štítek a kategorii, se kterými dále pracuje, když o těchto objektech přemýšlí. Celý obraz tak bývá podvědomě rozdělen na objekty a pozadí. To, jaký štítek a kategorii danému objektu přiřadí, závisí na předchozí zkušenosti s tím, že čím častěji podobný objekt viděl, tím snazší je jeho detekce. Pokud se člověk s nějakým objektem příliš často nasetkal, nebo vůbec nasetkal, je pro něj mnohem složitější ho zařadit a přiřadit mu štítek. To tedy znamená, že tento objekt má zvýšenou pravděpodobnost, že bude člověkem vnímán jako pozadí a nebude vědomě detekován. Tato vlastnost souvisí ještě s jedním aspektem lidského vnímání, kterým je tzv. tunelové vidění. Dochází k němu, když se člověk soustředí na nalezení nějaké konkrétní představy o objektu, nebo sleduje nějaký konkrétní objekt (Russ, 2016). Pozitivním důsledkem v kontextu vizuální inspekce je zvýšená pravděpodobnost nalezení objektů vyskytujících se na očekávaných místech, na která se člověk soustředí. Negativním důsledkem je snížená pravděpodobnost nalezení objektů s náhodnou pozicí. Člověk tak může přehlédnout i zdánlivě nepřehlédnutelné objekty (Kujawińska, 2015).

Zmíněné vlastnosti lidského vidění jsou problematické hlavně v situacích, kdy se člověk setkává s nestandardními objekty, se kterými se pravděpodobně v běžném životě nasetkal. Vzorovou situací tohoto typu je vizuální inspekce ve výrobě. Na druhou stranu je ale člověk

schopen se naučit tyto objekty detekovat, pokud o nich obdrží dostatečné informace. Z toho vyplývá, jak důležité je řádné zaučení operátorů vizuální inspekce. Nejedná se ale o univerzální řešení, protože rychlost, s jakou je člověk schopen se naučit detekovat všechny vady, je vysoce individuální. A čím složitější je kontrolovaný systém, tím méně lidí je schopno se naučit v něm detekovat vady v rozumném čase (Kujawińska, 2015; Russ, 2016).

Vizuální inspekce vyžaduje zvýšenou pozornost. Pozornost člověka při inspekci obecně závisí na jeho psychice a pracovním prostředí. Zatímco pracovní prostředí je možné upravit tak, aby mělo minimální negativní vliv na pozornost, psychickému rozpoložení operátora lze předejít jen omezenými prostředky. Pozornost člověka je závislá i na souladu času směn s jeho biorytmem. Při běžném směnovém provozu se člověk nemusí zvládnout plně přeorientovat na noční směny. Výsledkem je jeho snížená pozornost, a tudíž i zvýšený počet chyb při vizuální inspekci. V těchto zmíněných ohledech má automatická vizuální inspekce (AOI) jednoznačně navrch. Vývoj efektivního AOI pro složitější vizuální inspekce je ale velmi náročný, v některých případech téměř nemožný, a tudíž i nákladný. Proto jsou složitější vizuální inspekce často zastávány člověkem, i přes jeho zjevnou sníženou efektivitu (Kujawińska, 2015).

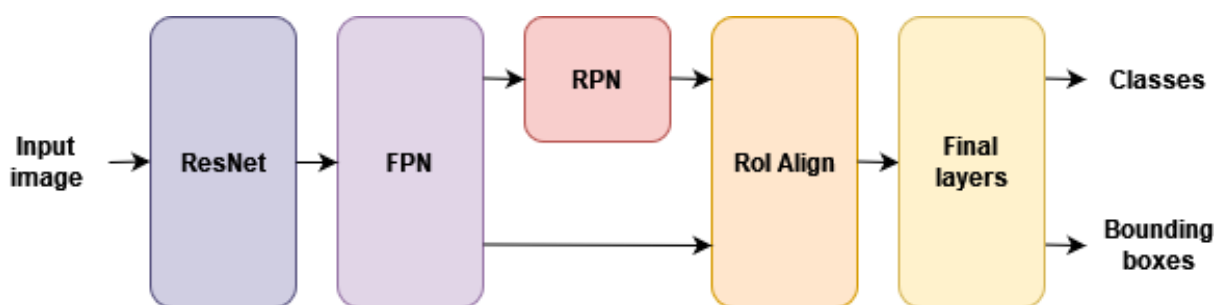
V čem je člověk prozatím nenahraditelný je schopnost adaptace na neočekávané změny. Například zatímco AOI může zvládnout přesně detekovat veškeré objekty a na základě jejich vyhodnocení určit, jestli je produkt v pořádku, neuvědomí si, že produkty se zdají být vadné jen kvůli náhlé změně nasvícení scény způsobené vlivy vnějšího prostředí, na které nebyl při jejím návrhu brán zřetel. V případě složitějších vizuálních inspekcí tedy může být vhodné využít AOI k usnadnění inspekce člověku, ale nikoliv k jeho úplnému nahrazení (IBE Electronics, 2022).

2 ARCHITEKTURY CNN

Jak již bylo zmíněno v kapitole 1.4, vizuální inspekce je ve své podstatě detekcí objektů a jejich klasifikace. Kromě nalezení určitých objektů může být výhodná i jejich segmentace, kterou lze využít k měření různých parametrů za účelem dodatečné klasifikace. V posledních letech bylo pro tyto účely vyvinuto velké množství architektur využívající konvoluční neuronové sítě (CNN). V následujících kapitolách budou popsány architektury, které byly zvoleny jako vhodné pro řešení dílčích úloh vizuální inspekce osazených PCB na základě literatury, volně dostupných benchmarků, dostupnosti jejich implementací a dostupnosti předtrénovaných vah. Detailnější zdůvodnění jejich volby bude uvedeno v kapitole 4.1.

2.1 FASTER R-CNN

Faster R-CNN je architektura pro detekci a následnou klasifikaci objektů v obraze. Navazuje na Fast R-CNN zrychlením procesu extrakce oblastí zájmu z mapy příznaků získané z páteřní sítě zavedením Region Proposal Network (RPN) (Ren, 2016). Od její prvotní publikace vznikla řada variant této architektury dále zlepšujících její efektivitu. V rámci této kapitoly bude popsána implementace Faster R-CNN z knihovny Detectron2, která na rozdíl od prvotní verze využívá páteřní síť ResNet v kombinaci s Feature Pyramid Network (FPN) (Wu, 2019). Dalším rozdílem je náhrada RoI Pooling modulu modulem RoI Align, který byl publikován později spolu s architekturou Mask R-CNN (He, 2018). Na obr. 2.1 je blokové schéma této verze Faster R-CNN.



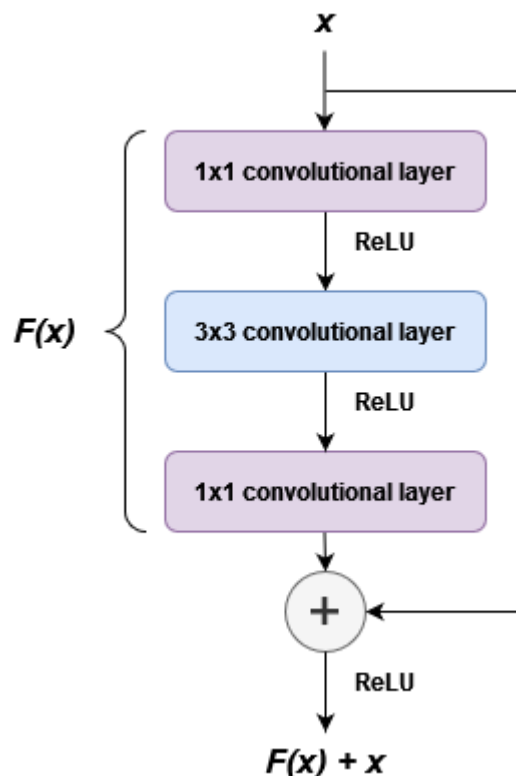
Obr. 2.1 – Blokové schéma Faster R-CNN ResNet FPN architektury

2.1.1 ResNet

ResNet, z anglického sousloví Residual Network, je typ hluboké neuronové sítě, který se obvykle používá jako páteřní síť složitějších architektur pro extrakci příznaků z obrazových dat. Výhodou této architektury je redukce problému mizejících gradientů, ke kterému dochází při zpětné propagaci chyby při trénování hluboké neuronové sítě. Toho dosahuje díky tzv. residuálnímu bloku, který je její základní stavební jednotkou. Tato architektura má několik variant, které se liší v počtu vrstev, tedy v hloubce sítě. Čím hlubší síť je, tím více informací je schopná se naučit a následně extrahovat z obrazových dat. Zároveň ale potřebuje větší a rozmanitější trénovací dataset. Protože na jednoduchém datasetu, který obsahuje v poměru s potenciálem sítě málo informací k naučení, může docházet k rychlému přeučení, a tedy neschopnosti generalizace na neznámých obrazových datech. V takovém případě je výhodnější varianta sítě s menší hloubkou. Hlubší varianty sítě navíc zabírají více paměti a jsou výpočetně pomalejší. Volba vhodné varianty sítě závisí hlavně na složitosti detekční úlohy. Ta v základu zahrnuje míru rozmanitosti obrazových dat a míru rozdílnosti mezi kategoriemi objektů. Dalším aspektem samozřejmě může být i výpočetní náročnost. Správně odhadnout složitost detekční úlohy bez předchozích zkušeností s podobnou úlohou je obtížné, proto je na místě experimentace s více variantami sítě. Obecně je ale vhodné začít od variant s menší hloubkou a v případě nedostatečných výsledků postupně přecházet k variantám s většími hloubkami. Pro konkrétní představu o struktuře sítě ResNet bude v rámci této kapitoly rozebrána varianta ResNet50, protože je ideálním startovním bodem pro řadu detekčních úloh, jelikož nabízí dobrý poměr mezi efektivitou a výpočetní náročností (He, 2015).

Základní stavební jednotkou sítě ResNet je tzv. residuální blok. Jeho struktura pro ResNet50 dle Detectron2 implementace je na obr. 2.2. Vstup x do bloku prochází přes trojici konvolučních vrstev. První vrstva provádí konvoluci s filtrem 1×1 , krokem 1 a definovaným počtem filtrů pro snížení počtu kanálů vstupu, což snižuje výpočetní náročnost následující vrstvy. Přičemž první blok v rámci každé fáze počítá tuto konvoluci s krokem 2 za účelem dvojnásobného snížení plošných rozměrů vstupní příznakové mapy a zároveň stejným poměrem navyšuje její počet kanálů, což je dáno počtem filtrů. Pro následující bloky fáze je konvoluce vypočtena s krokem 1 a nedochází tak k další plošné kompresi příznakové mapy. Druhá konvoluční vrstva s filtrem 3×3 je hlavním extraktorem příznaků ze vstupu. Třetí vrstva opět rozšiřuje počet kanálů na původní hodnotu, což v případě prvního bloku v rámci fáze znamená počet po kompresi (He, 2015; Wu, 2019).

Za každou konvoluční vrstvou se provádí dávková normalizace za účelem snížení variability vah a stabilizace jejich distribuce v rámci jedné dávky, což vede k rychlejšímu a stabilnějšímu tréninku. Efekt této normalizace je korigován dalšími naučenými parametry, protože nemusí být vždy pozitivní. Po dávkové normalizaci následují ReLU aktivace. Rozdíl nastává po třetí vrstvě, kde je před finální ReLU aktivací proveden součet transformovaného vstupu s původním vstupem, který v případě prvního bloku fáze také nejprve projde kompresí, aby byl součet možný. Právě toto přímé propojení vstupu x s výstupem $F(x)$ řeší problém degradace efektivity učení s rostoucí hloubkou sítě (He, 2015; Wu, 2019).

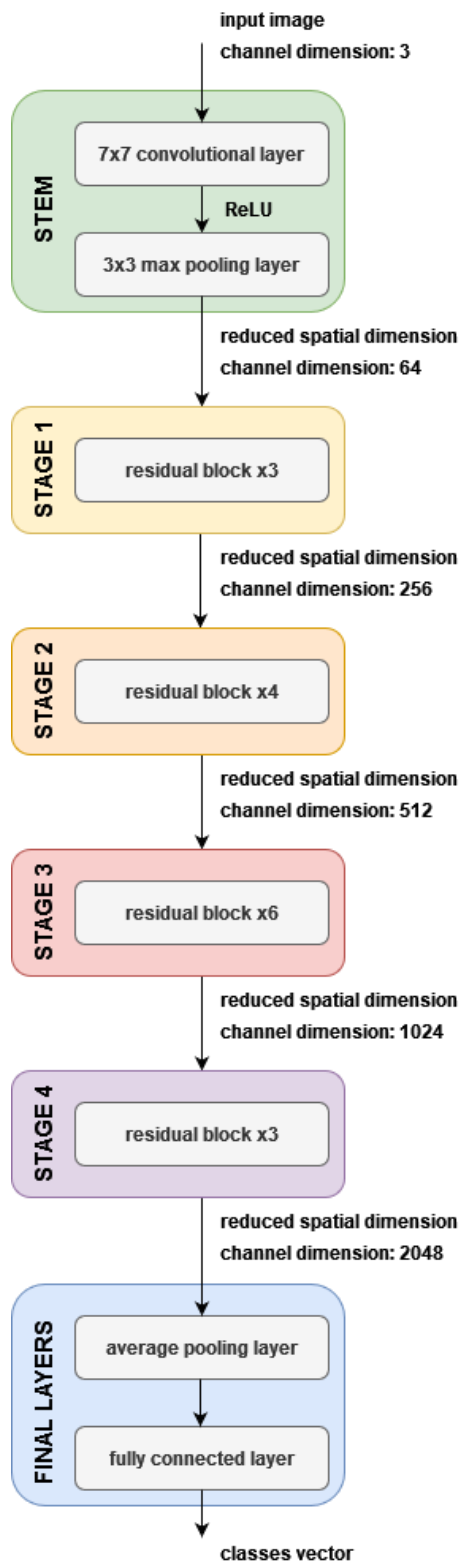


Obr. 2.2 – Struktura residuálního bloku ResNet50 dle Detectron2 implementace

Na obr. 2.3 je blokové schéma ResNet50 dle Detectron2 implementace. Tato architektura je tvořena počátečními vrstvami, čtyřmi fázemi extrakce příznaků a finálními vrstvami (Wu, 2019).

Počáteční vrstvy jsou obvykle sdružovány pod anglickým pojmem Stem, což v překladu znamená stonek. Jejich účelem je předběžná extrakce příznaků ze vstupních obrazových dat. Tím se vytvoří prvotní mapa příznaků, ze které je možné efektivně extrahovat další příznaky pomocí residuálních bloků. V případě ResNet50 jsou počáteční vrstvy tvořené konvoluční vrstvou s filtrem 7×7 , krokem 2 a 64 filtry následovanou dávkovou normalizací a 3×3 Max

Pooling vrstvou. Výstupem této skupiny vrstev je mapa příznaků, jejíž rozměry jsou zhruba čtyřikrát menší než rozměry vstupního obrazu a jejíž počet kanálů se zvýšil ze 3 na 64 (Wu, 2019).



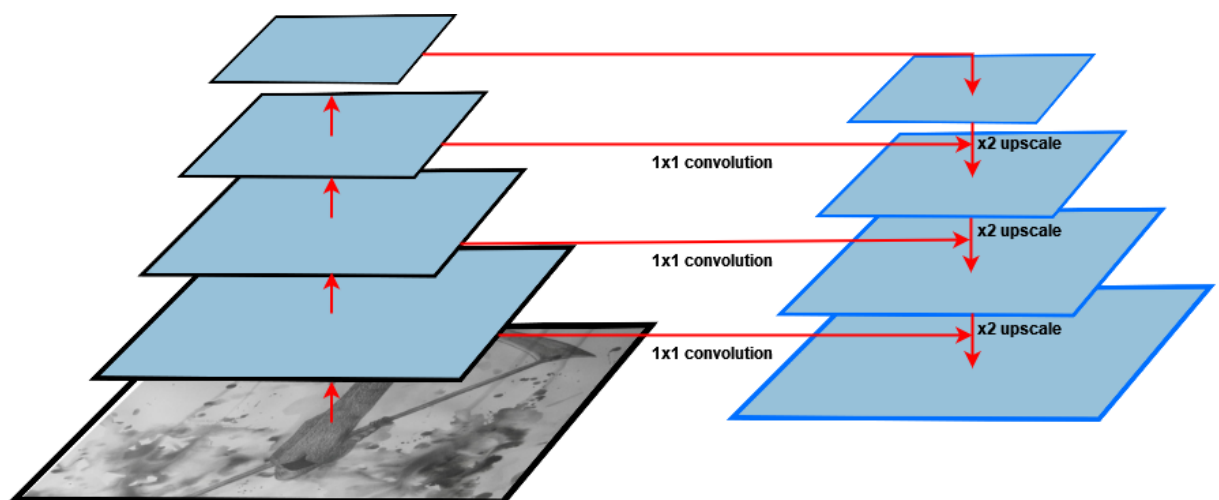
Obr. 2.3 – Blokové schéma ResNet50 dle Detectron2 implementace

Následuje čtveřice tzv. fází, které jsou tvořeny sériově propojenými residuálními bloky. Počty bloků v každé fázi závisí na variantě sítě ResNet. Pro ResNet50 je počet bloků na fázi: 3, 4, 6, 3, v tomto pořadí. Jak již bylo zmíněno, první residuální blok v rámci každé fáze, s výjimkou fáze první, dvakrát zmenšuje rozměry vstupní mapy příznaků a zvyšuje počet kanálů pomocí konvoluční vrstvy s filtrem 1×1 , krokem 2 a počtem filtrů dle fáze. Rozměry mapy příznaků se již dále v rámci jedné fáze nemění. Každá fáze tedy pracuje s jiným rozměrem mapy příznaků (Wu, 2019).

Výstupem poslední fáze je mapa příznaků určitého rozměru s 2048 kanály. Tato mapa poté putuje do Average Pooling vrstvy, která pro každý kanál vypočte jednu skalární hodnotu, čímž vznikne vektor o délce 2048. Tento vektor je vstupem do následující plně propojené vrstvy, která vrací vektor, o délce rovné počtu hledaných kategorií objektů, obsahující logity, tj. surová skóre jednotlivých kategorií. Na tyto logity se pak může aplikovat softmax funkce pro získání pravděpodobností výskytu jednotlivých kategorií v daném obraze. Zmíněné finální vrstvy se ale v případě Faster R-CNN ResNet50 FPN architektury nevyužívají, protože ResNet slouží pouze jako extraktor příznakových map, které jsou dále využívány v FPN (Wu, 2019).

2.1.2 FPN

Feature Pyramid Network (FPN) je architektura řešící problém detekce objektů různého měřítka. Dosahuje toho vytvořením pyramidy příznakových map různých rozměrů, které jsou následně předány dalšímu modulu nadřazené detekční architektury a detekce objektů pak probíhá na více úrovních (Lin, 2017). V případě Faster R-CNN je tímto modulem RPN, který bude popsán v následující kapitole 2.1.3 (Ren, 2016).



Obr. 2.4 – Schéma procesu FPN

Vstupem do FPN jsou celkem 4 příznakové mapy vytvořené páteří sítí ResNet v jednotlivých fázích. Tyto mapy se seřadí od nejmenší po největší a následně se kombinují dle schématu na obr. 2.4. Daná příznaková mapa vždy projde konvolucí s filtrem 1×1 , krokem 1 a počtem filtrů rovným počtu kanálů předchozí mapy. Předchozí mapa se zase dvakrát zvětší, aby měla stejné rozměry. Tím mají obě mapy stejný rozměr i počet kanálů a zkombinují se, což je obvykle realizováno prostým sečtením. Tím se kombinují sémanticky pokročilejší prvky mapy z hlubší úrovně s prostorovými detaily mapy mělčí úrovně. Výsledkem je příznaková mapa bohatá na sémantické i prostorové informace. Výstupem FPN jsou 4 takto zkombinované příznakové mapy (Lin, 2017; Wu, 2019).

2.1.3 RPN

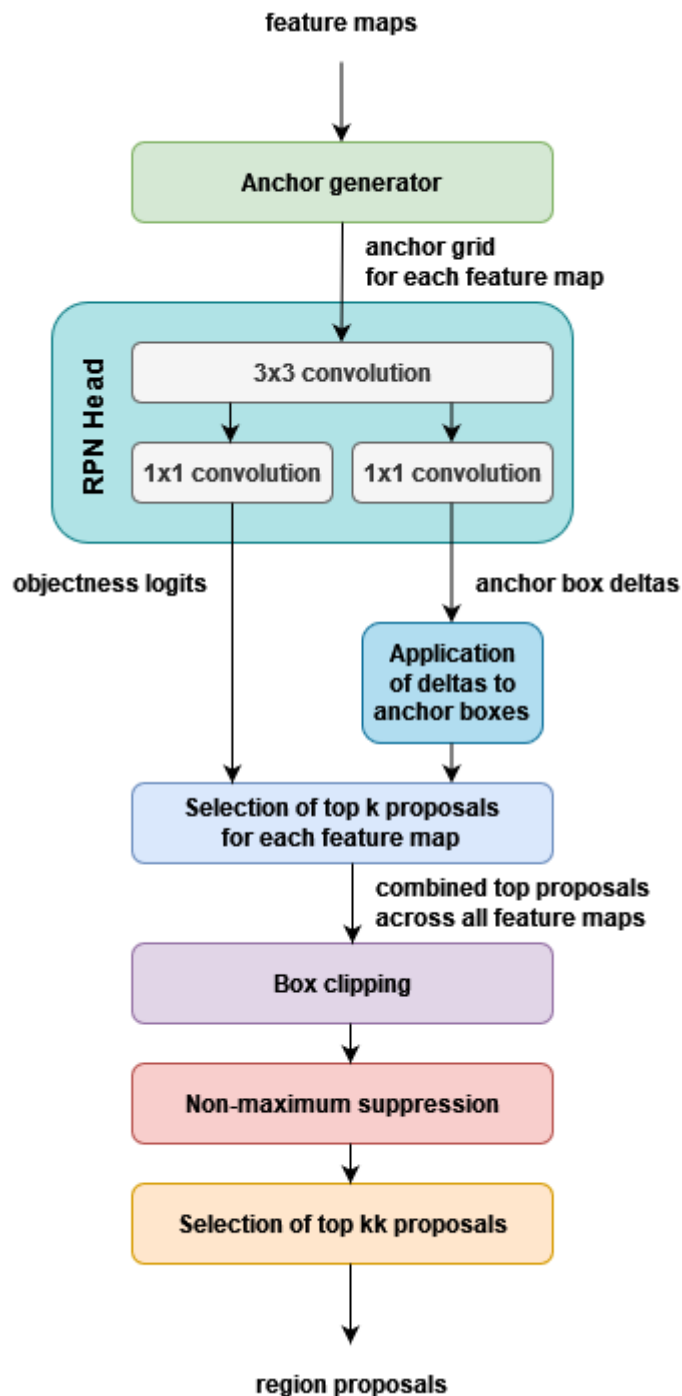
Region Proposal Network (RPN) je hlavním vylepšením architektury Faster R-CNN oproti předchozí verzi Fast R-CNN. Slouží k extrakci oblastí zájmu z mapy příznaků získané z páteří sítě (Ren, 2016). Blokované schéma tohoto modulu dle Detectron2 implementace je na obr. 2.5.

Vstupem do RPN jsou všechny příznakové mapy získané z páteří sítě. Anchor generátor pro každou mapu vypočte pozice bodů ležících na mřížce určitých rozměrů vycházejících z rozměrů mapy. V každém bodě pak vygeneruje tzv. anchory, což jsou obdélníkové oblasti se středem v tom bodě. Počet takto vygenerovaných anchorů vychází z konfigurace RPN. V konfiguraci se volí poměry stran a velikosti generovaných anchorů. Finální počet anchorů generovaných v každém bodě mřížky je roven počtu všech kombinací poměrů stran s velikostmi. Výstupem z anchor generátoru je tedy mřížka anchorů pro každou mapu příznaků (Ren, 2016; Wu, 2019).

RPN Head, jednoduchá konvoluční síť, následně pro každý anchor vypočte objectness logit, vyjadřující pravděpodobnost, že se v daném anchoru nachází objekt, a anchor box delty, což jsou navržené úpravy rozměrů anchoru, jejichž účelem je zajistit lepší ohraničení oblasti, ve které se nachází objekt. Tyto úpravy se aplikují a tím jsou získány oblasti zájmu (Ren, 2016; Wu, 2019).

Z každé mapy příznaků je vybrán definovaný počet oblastí zájmu s nejvyšším objectness logitem. Takto vybrané oblasti jsou pak sloučeny do jedné skupiny napříč všemi příznakovými mapami, přičemž informace o tom, z jaké mapy pochází, je zachována. Oblasti zájmu zasahující mimo mapu, jsou oříznuty (Wu, 2019).

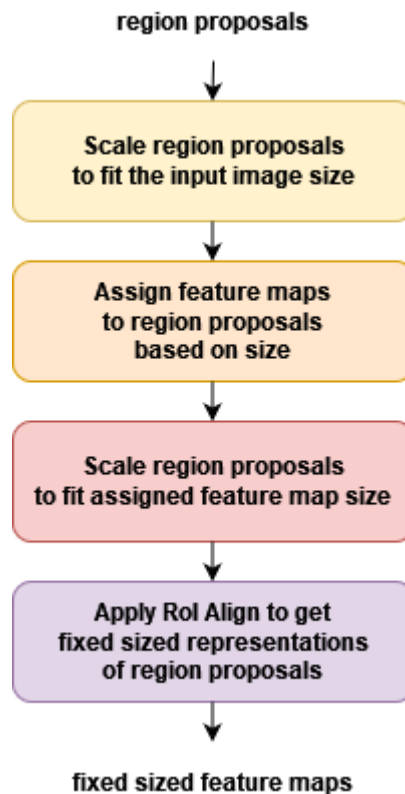
Následuje proces Non-maximum suppression, jehož účelem je eliminace oblastí zájmu překrývajících stejný objekt. V případě, že míra překryvu dvou oblastí překračuje určitý limit, je ponechána pouze oblast s vyšším objectness logitem. Tento proces je ale aplikován pouze na oblasti pocházející ze stejné mapy příznaků. Ze zbylých oblastí je opět vybrán definovaný počet oblastí s nejvyšším objectness logitem. Takto vybrané oblasti jsou výstupem z RPN (Wu, 2019).



Obr. 2.5 – Blokové schéma RPN dle Detectron2 implementace

2.1.4 RoI Align

Na základě oblastí zájmu z RPN je nutné z příznakových map z FPN vyříznout oblasti, které jsou vstupem do finálních vrstev architektury. Problémem je, že finální konvoluční vrstvy vyžadují vstupy jednotného rozměru, ale oblasti zájmu mají různé jak rozměry, tak souřadnicové systémy. Řešením je modul RoI Align, který sofistikovaným postupem, znázorněným na obr. 3.6, převádí oblasti zájmu na příznakové mapy fixní velikosti. V textu této práce je tento modul pro zjednodušení souhrnně nazýván RoI Align, přestože RoI Align je konkrétní matematická operace, která je součástí složitějšího procesu. Protože právě tato operace je zásadním krokem v jinak standardním procesu zpracování oblastí zájmu (He, 2018; Wu, 2019).



Obr. 2.6 – Blokové schéma modulu RoI Align dle Detectron2 implementace

Vstupem do modulu RoI Align jsou oblasti zájmu z RPN. Tyto oblasti jsou definovány v různých souřadnicových systémech v závislosti na tom, z jaké příznakové mapy z FPN pochází. Nejprve jsou tedy jejich souřadnice transformovány do souřadnicového systému obrazu vstupujícího do Faster R-CNN, aby bylo možné jejich vzájemné porovnávání z hlediska rozměrů. V dalším kroku je každé z takto transformovaných oblastí zájmu přiřazena jedna

z příznakových map z FPN na základě rozměrů. Poté jsou souřadnice oblasti zájmu opět transformovány, ale tentokrát do souřadnicového systému přiřazené příznakové mapy (Wu, 2019).

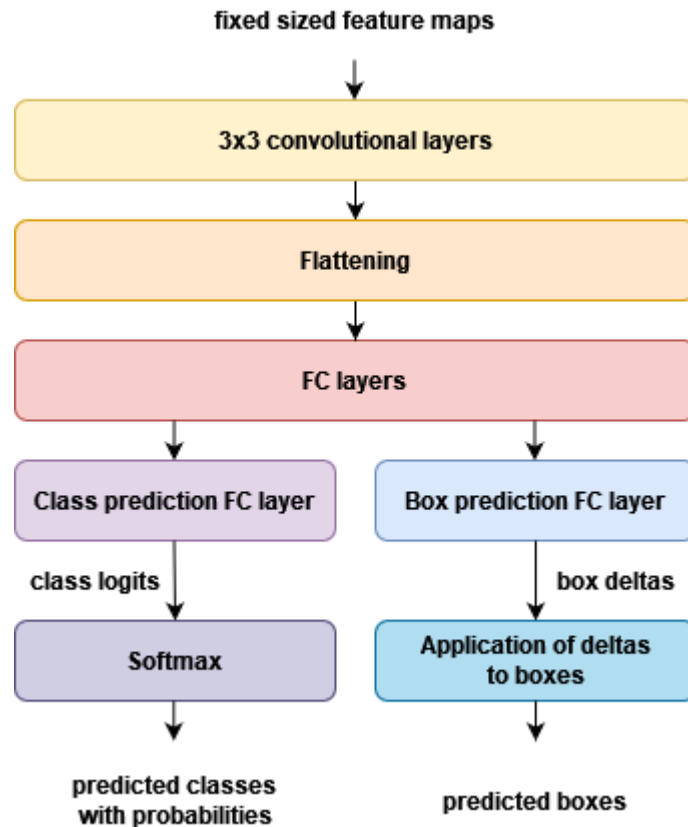
Následuje samotná operace RoI Align, jejíž úkolem je vytvořit reprezentaci oblasti zájmu ve formě příznakové mapy definovaných rozměrů. K tomu využívá bilineární interpolace. Oblast zájmu je rozdělena na stejné podoblasti, jejichž počet vychází z definovaných rozměrů výstupní příznakové mapy. Tedy v případě výstupu o rozměrech 7×7 bude počet podoblastí roven 49. V každé podoblasti se vypočtou pozice rovnoměrně rozložených bodů. Pro každý z těchto bodů se nalezne čtveřice sousedících pixelů příznakové mapy a vypočtou se vzdálenosti k jejich středům. Tyto vzdálenosti pak slouží jako váhy při výpočtu váženého průměru z hodnot sousedících pixelů. Výsledky takto vypočtených vážených průměrů pro všechny body v rámci jedné podoblasti jsou následně zprůměrovány do jedné hodnoty. Alternativou k průměrování je Max Pooling, který je méně výpočetně náročný, ale dochází při něm k větší ztrátě informací. Pro každou podoblast je ve výsledku vypočtena jedna hodnota, která ji charakterizuje. Tyto hodnoty pak tvoří příznakovou mapu o definovaných rozměrech reprezentující danou oblast zájmu, která je výstupem z modulu RoI Align a postupuje dále do finálních vrstev architektury (He, 2018; Wu, 2019).

Zásadní výhodou modulu RoI Align oproti modulu RoI Pooling, který je využíván prvotní verzí architektury, je absence kvantizace, a tudíž zachování informace o přesné poloze oblasti zájmu. Výsledkem transformace souřadnic definujících oblast zájmu jsou často desetinná čísla. Transformovaná oblast zájmu tedy nezapadá přímo do pixelové mřížky přiřazené příznakové mapy. RoI Pooling tento problém řeší již zmíněnou kvantizací, tedy procesem zaokrouhlování desetinných hodnot souřadnic tak, aby byly v souladu s pixelovou mřížkou. Nevyhnutelně pak dochází k posunutí oblasti zájmu a tím ztrátě informací. (He, 2018).

2.1.5 Finální vrstvy Faster R-CNN

Vstupem do finálních vrstev jsou příznakové mapy definované velikosti reprezentující oblasti zájmu, ze kterých vychází. Blokové schéma finálních vrstev je znázorněno na obr. 3.7. Každá příznaková mapa nejprve projde řadou konvolucí s filtrem 3×3 a ReLU aktivací, jejichž účelem je extrakce podstatných informací. Počet těchto konvolucí závisí na konkrétní konfiguraci architektury. Konvolucemi zpracovaná příznaková mapa je následně transformována do vektoru, aby mohla být vstupem do následujících plně propojených vrstev. Po každé z těchto vrstev následuje ReLU aktivace. Počet plně propojených vrstev rovněž závisí

na konkrétní konfiguraci architektury. Výstup těchto vrstev putuje do paralelní dvojice plně propojených vrstev, které vypočítají logity tříd a box delty. Při inferenci jsou logity pomocí softmax funkce přepočítány na pravděpodobnosti a na základě delt jsou vypočteny přesné rozměry a poloha predikovaných boxů. Finálním výstupem architektury při inferenci jsou tedy boxy s třídami přiřazenými s určitou pravděpodobností (Wu, 2019).



Obr. 2.7 – Blokové schéma finálních vrstev dle Detectron2 implementace

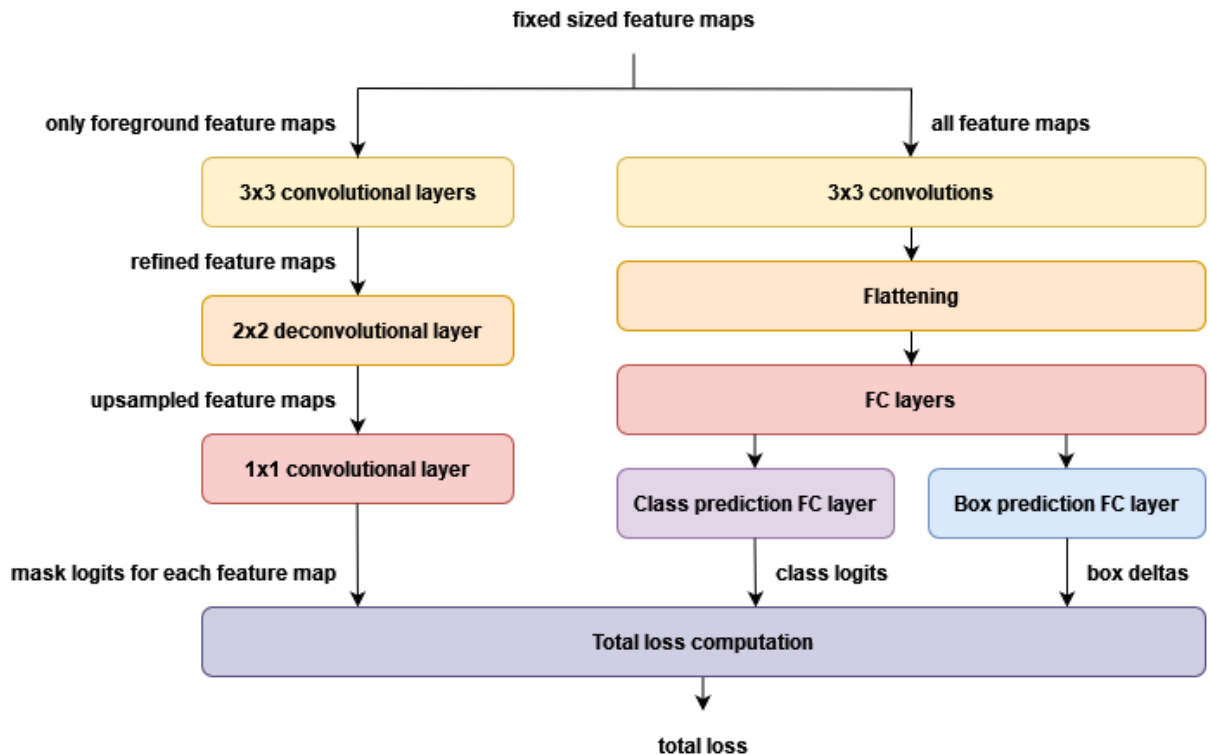
Během tréninku se boxy s třídami přiřazenými s určitou pravděpodobností standardně nevytváří. Místo nich je finálním výstupem architektury ztráta pracující přímo s logity a deltami. Celková ztráta je složena z váženého součtu klasifikační ztráty, vycházející z chyb při přiřazování tříd, a rozměrově-poziční ztráty boxů, vypočtené porovnáním box delt s referenčními boxy. Hodnota celkové ztráty je pak užita při úpravě vah pomocí trénovacího algoritmu, kterým je Stochastic Gradient Descent (SGD). Klasifikační ztráta se počítá křížovou entropií. Rozměrově-poziční ztráta boxů je standardně počítána metodou Smooth L1, která kombinuje kvadratickou a absolutní chybu tak, že kvadratický výpočet chyby je aplikován pouze na nižší hodnoty odchylek (Wu, 2019).

2.2 MASK R-CNN

Mask R-CNN je architektura pro detekci, klasifikaci a segmentaci objektů v obraze, tedy pro instanční segmentaci. Přímo navazuje na původní Faster R-CNN, od kterého se liší pouze přidáním dalších finálních vrstev pro predikci segmentačních masek a náhradou RoI Pooling modulu přesnějším RoI Align, bez kterého by byly predikované segmentační masky vlivem kvantizace příliš nepřesné (He, 2018). Instanční segmentace je v úloze detekce vad na osazených PCB velmi silným nástrojem, protože umožňuje mnohem přesnější lokalizaci objektů, než je možné prostřednictvím pouhých boxů, jelikož segmentační masky jsou tvořeny přímo na úrovni pixelů. Díky tomu je možné implementovat poměrně přesné měřicí metody, které umožňují robustní dodatečnou klasifikaci objektů na základě jejich rozměrů, tvarů, natočení a tak dále. V rámci této kapitoly budou popsány pouze finální vrstvy vybrané varianty Mask R-CNN, protože oproti již detailně popsané variantě architektury Faster R-CNN v kapitole 2.1 se liší pouze v nich.

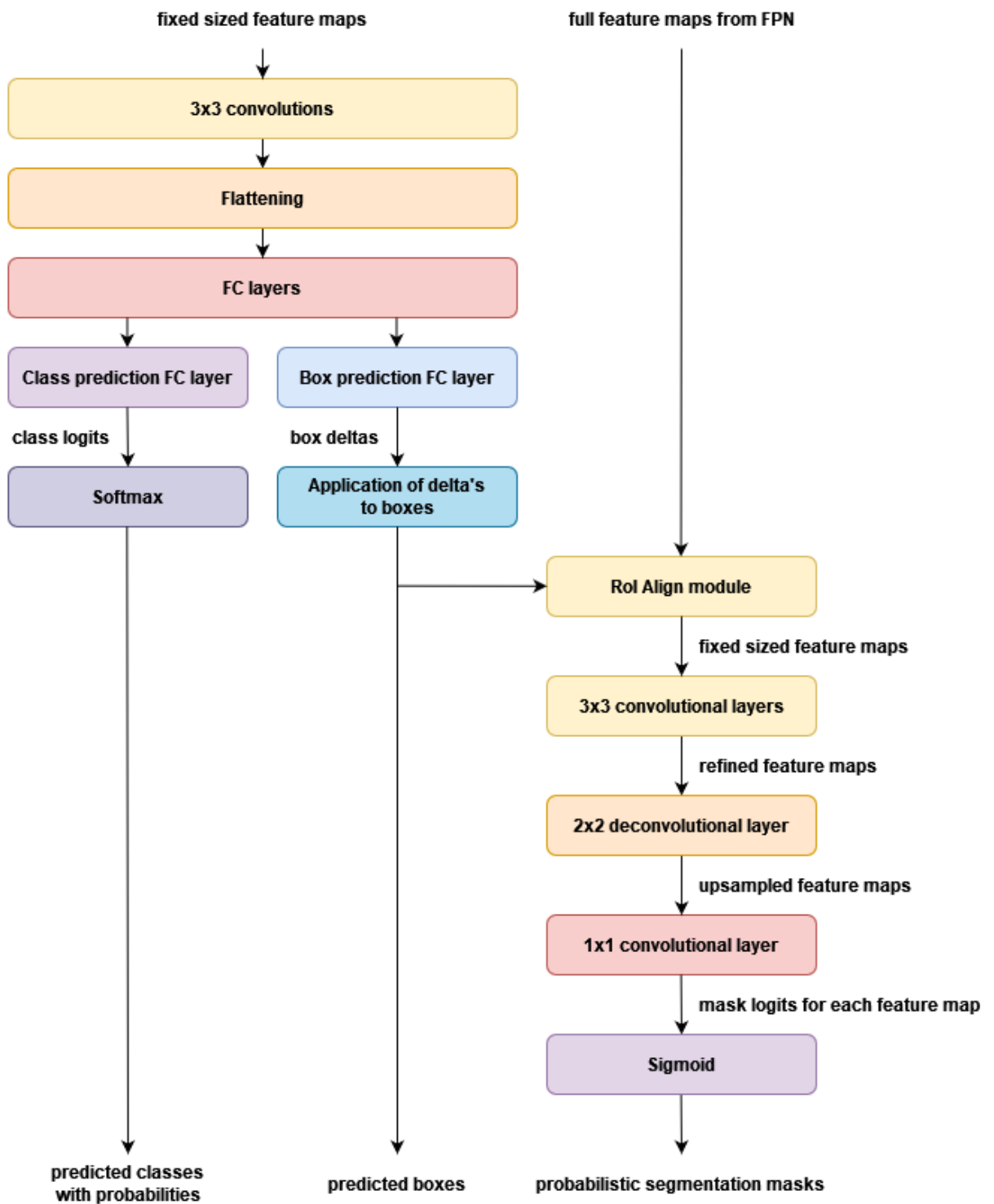
2.2.1 Finální vrstvy Mask R-CNN

Průběh predikce segmentační masky je při trénování a inferenci odlišný. Blokové schéma procesu predikce ve finálních vrstvách při tréninku je na obr. 2.8. Během tréninku probíhá predikce masek nezávisle na predikci boxů a tříd. Obě predikce vycházejí ze stejných příznakových map z modulu RoI Align, přičemž prediktor masek predikuje masky objektů pouze na příznakových mapách označených jako popředí, protože v pozadí žádné objekty nejsou. Příznakové mapy popředí vstupující do prediktoru masek nejprve projdou přes sérii 3×3 konvolučních vrstev s krokem 1 a ReLU aktivacemi, které z nich extrahují příznaky podstatné pro predikci masky. Počet těchto vrstev závisí na konkrétní konfiguraci architektury. Extrahované příznakové mapy dále putují skrz 2×2 dekonvoluční vrstvu s krokem 2 a ReLU aktivací, která zvětší jejich plošné rozměry, aby byla výsledná maska přesnější. Zvětšené extrahované příznakové mapy následně prochází 1×1 konvoluční vrstvou, která pro každý pixel mapy vypočte logit reprezentující míru, s jakou náleží k objektu. Porovnáním takto predikované masky s referenční maskou se poté binární křížovou entropií vypočte ztráta, která je spolu s klasifikační ztrátou a rozměrově-poziční ztrátou boxů součástí ztráty celkové (He, 2018; Wu, 2019).



Obr. 2.8 – Blokové schéma procesu finálních vrstev Mask R-CNN při tréninku

Při inferenci probíhá predikce segmentační masky až po predikci boxů a tříd, jak je znázorněno blokovým schématem na obr. 2.9, což zrychluje celý proces. Boxy a třídy jsou predikovány z příznakových map zpracovaných RoI Align modulem vycházejících přímo z RPN. Výsledkem této predikce je určitý počet boxů s nejvyšším skóre. Tyto boxy jsou následně využity k vytvoření nových příznakových map fixní velikosti z příznakových map z FPN prostřednictvím RoI Align modulu. Vytvořené příznakové mapy dále prochází stejným procesem jako při tréninku s tím rozdílem, že maska tvořená logity je přepočtena sigmoid funkcí na pravděpodobnostní segmentační masku. Ta je vedle boxů s třídami přiřazenými s určitou pravděpodobností finálním výstupem architektury Mask R-CNN (He, 2018; Wu, 2019).

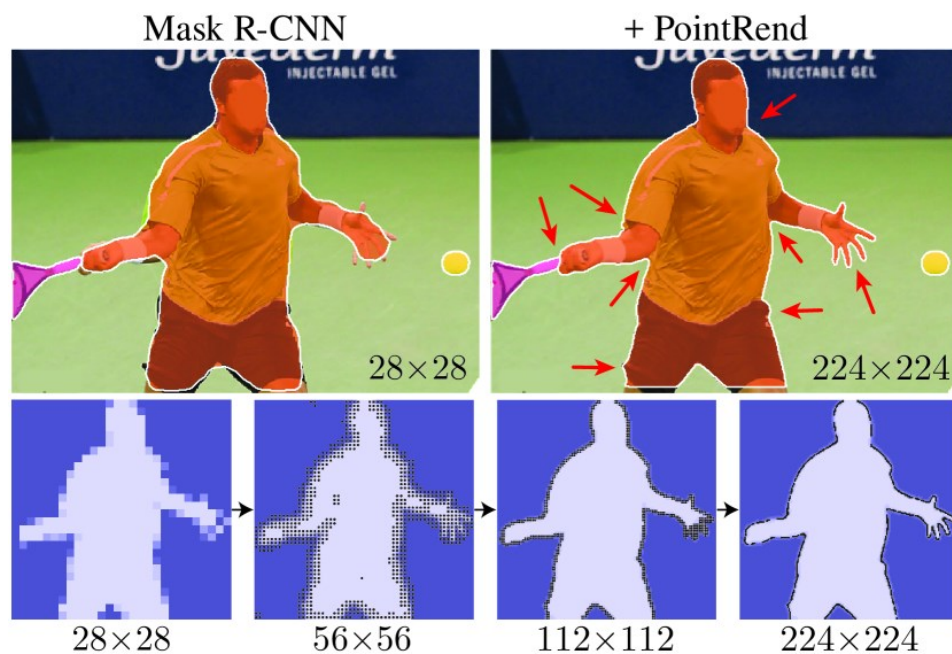


Obr. 2.9 – Blokové schéma procesu finálních vrstev Mask R-CNN při inferenci

2.2.2 PointRend

PointRend (Point-based Rendering) je rozšiřující modul architektur segmentujících obraz, jehož účelem je další zpřesnění výstupních pravděpodobnostních segmentačních masek (Kirillov, 2020). Tato kapitola se zaměřuje na implementaci tohoto modulu do architektury Mask R-CNN v Detectron2.

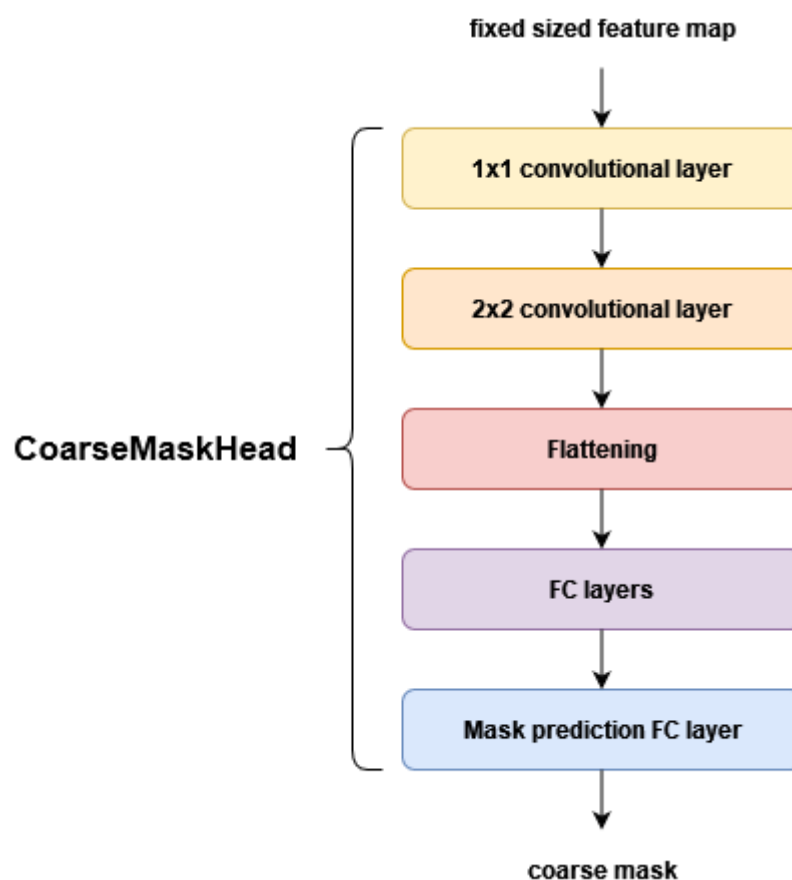
Segmentace architekturou Mask R-CNN není vždy dostatečně přesná na to, aby zachytila tvar složitějších objektů, což je převážně způsobeno nízkým rozlišením výstupní pravděpodobnostní segmentační masky. Zvýšení rozlišení celé masky, ale není na místě, protože by vedlo k výraznému zvýšení výpočetní náročnosti až do bodu, kdy by byla architektura příliš pomalá pro jakoukoliv praktickou aplikaci. PointRend tento problém řeší bodovou predikcí v adaptivně vybíraných oblastech, u kterých předpokládá nejvyšší míru nejistoty pravděpodobnostní segmentační masky (Kirillov, 2020). Na obr. 2.10 je znázorněn vliv aplikace PointRend modulu na pravděpodobnostní segmentační masku vystupující z architektury Mask R-CNN.



Obr. 2.10 – Vliv PointRend modulu na přesnost segmentace Mask R-CNN (Kirillov, 2020)

PointRend modul ve finálních vrstvách Mask R-CNN nahrazuje standardní proces predikující segmentační masku. Stejně jako u standardní Mask R-CNN je proces při inferenci odlišný od tréninku. První částí PointRend modulu je submodul CoarseMaskHead provádějící predikci tzv. hrubé masky, která je tvořena logity a je v podstatě ekvivalentní masce tvořené ve standardní Mask R-CNN. Na obr. 2.11 je blokové schéma procesu predikce této hrubé masky.

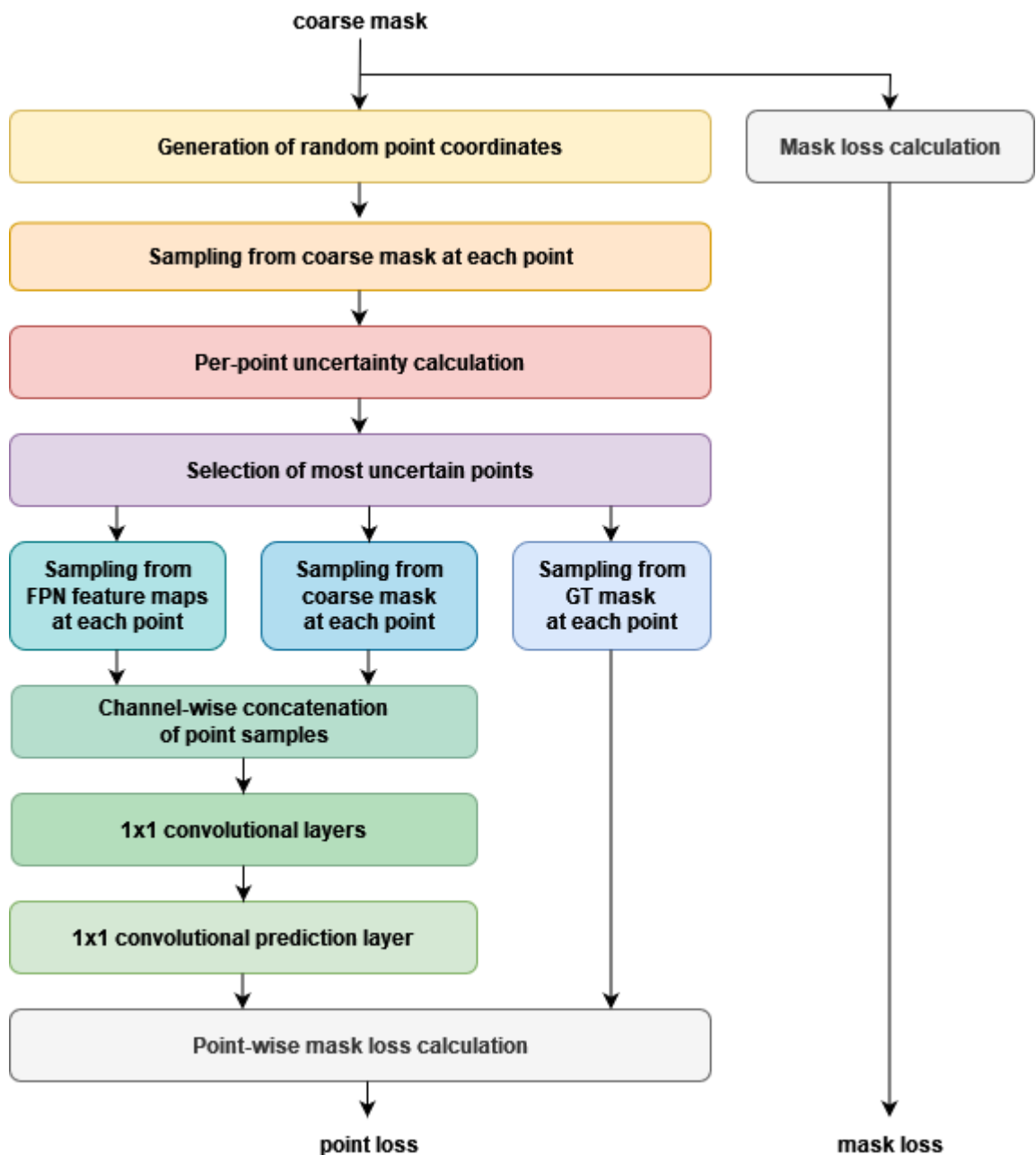
Při tréninku je vstupem do CoarseMaskHead submodule mapa příznaků fixní velikosti pocházející z RPN a zpracovaná RoI Align modulem. První 1×1 konvoluční vrstva s krokem 1 a počtem filtrů kompatibilním s dimenzí později následujících plně propojených vrstev pouze upravuje počet kanálů vstupní příznakové mapy. Po ní následuje 2×2 konvoluční vrstva s krokem 2, která dvakrát zmenšuje plošné rozměry vstupní příznakové mapy. Takto upravená příznaková mapa je transformována do vektoru, který postupuje do série plně propojených vrstev, a nakonec do poslední predikční plně propojené vrstvy. Po každé ze zmíněných vrstev s výjimkou té predikční následují ReLU aktivace. Výstupem je hrubá maska tvořená logity (Wu, 2019; Kirillov, 2020).



Obr. 2.11 – Blokové schéma procesu predikce hrubé masky modulem PointRend

Obr. 2.12 znázorňuje blokovým schématem proces probíhající v PointRend modulu během tréninku. Vstupem do tohoto procesu je hrubá maska z CoarseMaskHead. Na základě této masky je porovnáním s referenční maskou vypočtena ztráta ekvivalentní ztrátě přesnosti masky Mask R-CNN. Hrubá maska také prochází bodovou predikcí. V rámci této predikce jsou nejprve vygenerovány náhodné souřadnice definovaného počtu bodů. Hodnoty souřadnic jsou

desetinná čísla normalizovaná na rozsah 0 až 1 v rámci souřadnicového systému hrubé masky. Ze všech náhodných bodů hrubé masky je odebrán vzorek. Souřadnice jsou ale desetinná čísla, která přesně nezapadají do mřížky hrubé masky, proto je při odebírání vzorku využito bilineární interpolace, podobně jako v modulu RoI Align. Výsledkem je skalární hodnota zachovávající informaci o sousedních pixelech náhodně generovaného bodu. Pro každý ze vzorků hrubé masky je vypočtena nejistota, což je inverze vzdálenosti hodnoty logitu od nuly. Tedy čím dále je hodnota logitu od nuly, tím menší je nejistota (Wu, 2019).



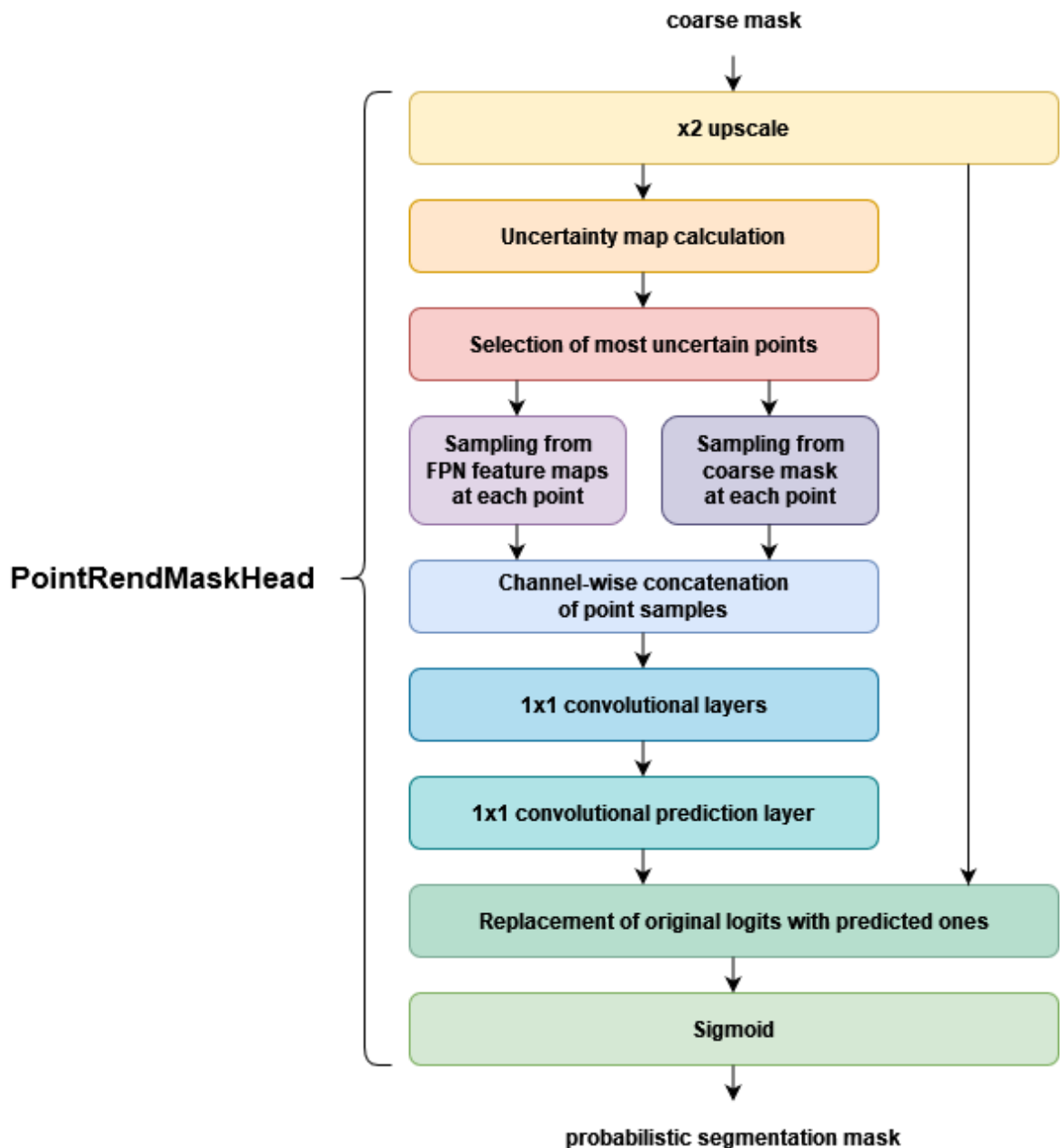
Obr. 2.12 – Blokové schéma modulu PointRend při tréninku

Na základě vypočtených nejistot je vybrán definovaný počet bodů s nejvyšší nejistotou. Tyto body jsou následně využity pro extrakci bodových vzorků ze všech map příznaků z FPN. Před touto extrakcí jsou jejich souřadnice transformovány ze souřadnicového systému hrubé masky do souřadnicového systému příznakových map z FPN. Také je znovu pro každý z vybraných bodů extrahován bodový vzorek z hrubé masky, což vychází ze struktury Detectron2 softwaru implementujícího PointRend, kde bylo pravděpodobně výhodnější vzorky odebrat znovu, než je udržovat v rámci složitější datové struktury spolu se souřadnicemi bodů. Vzorky odebrané z hrubé masky a z příznakových map z FPN jsou pro každý bod zvlášť spojeny podél kanálové dimenze tak, že každý kanál obsahuje jiný vzorek (Wu, 2019).



Obr. 2.13 – Blokové schéma finálních vrstev Mask R-CNN PointRend při inferenci

Tyto vícekanálové bodové vzorky dále prochází sérií 1×1 konvolučních vrstev s krokem 1 a finální predikční 1×1 konvoluční vrstvou s krokem 1, která pro každý z nich vypočte jeden logit. 1×1 konvoluční vrstvy jsou v tomto případě ekvivalentní plně propojeným vrstvám s tím rozdílem, že není nutná transformace vstupních vzorků do vektoru. Nakonec je vypočtena ztráta bodové predikce porovnáním vypočtených logitů jednotlivých bodů s logity ve stejných bodech referenční segmentační masky. Tato ztráta je spolu se standardní ztrátou masky, klasifikační ztrátou a rozměrově-poziční ztrátou boxů součástí celkové ztráty (Wu, 2019).



Obr. 2.14 – Proces zpřesnění segmentační masky modulem PointRender při inferenci

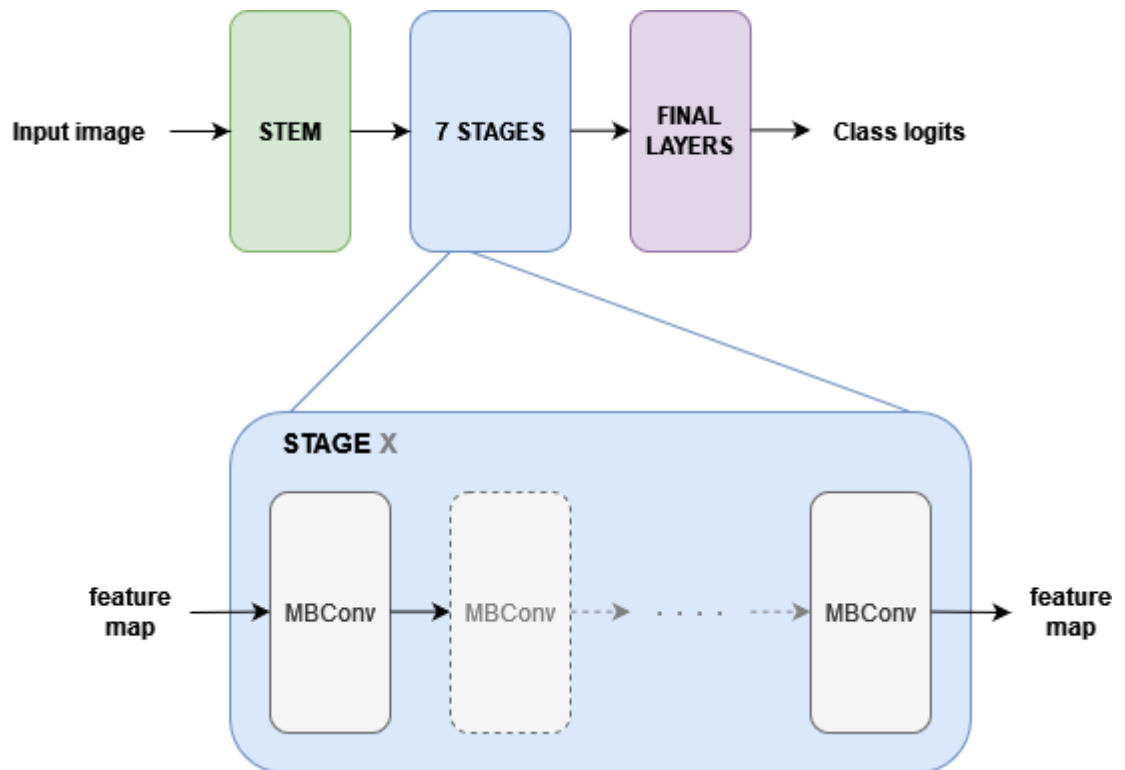
Při inferenci probíhá predikce masky až po predikci boxů stejně jako u Mask R-CNN, a to logicky platí i pro bodovou predikci. Na obr. 2.13 je blokové schéma finálních vrstev Mask R-CNN s modulem PointRend při inferenci. Příznakové mapy fixní velikosti nejprve projdou submodule CoarseMaskHead, jehož blokové schéma je na obr. 2.11, stejně jako při tréninku. Vypočtené hrubé masky dále postupují do submodule PointRendMaskHead, jehož blokové schéma je na obr. 2.14. V rámci tohoto submodule je hrubá maska zvětšena na dvojnásobnou velikost. Poté je pro každý logit této zvětšené masky vypočtena nejistota, čímž je vytvořena mapa nejistot. Z této mapy je vybrán definovaný počet bodů s nejvyšší hodnotou nejistoty. Následující proces extrakce vzorků a výpočtu logitu pro každý z vybraných bodů je stejný jako při tréninku. Vypočtené logity jsou použity k úpravě zvětšené hrubé masky, která spočívá v prostém nahrazení původních logitů těmi novými. Zvětšená hrubá maska je nakonec sigmoid funkcí převedena na pravděpodobnostní segmentační masku, která je finálním výstupem PointRend modulu a zároveň i celé Mask R-CNN PointRend architektury (Wu, 2019).

2.3 EFFICIENTNET

EfficientNet je typ hluboké konvoluční neuronové sítě pro klasifikaci celého obrazu, který jako první zavedl tzv. složené škálování sítě ve třech dimenzích: hloubce, šířce a rozlišení vstupu. Hloubkou sítě je myšlen počet vrstev a šířkou sítě počty filtrů jednotlivých konvolučních vrstev. EfficientNet ukazuje, že škálování těchto dimenzí jednotným měřítkem umožňuje dosáhnout efektivní rovnováhy mezi přesností, počtem parametrů, tedy velikostí modelu, a výpočetní náročností. Ve výsledku tedy může dosáhnout vyšší přesnosti s nižším počtem parametrů než sítě škálované jen jednou dimenzí, jako je např. ResNet popsáný v kapitole 2.1.1. Základní EfficientNet je k dispozici celkem v 8 stupních škálování označených B0 až B7. Verze B0 má nejnižší potenciální přesnost a zároveň nejnižší výpočetní náročnost, zatímco verze B7 je z nich nejpresnější za cenu vyšší výpočetní náročnosti. Volba správné verze se odvíjí od složitosti konkrétní úlohy. Například u jednoduchých úloh může dosahovat verze B0 srovnatelné přesnosti jako B7, ale s mnohem nižší výpočetní náročností (Tan, 2020).

Blokové schéma obecné EfficientNet architektury je na obr. 2.15. EfficientNet je svou základní strukturou podobný architektuře ResNet. Je složen ze Stem bloku následovaného sedmi fázemi a zakončený finálními vrstvami, které na základě zpracovaných příznakových map predikují kategorii vstupního obrazu. Stejně jako ResNet, má EfficientNet svůj základní stavební blok. Jedná se o tzv MConv, z anglického mobile inverted bottleneck convolution, rozšířený o squeeze-and-excitation (SE) vrstvu. Každá ze sedmi fází je tvořena určitým počtem

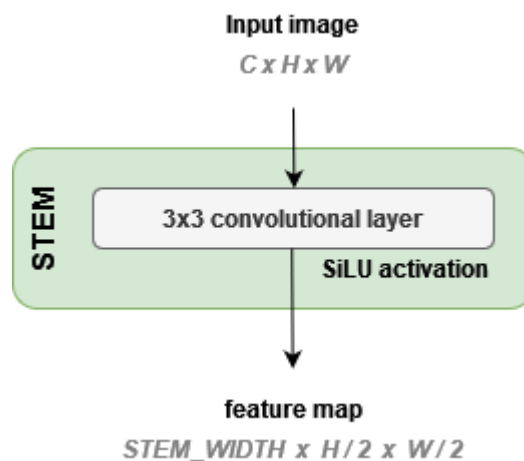
těchto bloků. Počty bloků na fázi jsou škálovatelné a napříč verzemi B0 až B7 se tedy liší (Tan, 2020).



Obr. 2.15 – Blokové schéma obecné EfficientNet architektury

2.3.1 Stem

Stem blok obecné EfficientNet architektury je tvořen pouze jednou 3×3 konvoluční vrstvou s krokem 2, která dvakrát snižuje plošné rozměry a navyšuje počet kanálů dle



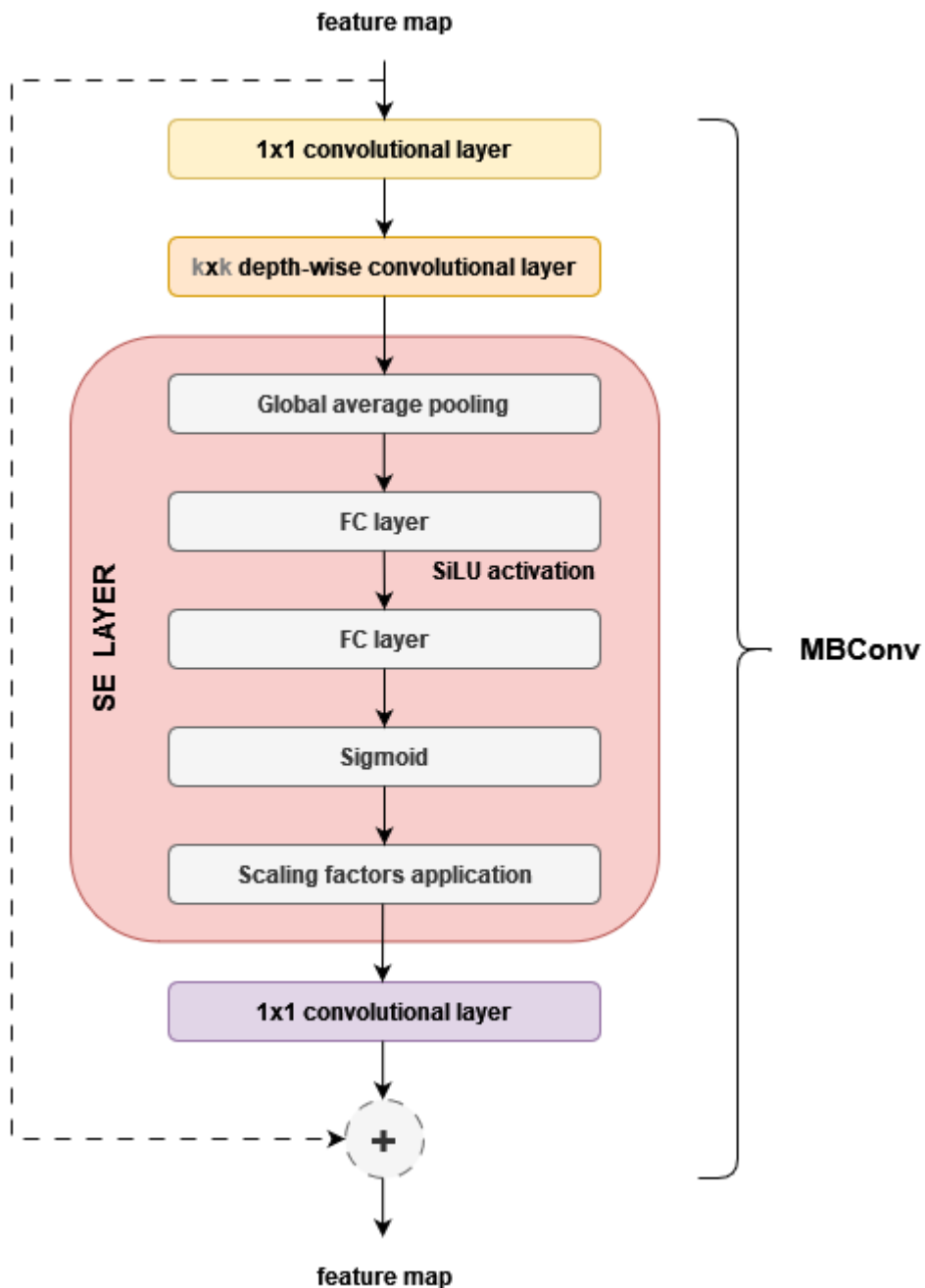
Obr. 2.16 – Blokové schéma EfficientNet Stem bloku dle pycls implementace

definovaného počtu filtrů. Tato vrstva je následována dávkovou normalizací a SiLU aktivací (Tan, 2020; Radosavovic, 2021).

SiLU, neboli swish aktivační funkce, je využívána v celé pycls EfficientNet implementaci. Oproti široce využívané ReLU aktivaci má SiLU výhodu v tom, že v menší míře uchovává i informace o záporných hodnotách, a nedochází tak trvalým nulovým spojením. To pomáhá dosáhnout lepších výsledků při trénování sítě, za cenu vyšší výpočetní náročnosti. EfficientNet má ale v porovnání například s Faster R-CNN architekturou, popsanou v kapitole 2.1, mnohem méně vrstev, použití této aktivační funkce má tedy mnohem nižší dopad (Elfwing, 2017; Tan, 2020).

2.3.2 MBConv blok

Jak již bylo zmíněno, MBConv bloky tvoří jednotlivé fáze EfficientNet architektury. Obecná struktura tohoto bloku je na obr. 2.17. Vstupem do MBConv bloku je vždy příznaková mapa určitých rozměrů. Tato mapa nejprve prochází 1×1 konvoluční vrstvou s krokem 1, která navyšuje její počet kanálů určitým poměrem definovaným v konfiguraci sítě. Následně prochází hloubkovou konvoluční vrstvou, která je hlavním extraktorem příznaků. Velikost filtru je závislá na pořadí fáze a velikost kroku na pořadí bloku ve fázi. V prvním bloku fáze má tato hloubková konvoluční vrstva krok 2 a dvojnásobně tak snižuje plošné rozměry příznakové mapy. Ve zbylých blocích fáze je krok roven 1. Následuje SE vrstva, jež pro každý kanál příznakové mapy vypočte škálovací faktor, což je číslo, kterým jsou následně vynásobeny všechny hodnoty daného kanálu. Tím dochází k utlumení kanálů, které obsahují méně důležité informace, a zvýraznění těch užitečnějších. Při výpočtu škálovacích faktorů se vychází z příznakové mapy. Každý kanál této mapy je nejprve zprůměrován do skalární hodnoty. Vektor těchto hodnot, jehož délka odpovídá počtu kanálů příznakové mapy, dále prochází plně propojenou vrstvou se SiLU aktivací následovanou další plně propojenou vrstvou. Na výstup těchto vrstev je aplikována sigmoid funkce za účelem konverze hodnot do rozsahu 0 až 1. Těmito hodnotami jsou následně vynásobeny příslušné kanály příznakové mapy. Poslední vrstvou MBConv bloku je 1×1 konvoluční vrstva s krokem 1, která mění počet kanálů na hodnotu definovanou v konfiguraci. Počet kanálů je škálovatelný a fázi po fázi se postupně navyšuje. Ve výsledku každá fáze dvojnásobně sníží plošné rozměry a navýší počet kanálů příznakové mapy, která do ní vstoupila. Kromě prvního bloku mají všechny bloky fáze přímou vazbu vstupu na výstup, stejně jako residuální blok architektury ResNet (Tan, 2020; Radosavovic, 2021).

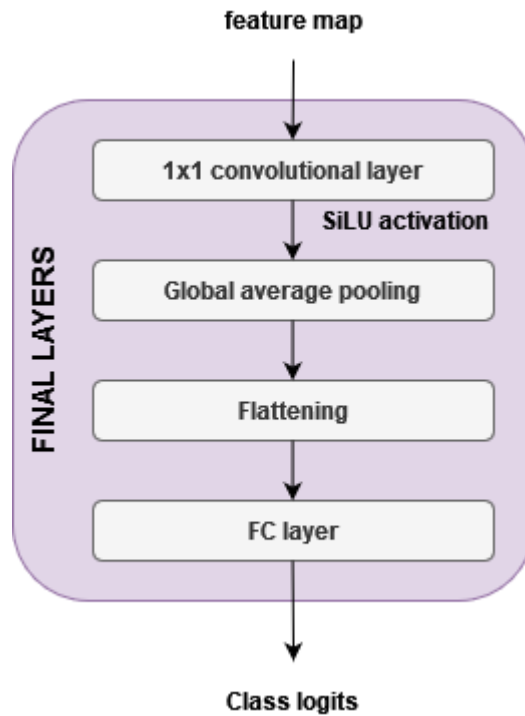


Obr. 2.17 – Blokové schéma EfficientNet MBConv bloku dle pycls implementace

2.3.3 Finální vrstvy architektury EfficientNet

Účelem finálních vrstev architektury EfficientNet, znázorněných na obr. 2.18, je predikce kategorií vstupního obrazu na základě zpracované příznakové mapy. První 1×1 konvoluční vrstva s krokem 1 a SiLU aktivací navyšuje počet kanálů na definovanou škálovatelnou hodnotu. Pro každý kanál je následně vypočtena průměrná hodnota. Tyto hodnoty se transformují do vektoru a prochází skrz plně propojenou vrstvu, jejímž výstupem je vektor logitů jednotlivých kategorií. Tento vektor je finálním výstupem architektury

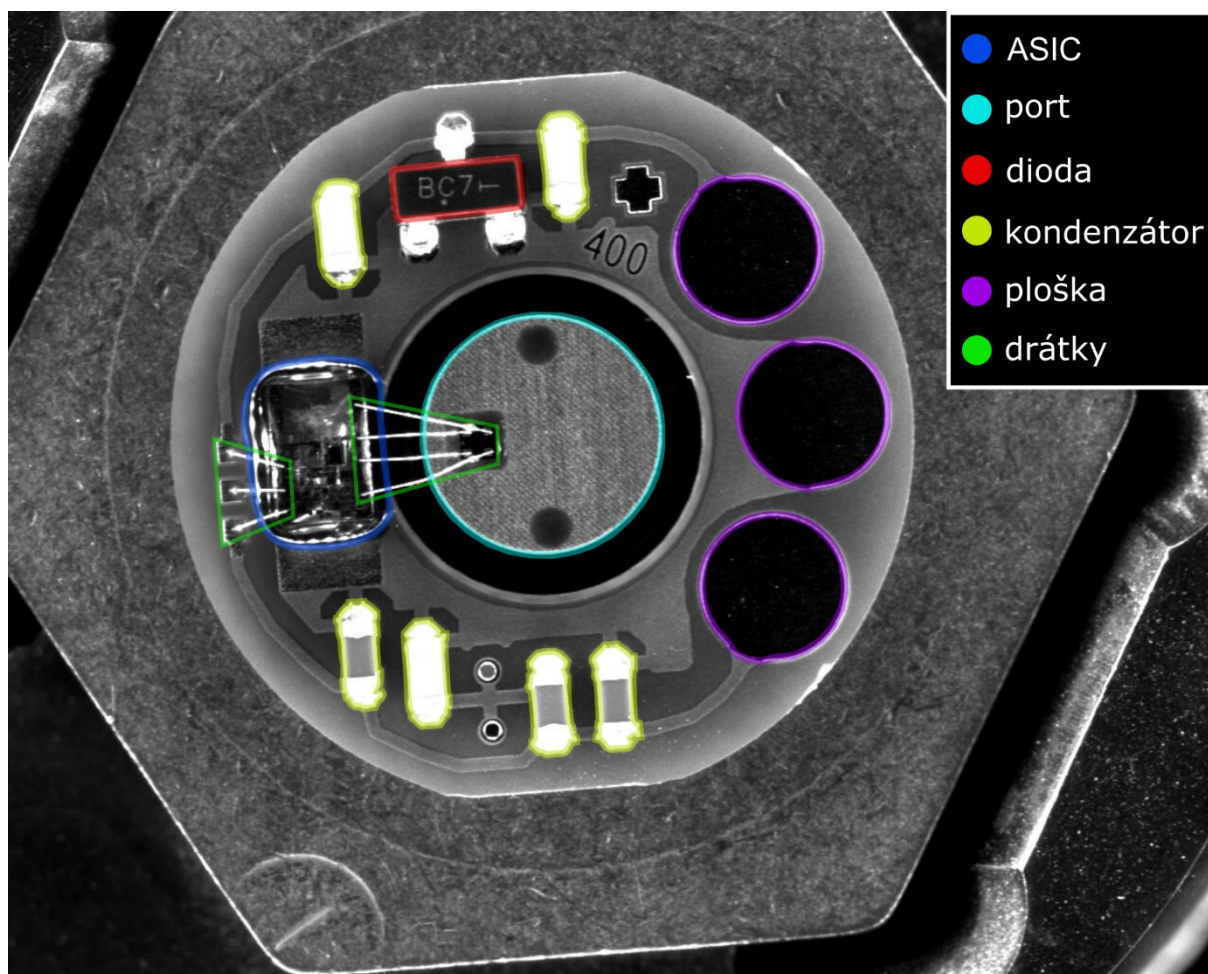
EfficientNet. Pro přehlednější znázornění výsledků je vhodné na získané logity aplikovat softmax funkci, která vypočte pravděpodobnostní distribuci kategorií. Při tréninku je pomocí křížové entropie z logitů a referenčních hodnot vypočtena ztráta, která je použita k úpravě vah algoritmem SGD (Tan, 2020; Radosavovic, 2021).



Obr. 2.18 – Finální vrstvy EfficientNet architektury dle pycls implementace

3 DEFINICE DETEKČNÍ ÚLOHY

Stávající vizuální inspekce osazených PCB na výrobní lince ve firmě Vitesco Technologies v Trutnově je prováděna operátorem na fotografiích pořízených zavedeným kamerovým systémem. Navrhovaný detekční systém by měl v konečné fázi nahradit operátora. Na obr. 3.1 je popis ukázkové inspekční fotografie. Za účelem splnění podmínek pro uveřejnění této diplomové práce nebude blíže specifikováno, o jaké zařízení se jedná, jak a z čeho se vyrábí a nebude zde detailněji popisován ani možný původ zmiňovaných defektů.



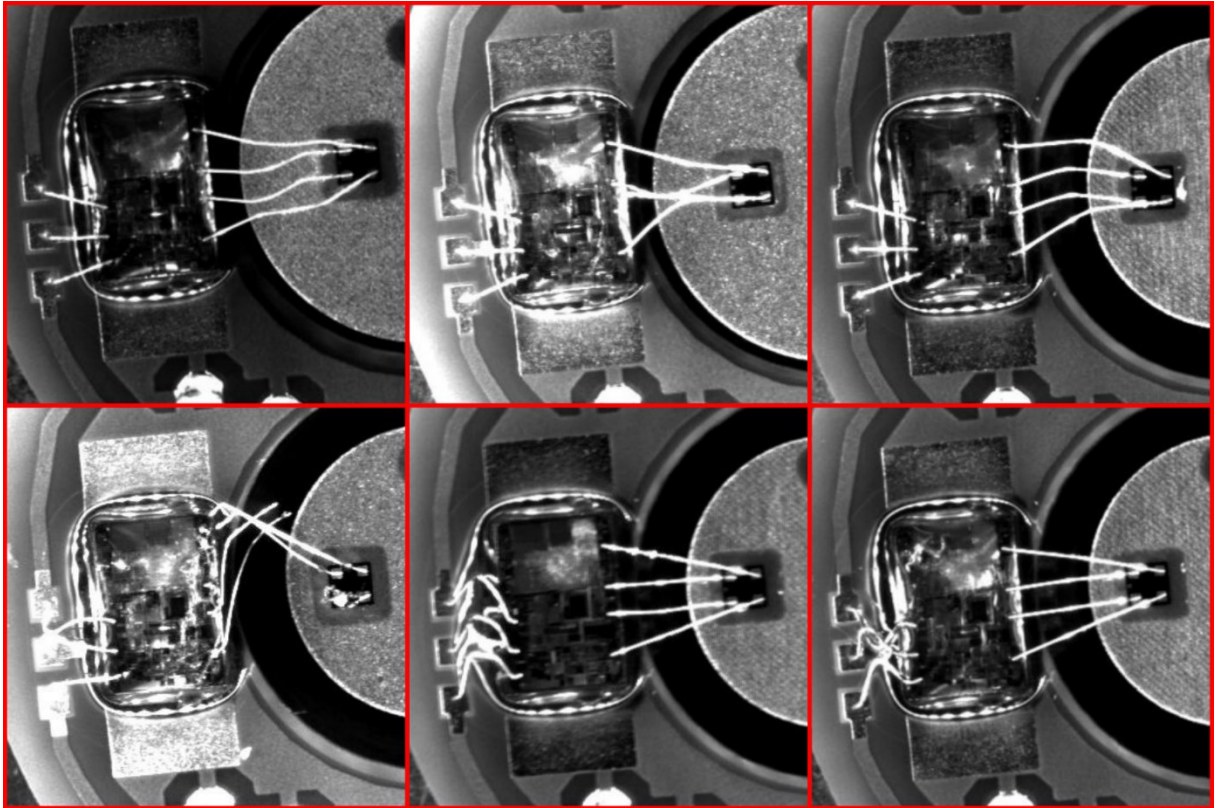
Obr. 3.1 – Popis ukázkové inspekční fotografie

3.1 IDENTIFIKACE A KATEGORIZACE DEFEKTŮ

Identifikace a kategorizace jednotlivých defektů vychází z historických obrazových dat poskytnutých firmou Vitesco Technologies a informací od znalých zaměstnanců firmy. Pojmem komponenty jsou v následujících kapitolách označovány kondenzátory a dioda.

3.1.1 Deformace bondovaných drátků

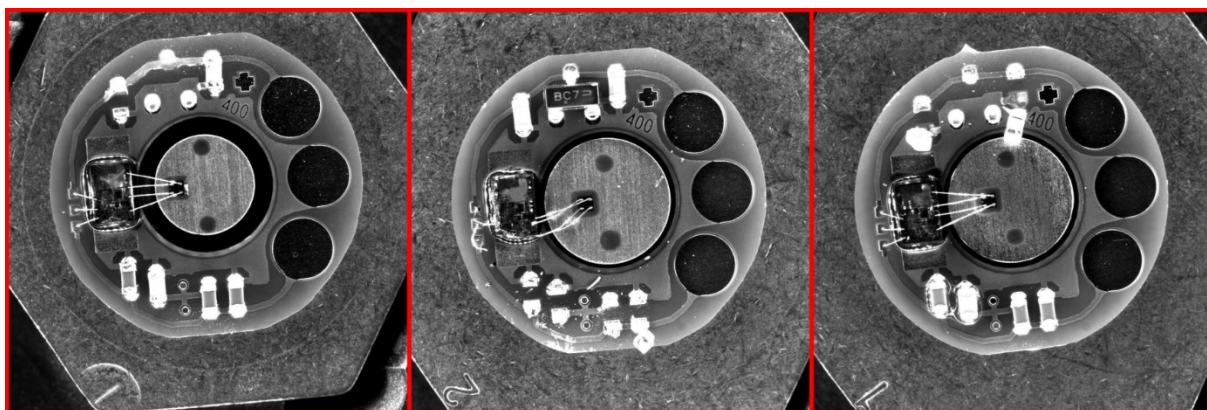
Bondované drátky jsou kritickým komponentem vyráběného zařízení a jejich deformace nebo úplné odtržení znamená ztrátu funkce. Detekce tohoto defektu má tedy vysokou prioritu. Na obr. 3.2 jsou příklady možných typů defektů bondovaných drátků.



Obr. 3.2 – Ukázky různých typů defektů bondovaných drátků

3.1.2 Chybějící komponenty

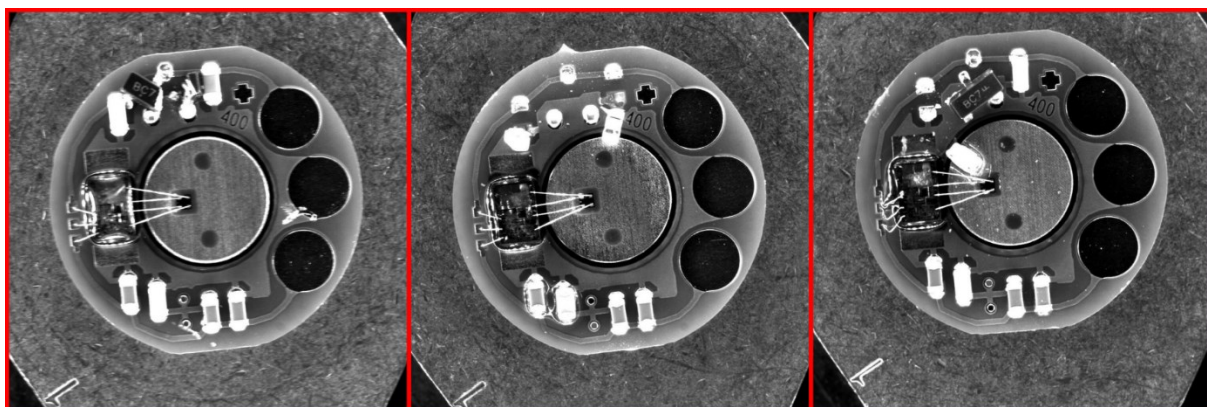
Chybějící komponenty jsou pochopitelně dalším kritickým defektem. Jejich detekce je ale poměrně snadná, protože bývají doprovázeny kontaminací. Je důležité si uvědomit, že z hlediska detekce objektů pomocí konvolučních neuronových sítí znamená i zakrytí komponentu, vlivem kontaminace, chybějící komponent. Na obr. 3.3 jsou příklady inspekčních fotografií s chybějícími komponenty.



Obr. 3.3 – Ukázky inspekčních fotografií s chybějícími komponenty

3.1.3 Chyba pozice nebo natočení komponent

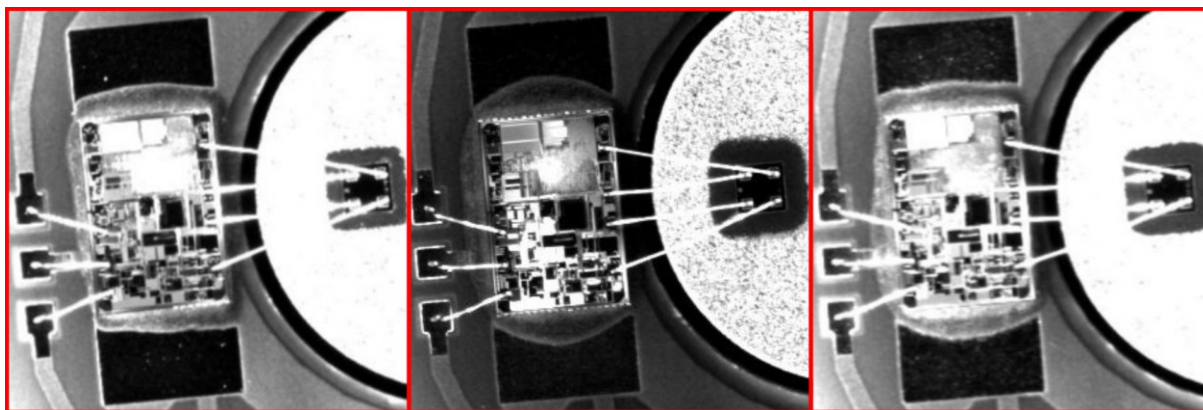
Špatná pozice nebo natočení komponent je také kritickým defektem. Stejně jako v případě chybějících komponent bývá tento defekt doprovázen kontaminací. Na obr. 3.3 jsou příklady inspekčních fotografií s komponenty mimo pozici a špatným natočením.



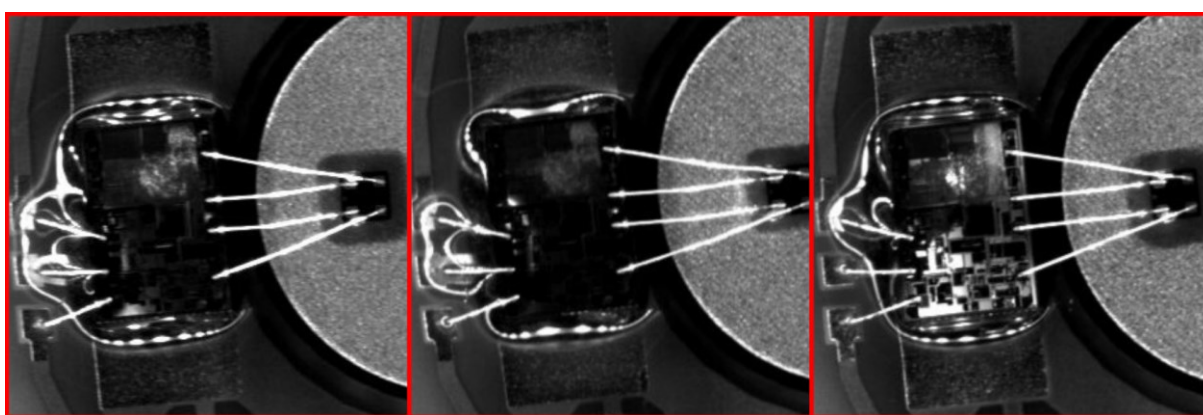
Obr. 3.4 – Ukázky inspekčních fotografií s komponenty mimo pozici a špatným natočením

3.1.4 Nadměrné množství nebo absence gelu na ASIC

Ochranný gel na ASIC (Application Specific Integrated Circuit) upevňuje bondy a zabraňuje jejich postupné degradaci. Chybějící gel tedy představuje riziko přerušení bondovaného spoje. Naopak nadměrné množství gelu může způsobit deformaci bondovaných drátků. Oba defekty jsou považovány za kritické. Na obr. 3.5 jsou ukázky absence gelu a na obr. 3.6 ukázky nadměrného množství gelu.



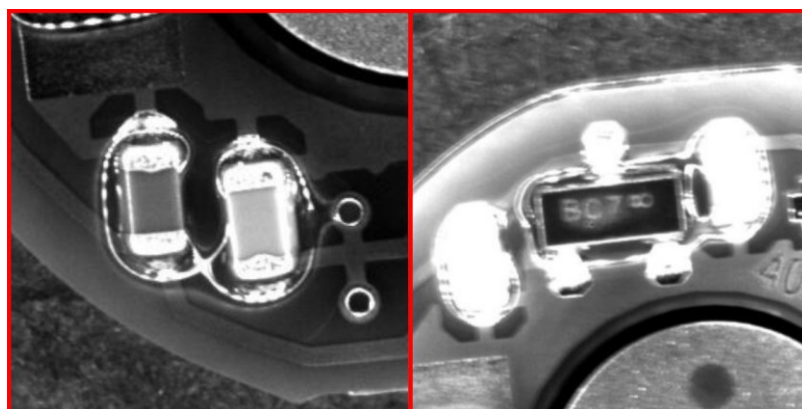
Obr. 3.5 – Ukázky vzhledů absence ochranného gelu na ASIC



Obr. 3.6 – Ukázky vzhledů nadměrného množství ochranného gelu na ASIC

3.1.5 Přítomnost gelu na komponentech

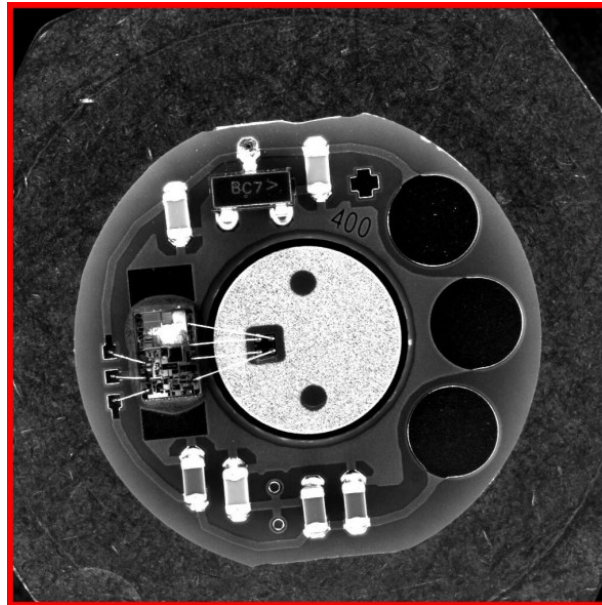
Ochranný gel se může dostat na komponenty, což není považováno za kritický defekt. Tento defekt by ale měl být detekován, a to hlavně v případě jeho systematického opakování. Na obr. 3.7 jsou ukázky ochranného gelu na komponentech.



Obr. 3.7 – Ukázky ochranného gelu na komponentech

3.1.6 Absence gelu na portu

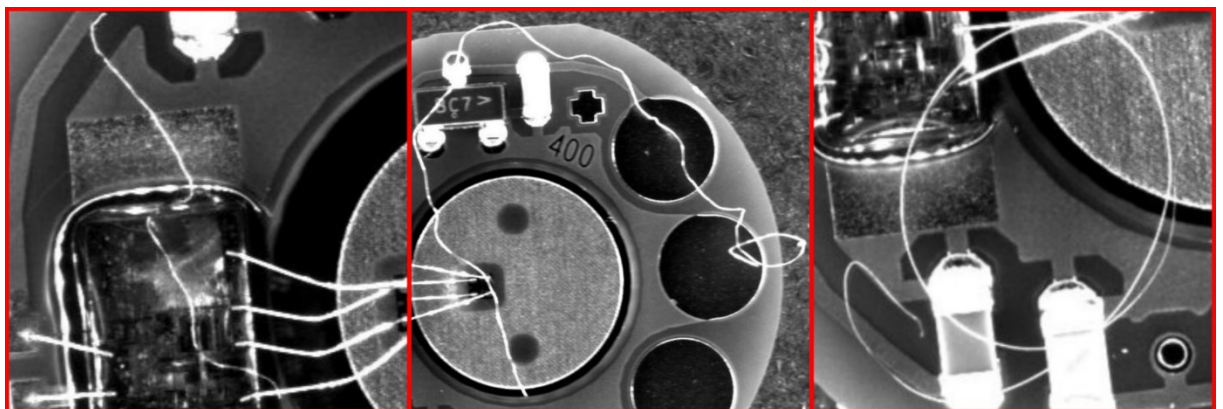
Absence nebo nedostatek gelu na portu je kritický defekt. Na fotografiích se vyznačuje velmi světlým až bílým zbarvením portu, protože při nedostatku gelu je zvýšena odrazivost světelných paprsků do kamery. Na obr. 3.8 je inspekční fotografie s chybějícím gelem na portu.



Obr. 3.8 – Ukázka inspekční fotografie s chybějícím gelem na portu

3.1.7 Kontaminace vlákny

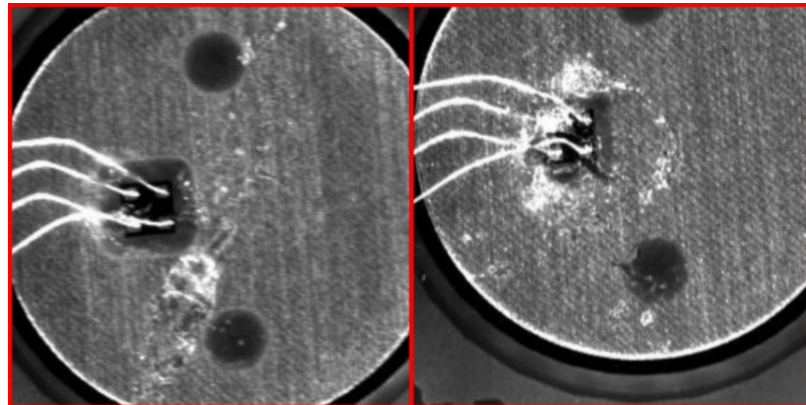
V případě silné kontaminace nebo přítomnosti vláken v místě bondovaných drátků je kontaminace vlákny považována za kritický defekt. Nicméně detekce vláken obecně je důležitá pro prevenci systematické kontaminace. Na obr. 3.9 jsou příklady inspekčních fotografií s kontaminací vlákny.



Obr. 3.9 – Ukázky inspekčních fotografií s kontaminací vlákny

3.1.8 ECA kontaminace

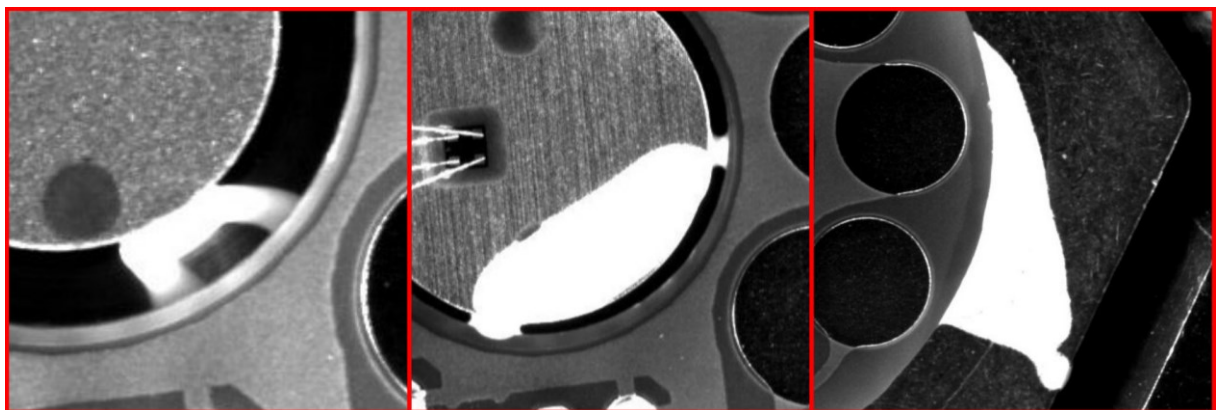
ECA (Electrically conductive adhesive) kontaminace jsou nežádoucí v místě bondovaných drátků, na ASIC a na komponentech, kde se jedná o kritický defekt. Dále jsou nežádoucí ve větším množství na portu a ploškách. Samotná přítomnost ECA by měla být detekována pro prevenci systematické kontaminace. Na obr. 3.10 jsou příklady ECA kontaminací.



Obr. 3.10 – Ukázky ECA kontaminací

3.1.9 Nadměrné množství a kontaminace silikonem

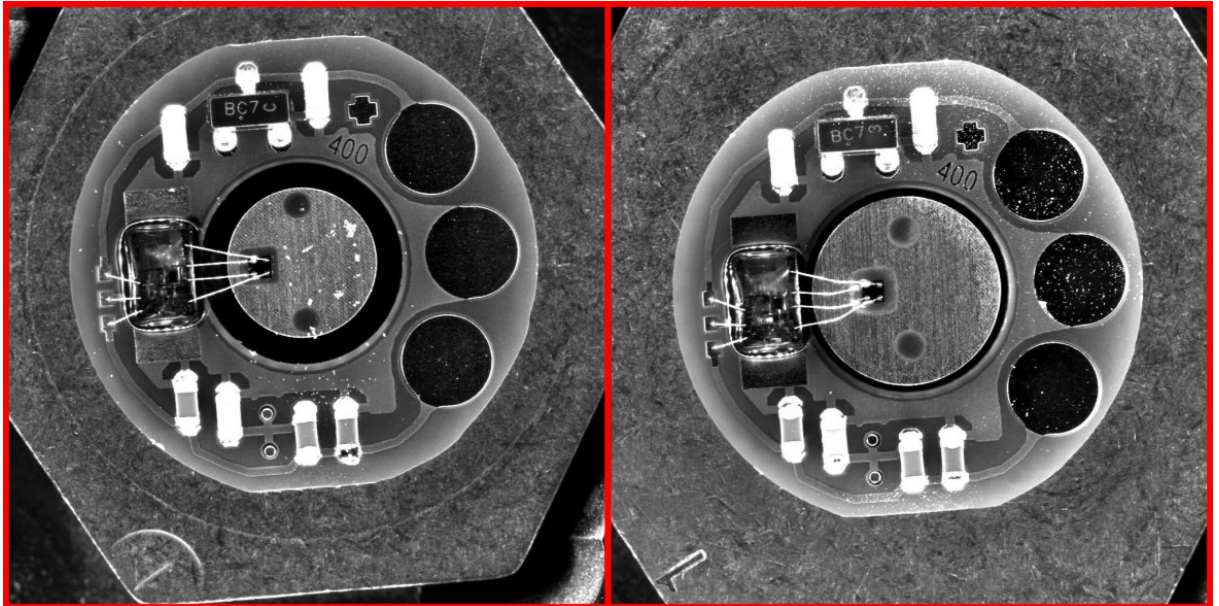
Nadměrné množství silikonu se projeví vytékáním z pod PCB, kde je kritické pouze ve větším množství, nebo viditelností v mezeře okolo portu, kde je kritické vždy. Silikon se může objevit i přímo na PCB, kde nesmí přijít do kontaktu s drátkou, ASIC, komponenty a ve větším množství s ploškami a portem. Mimo kritické případy je jeho detekce, stejně jako u ostatních typů kontaminace, vhodná pro prevenci systematické kontaminace. Na obr. 3.11 jsou ukázky nadměrného množství silikonu a ukázka silné kontaminace silikonem na portu.



Obr. 3.11 – Ukázky nadměrného množství a kontaminace silikonem

3.1.10 Kontaminace částicemi

Kontaminace částicemi je kritická jen ve velkém množství a v takovém případě je obvykle doprovázena dalšími defekty. Nežádoucí je hlavně na ploškách. Za účelem prevence systematické kontaminace částicemi je vhodné částice detekovat, ale prioritou není vysoká. Na obr. 3.12 jsou příklady inspekčních fotografií se silnou kontaminací částicemi.



Obr. 3.12 – Ukázky inspekčních fotografií se silnou kontaminací částicemi

3.1.11 Shrnutí identifikovaných kategorií defektů

V tab. 3.1 jsou přehledně shrnuty všechny kategorie defektů, které byly identifikovány z historických obrazových dat, orientačně seřazeny dle priority detekce. První tři kategorie defektů jsou kritické, protože s nimi zařízení ztrácí funkci. Kategorie 4 a 5 jsou kritické, protože nevyhnutelně vedou ke ztrátě funkce zařízení v blízkém časovém horizontu. Kategorie 6 až 8 jsou kontaminace, jejichž kritičnost je závislá na jejich míře a pozici. Kategorie 9 a 10 jsou nežádoucí, ale nemusí nutně způsobit selhání zařízení.

Tab. 3.1 – Identifikované kategorie defektů orientačně seřazené dle priority

1	Deformace bondovaných drátků
2	Chybějící komponenty
3	Chyba pozice nebo natočení komponent
4	Nadměrné množství nebo absence gelu na ASIC
5	Absence gelu na portu
6	Kontaminace vlákny
7	Nadměrné množství a kontaminace silikonem
8	ECA kontaminace
9	Kontaminace částicemi
10	Přítomnost gelu na komponentech

4 NÁVRH DETEKČNÍHO SYSTÉMU

Dle tab. 3.1 bylo z historických obrazových dat identifikováno celkem 10 kategorií defektů, což není příliš velké množství. Teoreticky by bylo možné vyřešit detekci všech těchto kategorií jedním modelem provádějícím instanční segmentaci a klasifikaci objektů doplněným o dodatečná měření. Takový model by ale vyžadoval obrovský trénovací dataset, jehož tvorba by jedinci zabrala stovky hodin, a jeho optimalizace by byla vzhledem k počtu kategorií velmi náročná. Navíc by ani nebylo předem jisté, zda je optimalizace takového modelu na úroveň uspokojivé přesnosti v rozumném časovém horizontu vůbec možná. Problémem by byl i nevyhnutelný nepoměr v zastoupení jednotlivých tříd. Zatímco třída vyskytující se na každé inspekční fotografii několikrát, protože se například jedná o nějaký klíčový komponent zařízení, by byla na hranici přetrénování, třídy vyskytující se jen zřídka, jako například různé kontaminace, by stále nebyly detekovány s dostatečnou přesností. S tím souvisí i rozdílnost ve vizuální variabilitě jednotlivých tříd. Třídy s vysokou vizuální variabilitou vyžadují mnohonásobně větší počet trénovacích vzorků. Další překážkou je složitost údržby. V první řadě se v prvních letech od nasazení modelu dá předpokládat nutnost doučení na vzorcích, které se statisticky objevují jen zřídka a nejsou v datasetu dostatečně zastoupeny. Dále se může objevit lehce odlišné provedení kontrolovaného výrobku, které by mohlo představovat rozsáhlé změny, jako například další kategorii defektů. Údržba by mohla znamenat i kompletní přetrénování modelu, přičemž dosažení původní přesnosti by pravděpodobně vyžadovalo značné množství pokusů. Ve výsledku je tedy tvorba a údržba takového modelu velmi neefektivní. Proto je při návrhu detekčního systému klíčové rozložení dílčích detekčních úloh mezi více modelů, což umožňuje i užší specializaci jednotlivých modelů na dané úlohy. Nevýhodou tohoto rozložení je pravděpodobně delší doba inspekce. Na druhou stranu užší specializace jednotlivých modelů umožňuje větší míru optimalizace jejich výpočetní náročnosti, než je tomu u zmíněného globálního modelu.

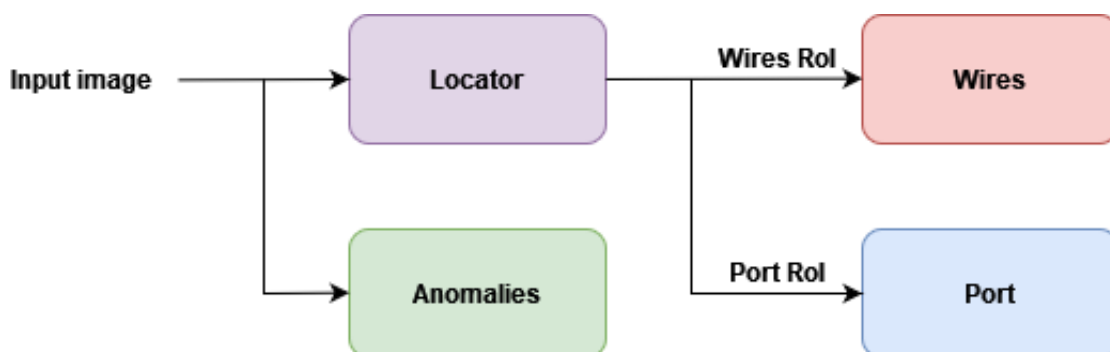
Vzhledem k této úvaze bylo prvním krokem při návrhu systému rozdělení kategorií defektů, uvedených v tab. 3.1, do skupin dle společných znaků. Toto rozdělení je uvedeno v tab. 4.1. Kategorie defektů skupiny A se týkají objektů, které by se měly na inspekční fotografii objevit vždy, protože jsou součástí zařízení, a které mají srovnatelnou vizuální variabilitu. Jejich dalším společným znakem je nutnost přesného měření parametrů vycházejících z jejich tvaru, pozice a natočení. Pro tuto skupinu by byla vhodnou volbou architektura provádějící instanční segmentaci a klasifikaci objektů. V rámci této práce bude model pro detekci defektů této skupiny označován názvem Locator, viz tab. 4.1. Oproti tomu kategorie skupiny B jsou

anomální jevy, které se vyskytují jen zřídka a které nevyžadují sofistikovaná měření. Architektura pro detekci a klasifikaci objektů je pro skupinu B dostačující. Model pro tuto skupinu bude dále označován názvem Anomalies, viz tab. 4.1. Zbývají skupiny AB a AC. Tyto skupiny se týkají součástí zařízení a měly by se tím pádem vyskytovat na každé inspekční fotografii, ale nevyžadují segmentaci na úrovni celé inspekční fotografie. Skupina AB zahrnuje pouze deformace bondovaných drátků, které mají vysokou prioritu detekce a jejichž vizuální variabilita je v porovnání s ostatními kategoriemi skupiny A mnohem vyšší. Z tohoto důvodu je vhodnější dedikovat této skupině samostatný model. Vzhledem k vizuálnímu charakteru defektů této skupiny je vhodnou volbou architektura pro detekci a klasifikaci, nebo pouze klasifikaci. Pro zúžení detekční oblasti tohoto modelu za účelem zjednodušení detekční úlohy lze použít informace z modelu Locator. Model této skupiny bude dále uváděn pod názvem Wires, viz tab. 4.1. Součástí skupiny AC je pouze absence gelu na portu, ale mohou se na něm vyskytovat i kontaminace ze skupiny B, které mohou narušit i bondované drátky. Model této skupiny tedy může být využit ke zlepšení celkové přesnosti detekce defektů skupiny AB a B. Pro detekci těchto defektů stačí klasifikační architektura v oblasti definované informacemi z modelu Locator. Model této skupiny bude označován názvem Port, viz tab. 4.1.

Tab. 4.1 – Přehled rozřazení kategorií defektů do skupin dle podobnosti

Kategorie defektů	Skupina	Model
Chybějící komponenty	A	Locator
Chyba pozice nebo natočení komponent	A	Locator
Přítomnost gelu na komponentech	A	Locator
Nadměrné množství nebo absence gelu na ASIC	A	Locator
Deformace bondovaných drátků	AB	Wires
Absence gelu na portu	AC	Port
Kontaminace vláknů	B	Anomalies
Nadměrné množství a kontaminace silikonem	B	Anomalies
ECA kontaminace	B	Anomalies
Kontaminace částicemi	B	Anomalies

Z rozřazení kategorií defektů vzniká základní struktura navrhovaného detekčního systému daná sekvencí modelů, která je znázorněna na obr. 4.1



Obr. 4.1 – Blokové schéma návaznosti modelů navrhovaného detekčního systému

4.1 MODELY

V rámci této kapitoly budou rozebrány jednotlivé modely navrhovaného detekčního systému. To zahrnuje volbu jejich konkrétní architektury a vymezení klasifikačních tříd. Dále bude obecně popsán proces tvorby datasetů a k tomu použitý software.

Při výběru architektur pro jednotlivé modely systému byly brány v úvahu jejich obecné vlastnosti popsané v oficiálních vědeckých publikacích, množství existujících implementací, dostupnost předtrénovaných vah a skóre v rámci relevantních benchmarků. Papers with Code (2024) poskytuje všechny tyto informace na jednom místě. Nejdůležitějším aspektem při výběru bylo množství existujících implementací, protože u takových architektur se dá předpokládat ověřená schopnost adaptace na široké spektrum úloh. Pro tyto architektury jsou navíc dostupné předtrénované modely a konfigurovatelné implementace, což výrazně usnadňuje jejich aplikaci na vlastní úlohu. Pro navrhovaný systém je důležitá hlavně schopnost odhalit defekty a nebyl explicitně zadán požadavek na rychlost. Předpokládá se ale rozumná doba predikce celého systému v řádu sekund na kus, což vybrané architektury splňují.

4.1.1 Locator

Účelem tohoto modelu je lokalizace a segmentace objektů, které by se měly nacházet na každé inspekční fotografii, a také objektů potřebných pro metody měření popsané v kapitole 4.2. Vzhledem k tomu, že přesnost měření je závislá hlavně na přesnosti segmentace, byl při výběru vhodné architektury kladen důraz právě na ni. Jako vhodný kandidát se ukázala

architektura Mask R-CNN s rozšířením PointRend, díky kterému dosahuje dostatečné přesnosti segmentace pro účely této úlohy.

Tuto variantu Mask R-CNN implementuje Detectron2. Mezi jeho tvůrce patří i vědci, kteří se podíleli na tvorbě PointRend rozšíření a Faster R-CNN. Proto se dá předpokládat vysoká kvalita konfigurovatelných implementací architektur, které Detectron2 nabízí. Pro Mask R-CNN PointRend jsou v Detectron2 dostupné váhy předtrénované na train2017 COCO datasetu s FPN a páteřními sítěmi ResNet50, ResNet101 a ResNeXt101. Pro model Locator byla nejprve zvolena varianta s ResNet50 s váhami trénovanými 37 epoch, což je standardní doba tréninku pro všechny plně předtrénované váhy dostupné v Detectron2. Varianta s ResNet50 byla zvolena, protože má oproti nejpřesnějšímu ResNeXt 101 pouze o zhruba 7 % nižší přesnost, ale ResNeXt je o přibližně o 160 % pomalejší (Wu, 2019). Zvolená varianta se pro zadanou detekční úlohu ukázala jako dostačující.

Tab. 4.2 obsahuje seznam tříd modelu Locator. Segmentace objektů tříd asic, capacitor a diode jsou využity k měření parametrů těchto objektů pomocí metod popsanych v kapitole 4.2. Účelem třídy port je nalezení přesné oblasti, ve které se nachází port. Tato oblast je pak vstupem do Port modelu. Do třídy plate patří plošky, jejichž poloha slouží jako reference pro vyhodnocení pozice a natočení komponent.

Tab. 4.2 – Třídy modelu Locator

Třída
asic
capacitor
diode
port
plate

4.1.2 Wires

Model Wires slouží k detekci deformací bondovaných drátků. Při navrhování vhodné metody detekce byly uvažovány dva přístupy. Prvním z nich je segmentace jednotlivých drátků a následné vyhodnocení jejich deformace na základě kontinuity a tvaru jejich segmentačních masek. Výhodou tohoto přístupu je robustnost klasifikace v případě, že je segmentace přesná. Problémem je vysoká variabilita tvaru drátků, která vyžaduje ošetření řady krajních případů, což je časově náročný proces, a časově náročná tvorba kvalitních anotací pro takový model. Druhým přístupem je klasifikace obrazu bondovaných drátků. Tento přístup je v porovnání s předchozím výrazně jednodušší a rychlejší na implementaci, přičemž při dostatečném odladění by měl dosáhnout podobné přesnosti. Nevýhodou je určitá pravděpodobnost neodhalení jemnějších deformací, jelikož žádný klasifikační model není z principu jeho fungování stoprocentně přesný. Vzhledem k časové náročnosti prvního přístupu byl zvolen přístup druhý, klasifikace obrazu bondovaných drátků. Přičemž v případě nedostatečných výsledků je tu v budoucnu možnost kombinace s prvním přístupem.

ASIC je připojován bondovanými drátky z obou stran a drátky na těchto stranách jsou vizuálně odlišné. Proto by byly vhodné dva klasifikační modely, za účelem usnadnění optimalizace a pravděpodobné vyšší výsledné přesnosti. Detekce by sice byla pomalejší, ale vzhledem k rychlosti klasifikačních architektur by to v této úloze nebyl problém. Při deformaci bondovaných drátků se ale může stát, že se jeden z nich ohne mimo standardní vyřezávanou oblast, zatímco ostatní drátky zůstanou s jemnou deformací. Pokud by vyříznutá oblast měla vždy fixní velikost, ohnutý drátek by se v ní nenacházel a vzhledem k jemné deformaci zbytku drátků by se mohlo stát, že je klasifikátor označí jako OK. Proto je nutná adaptace velikosti vyřezávané oblasti. Klasifikační konvoluční neuronové sítě vyžadují vstup fixních rozměrů. V případě obrazových dat různých rozměrů je tedy nutná jejich transformace. Buď se obraz na požadované rozměry dostane změnou poměru stran a velikosti, nebo je poměr stran zachován přidáním černých pruhů (tzv. padding) a až poté je změněna velikost jeho rozměrů. Pokud by ale nebyl zachován poměr stran obrazu bondovaných drátků, jejich korektní klasifikace by byla nemožná. Ve výsledku je tedy nutná adaptace vyřezávaných oblastí na tvar drátků a padding před vstupem do modelu. Zmíněné adaptace by se ale muselo dosáhnout jiným modelem pro detekci objektů.

Na základě této úvahy byla tedy pro detekci deformací bondovaných drátků zvolena architektura pro detekci a klasifikaci objektů, protože provádí klasifikaci objektů v oblastech velikosti adaptované velikosti těchto objektů. Obraz v rámci této architektury nejprve prochází

procesem detekce objektů, při kterém mohou být už předem vyřazeny silné deformace bondovaných drátků, a až poté samotnou klasifikací. Je tedy do jisté míry zaručena detekce silných deformací. Alternativou v případě nedostatečné přesnosti klasifikace tohoto modelu je její nahrazení dvojicí klasifikačních modelů, jak již bylo zmíněno.

Detectron2 kromě architektury zvolené pro model Locator nabízí i implementace architektur pro detekci a klasifikaci objektů. Pro model Wires byla vybrána architektura Faster R-CNN s FPN a páteří sítí ResNet, protože v porovnání s ostatními architekturami stejného typu implementovanými v Detectron2 nabízí nejlepší poměr mezi rychlostí a přesností. Konkrétně byla zvolena páteří síť ResNet50 ze stejného důvodu. Model Wires byl trénován z vah předtrénovaných 37 epoch na train2017 COCO datasetu (Wu, 2019).

Tab. 4.4 obsahuje seznam tříd modelu Wires. Třídy OK_port_wires a NOK_port_wires klasifikují drátky na straně portu a třídy OK_side_wires a NOK_side_wires na opačné straně.

Tab. 4.3 – Třídy modelu Wires

Třída
OK_port_wires
NOK_port_wires
OK_side_wires
NOK_side_wires

4.1.3 Port

Hlavním účelem modelu Port je detekce absence gelu na portu. Sekundárním účelem je detekce výraznějších kontaminací v jeho oblasti. Tím slouží jako záchranná vrstva pro odchytní kontaminací v oblasti bondovaných drátků, pokud by prošly modely Wires a Anomalies bez detekce.

Pro zmíněné detekční úlohy je ideální klasifikační síť. Absence gelu na portu se projevuje zvýšenou odrazivostí jeho povrchu a port je tak na inspekční fotografii mnohem světlejší, dokonce až bílý. Jedná se tedy o poměrně jednoduchý klasifikační úkol, který by šel řešit i prahováním. Ale výhodou klasifikační sítě je schopnost klasifikovat i na základě jiných vlastností obrazu, jako jsou například zmíněné kontaminace. Model Port tedy funguje i jako záchranná vrstva, jak již bylo zmíněno, čímž zpřesňuje celý systém.

Pro Port model byla zvolena architektura EfficientNet, která je jednou z nejčastěji implementovaných klasifikačních CNN architektur. Mezi populárnější CNN architektury patří například ResNet a MobileNetV2, ale EfficientNet je přesnější (Papers with Code, 2024). Konkrétně byla zvolena verze B4 primárně kvůli tomu, že rozlišení vyříznuté oblasti portu je blízké vstupnímu rozlišení této verze. Pokud by byla zvolena verze s nižším vstupním rozlišením, model by byl rychlejší, ale zhoršila by se jeho schopnost detekovat kontaminace. Na druhou stranu verze s vyšším rozlišením, a tedy s větší šířkou a hloubkou, by připadaly v úvahu v případě nedostatečné přesnosti verze B04.

EfficientNet je implementován v pycls, kde jsou k dispozici i váhy předtrénované 100 epoch na ImageNet datasetu pro verze B0 až B5. Na tvorbě pycls se podíleli vědci, kteří jsou spoluautory Mask R-CNN a FPN, dá se tedy předpokládat, že konfigurovatelné implementace architektur budou kvalitní. Navíc je platforma pycls postavena na frameworku PyTorch stejně jako Detectron2, což usnadňuje tvorbu software pro aplikaci na vlastní úloze.

Model provádí pouze binární klasifikaci, kterou určuje, jestli je vstupní obraz OK nebo NOK. Pro úplnost jsou tyto třídy uvedeny v tab. 4.6.

Tab. 4.4 – Třídy modelu Port

Třída
OK
NOK

4.1.4 Anomalies

Model Anomalies má detekovat kontaminace, tedy obecně objekty, které nejsou součástí kontrolovaného zařízení a objevují se s nižší frekvencí. Konkrétně se jedná o vlákna, silikon, ECA a částice. Nezáleží přitom příliš na přesnosti ohraničení oblastí, ve kterých se objekty nachází. Důležitější je takové objekty nalézt.

Jelikož se jedná o v podstatě stejnou úlohu jako u modelu Wires, byla pro model Anomalies zvolena stejná architektura, Faster R-CNN s FPN a páteřní síť ResNet50, a tím pádem i stejné předtrénované váhy. Třídy modelu Anomalies jsou uvedeny v tab. 4.8.

Tab. 4.5 – Třídy modelu Anomalies

Třída
fiber
silicon
eca
particles

4.1.5 Tvorba datasetů

Trénovací dataset je nejdůležitějším aspektem ovlivňujícím přesnost trénovaného modelu na neznámých datech. Při jeho tvorbě je nutné klást důraz na jeho rozmanitost a zachycení veškerých situací, se kterými by se mohl model setkat. Evaluační dataset by měl být stejně rozmanitý jako ten trénovací, aby bylo možné odhadnout kvalitu modelu (Gong, 2023). Datasetsy pro jednotlivé modely byly tvořeny postupně na základě chování konkrétních modelů na testovacích sadách.

Trénovací a evaluační dataset je někdy získáván rozdělením jednotné náhodně promíchané množiny dat určitým poměrem. Tento přístup se může hodit pro získání odhadu o schopnostech konkrétní architektury, ale pro trénování modelů s cílem vysoké přesnosti na konkrétních úlohách je naprosto nevhodný. Každý dataset je možné rozdělit na určité podskupiny vizuálně podobných vzorků. Jak již bylo zmíněno, rozmanitost datasetu je velmi důležitá. Měl by tedy obsahovat dostatečné zastoupení každé podskupiny. Vychází-li ale z části náhodně promíchané množiny, je pravděpodobné, že některé podskupiny nebudou dostatečně zastoupeny a model se je zkrátka nenaučí.

Modely Locator, Wires a Anomalies potřebují k tréninku kromě obrazových dat také anotace obsahující referenční výstupy pro jednotlivé obrazy. K tvorbě těchto anotací byl využit software Label Studio.

Label Studio je open-source nástroj pro anotaci dat ve formě webové aplikace, který umožňuje centrální správu dílčích anotačních projektů, sdílení mezi uživateli a jejich vzájemnou spolupráci. Aktuálně je možné tímto nástrojem vytvářet anotace pro modely zpracovávající obraz, zvuk, text, grafy, ale i anotace pro specifické úlohy jako hodnocení webové stránky nebo klasifikace videa. Label Studio umožňuje také integraci modelů přímo do jeho prostředí, díky čemuž je možné spouštět trénink modelů přímo z prostředí Label Studia

nebo automaticky vytvářet anotace. Také podporuje export anotací v různých formátech (HumanSignal, 2024).

Label Studio bylo k dispozici ve firmě Vitesco Technologies. Pro realizaci praktické části této práce tedy nemuselo být instalováno. Byla provedena pouze úprava rozložení uživatelského prostředí prostřednictvím konfiguračního souboru.

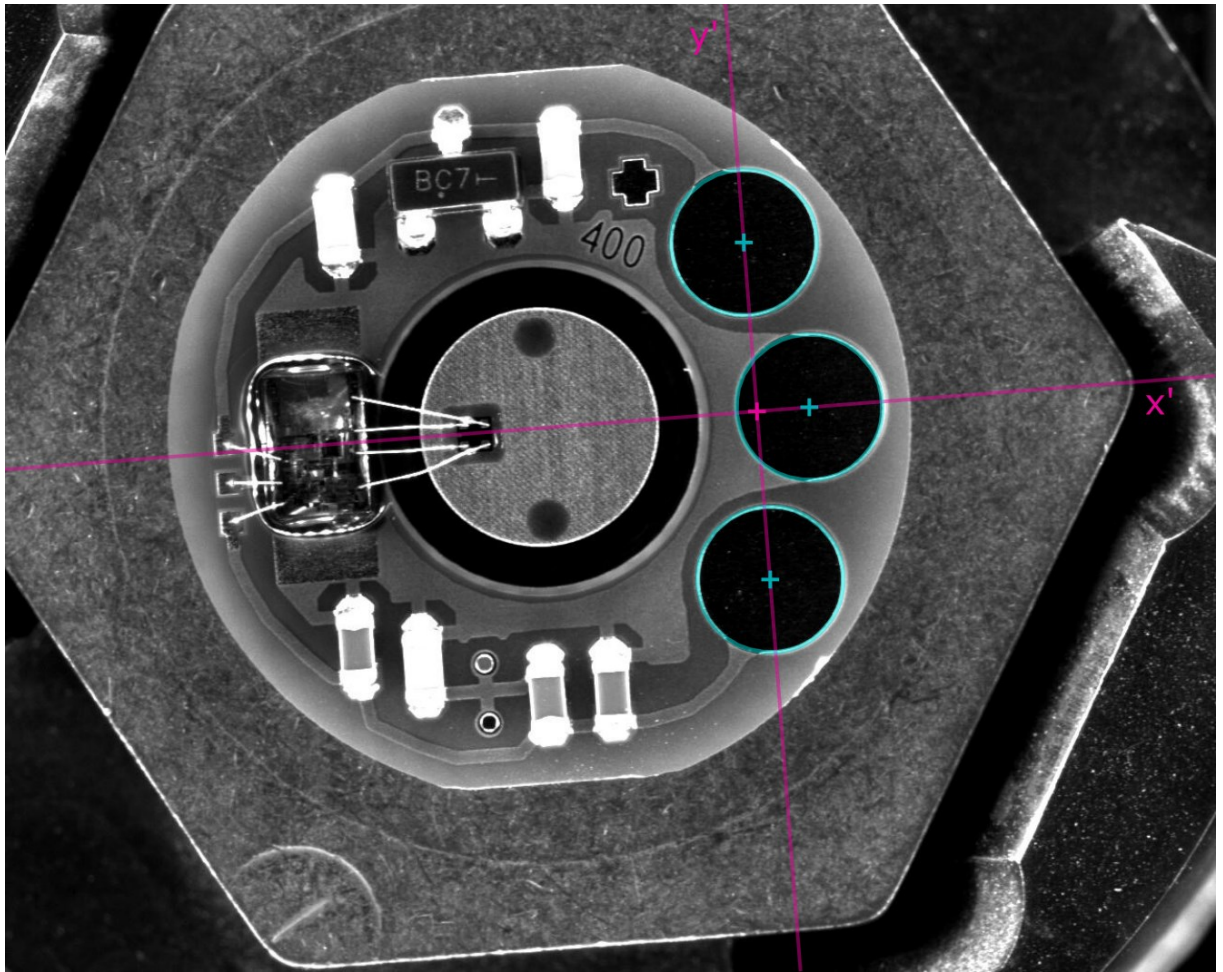
4.2 NAVRŽENÉ METODY MĚŘENÍ

Segmentační masky lze dále zpracovávat a měřit jejich různé parametry, čímž se přeneseně měří parametry segmentovaného objektu. Na základě hodnot měřených parametrů je možné segmentované objekty velmi přesně klasifikovat, ale také statisticky regulovat výrobní proces. Segmentační maska vystupující z modelu Locator implementovaného prostřednictvím Detectron2, je ve formátu sekvence bodů ohraničujících segmentovaný objekt.

4.2.1 Měření pozice komponent

Při měření pozice komponent je nejprve nutné stanovit referenční bod, vůči kterému bude pozice měřena. Standardně je tímto bodem levý horní roh vstupního obrazu, kde je počátek souřadného systému jeho pixelů. Použití tohoto bodu jako reference není v případě této úlohy vhodné, protože není jisté, že PCB bude na inspekční fotografii vždy ve stejné pozici. Proto je nutné zvolit referenční bod přímo na PCB, na základě pozice určitých stálých prvků PCB, které nemění pozici. Pro tento účel byla zvolena trojice plošek, jejichž přesná pozice je detekována modelem Locator. Ten je schopen velmi přesné segmentace těchto plošek, protože se jedná o jednoduché kruhové tvary, jejichž vzhled se příliš nemění. Referenční bod pro měření pozice se nachází přesně mezi dvojicí nejvzdálenějších plošek, jak znázorňuje obr. 4.2.

Pozice komponent je měřena jako odchylka od referenční pozice. Referenční pozice komponent jsou uloženy v konfiguračním souboru `pcb_parameters.yaml` ve formě bodů, jejichž souřadnice mají počátek v referenčním bodě. Jsou tedy součástí souřadnicového systému PCB, které může být na inspekční fotografii různě natočeno. Pozice komponentu je definována bodem v jeho středu. Aby bylo možné tento bod porovnat s referenční pozicí, musí se souřadnice referenční pozice transformovat ze souřadnicového systému PCB do souřadnicového systému inspekční fotografie. Odchylka středového bodu komponentu od referenční pozice je vypočtena jako vzdálenost mezi těmito dvěma body.



Obr. 4.2 – Vypočtené osy souřadnicového systému PCB

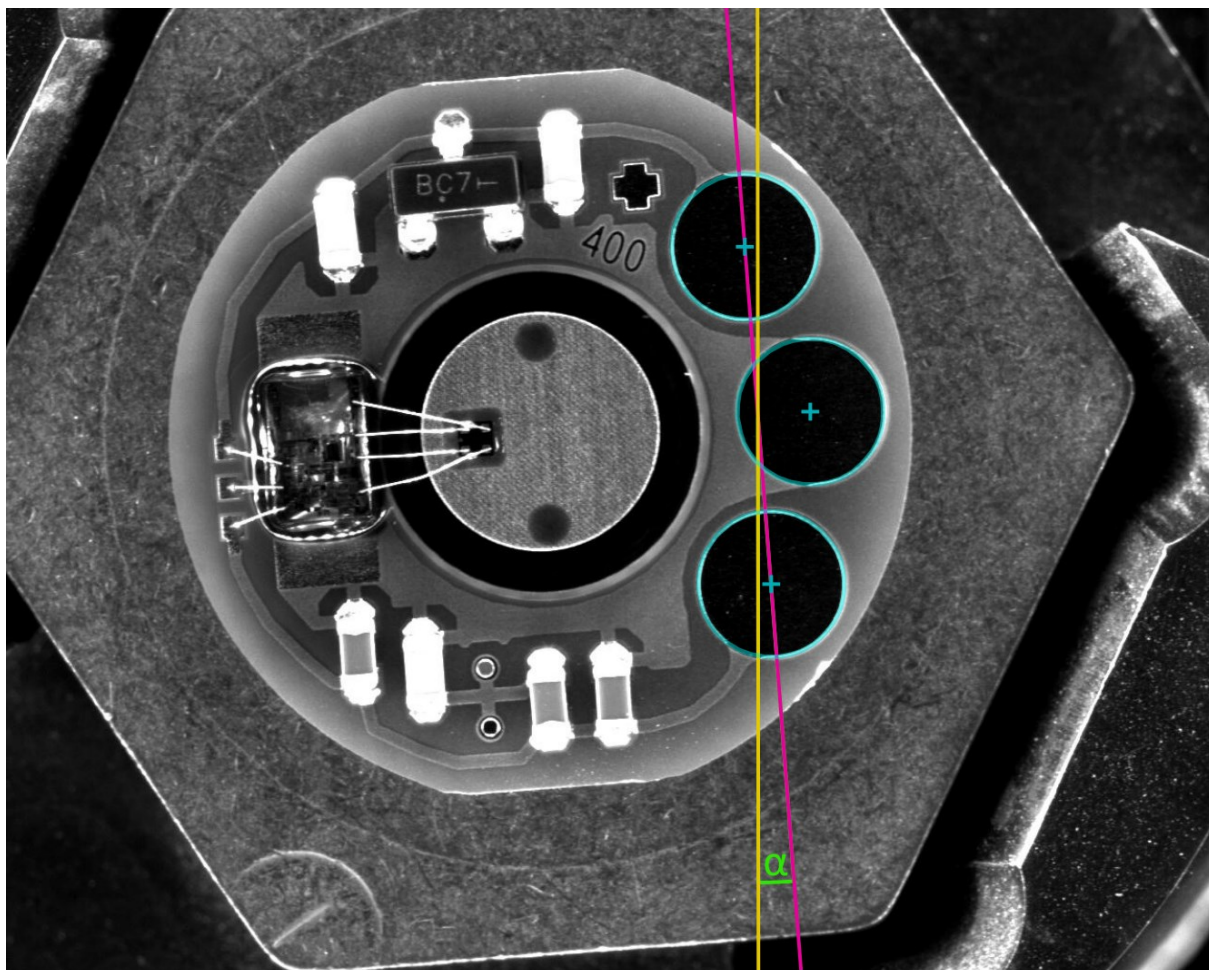
4.2.2 Měření rotace komponent

Podobně jako u měření pozice komponent, je pro měření rotace komponent potřeba referenční úhel, což je úhel natočení celého PCB. K výpočtu tohoto úhlu je opět využito trojice plošek. Nejprve je vypočten úhel natočení přímky procházející středy krajních plošek vůči svislé hraně inspekční fotografie, jak je znázorněno na obr. 4.3. Na základě tohoto úhlu je stanoveno ideální natočení jednotlivých komponent. Tedy pro kondenzátory je toto natočení rovno natočení přímky a pro diodu je toto natočení rovno natočení kolmice na tuto přímku.

Při návrhu metody měření úhlu natočení segmentační masky komponentu byly uvažovány dva přístupy. Jedním z nich je lineární regrese, tedy proložení množiny bodů ohraničujících segmentační masku přímkou (Weisberg, 2013). Natočení této přímky je rovno úhlové odchylce od vodorovné hrany inspekční fotografie. Druhým přístupem je umístění obdélníku na polygon tvořený sekvencí bodů ohraničujících segmentační masku tak, aby měl minimální plochu a celý polygon byl v něm. K tomuto je možné využít funkci `minAreaRect`

knihovny OpenCV. Tato funkce vrací rozměry a úhel natočení umístěného obdélníku (OpenCV, 2024). Jeho úhel natočení pak odpovídá úhlu natočení segmentovaného objektu.

Oba přístupy fungují pouze v případě podlouhlého tvaru segmentační masky. Čím blíže je maska tvaru kruhu, tím více se zvyšuje nejistota jejího natočení, protože pro masku tvaru kruhu existuje nekonečné množství směrů natočení neboli neexistuje pro ni směr natočení.



Obr. 4.3 – Vypočtený úhel natočení PCB

První přístup by byl přesnější v případě tvaru masky připomínajícího hroznýše, jak zažívá slona, známé kresby z literárního díla Malý Princ (Saint-Exupéry, 2014). Protože úhel natočení obdélníku získaného druhým přístupem by byl silně závislý na tvaru tenkých protáhlých částí masky. Na druhou stranu pro tvary masky blízké obdélníku je přesnější druhý přístup. Jelikož kontrolované komponenty se svým tvarem blíží spíše obdélníku, byl nakonec zvolen přístup druhý. Finální hodnoty parametrů jsou vypočteny jako odchylky natočení komponent od jejich ideálního natočení.

4.2.3 Měření tvaru komponent

Tvar segmentační masky je možné vyhodnotit nespočtem způsobů. Zkoumané parametry související s tvarem vychází z konkrétní úlohy. Parametry mohou být jednoduché, jako šířka, výška či nejužší místo, ale i složitější zkoumající konvexnost nebo deformaci vzhledem k referenčnímu tvaru.

Účelem kontroly tvaru komponent je především odhalení přítomnosti gelu. Model Locator je trénovaný k segmentaci komponent včetně případného gelu. Segmentační maska daného komponentu je tedy v případě přítomnosti gelu mnohem větší a její původně obdélníkový tvar je pak eliptický. V extrémním případě může mít maska kruhový tvar. Z toho vyplývá, že nejjednodušším způsobem, jak tento defekt odhalit, je měřit kratší z rozměrů (šířka a výška) komponent, protože delší z nich se při defektu mění méně. Tento rozměr lze získat z obdélníku použitého k výpočtu rotace komponent, jak bylo popsáno v předchozí kapitole 4.2.2.

4.2.4 Měření gelu na ASIC

Při aplikaci ochranného gelu na ASIC se jeho segmentační maska zvětší do všech stran přibližně rovnoměrně. Je-li gelu příliš velké množství, rozlije se směrem k bondovaným drátkům, což bude detekováno modelem Wires. Jde tedy hlavně o měření za účelem detekce nedostatečného množství nebo úplné absence gelu. Vzhledem k tomu, že segmentační maska se zvětšuje rovnoměrně, byla jako měřený parametr zvolena její plocha.

5 SOFTWARE PRO TRÉNOVÁNÍ A TESTOVÁNÍ SYSTÉMU

V rámci této kapitoly bude popsána základní struktura vytvořeného softwaru. Kompletní okomentovaný kód je součástí přílohy A této práce. Software byl psán v jazyce Python, protože jsou v něm psány i obě použité platformy Detectron2 a pycls. Obě platformy jsou postaveny na PyTorch frameworku. Vytvořený software zahrnuje prostředky pro testování celého navrženého systému a prostředky pro trénování modelů. Z platformy pycls byl extrahován relevantní kód, který byl následně modifikován tak, aby bylo možné trénovat a testovat modely na vlastních datech. Na obr. 5.1 je stromové schéma repozitáře obsahujícího veškerý software.



Obr. 5.1 – Stromové schéma repozitáře vytvořeného softwaru

Software byl vytvářen v izolovaném prostředí spravovaném nástrojem Anaconda. Pomocí souboru `anaconda_environment.txt` je možné vytvořit identické prostředí. Soubor `requirements.txt` obsahuje seznam použitých knihoven a jejich verze.

5.1 TRÉNOVÁNÍ MODELŮ

Modely, jejichž implementace pochází z platformy Detectron2, jsou trénovány prostřednictvím vytvořené třídy `Detectron2Trainer`, která zprostředkovává inicializaci veškerých konfiguračních parametrů tréninku a spouští trénink. Trénink probíhá za pomoci nástrojů z Detectron2. Byla implementována také možnost vizualizace trénovacího datasetu a možnost průběžné evaluace modelů s protokolováním výsledků.

Pro trénink konkrétního modelu je vždy vytvořena třída dědicí z třídy `Detectron2Trainer`, která modifikací metody `custom_cfg_init` nastavuje vlastní parametry tréninku. Výhodou tohoto přístupu je možnost modifikace dalších metod.

Modely implementovány v modifikované verzi `pycls` se trénují prostřednictvím skriptu `run_net.py`, ze kterého je možné i jejich testování. Parametry tréninku i trénovaného modelu vychází z konfiguračního souboru, k němuž se zadává cesta. Augmentace obrazových dat pro trénink i testování se mění dynamickým nahrazením metody `_prepare_im` vytvořené třídy `CustomDataset` modifikovanou metodou `__prepare_im`, která je definována ve skriptu `run_net.py`.

5.2 TESTOVÁNÍ NAVRŽENÉHO SYSTÉMU

Test celého systému se spouští ze souboru `run.py`, ve kterém se nachází i globální konfigurace. Na obrázku 5.2 je kód obsažený v tomto souboru. V enumerátoru `Models` se nastavují cesty k modelům a ve třídě `Config` se nastavují parametry systému. `Config` zahrnuje nastavení cesty ke vstupním datům včetně možnosti filtrování na základě názvu souboru a nastavení cest k souborům obsahujícím limity pro měřené parametry a referenční hodnoty pozic komponent. Dále je možné nastavit, které modely systému budou aktivní a prahy pro skóre výstupu. Do systému byl implementován systém tzv. statusů, které umožňují monitorovat průběh detekce a filtrovat výstupy na základě výsledků. Dá se využít i ke sledování chyb. V průběhu detekce se tyto statusy generují na relevantních místech v kódu. Každý status je přidán do seznamu, který se resetuje s každým novým vstupem do systému. Jednotlivé statusy jsou definovány v enumerátoru `AOIRunStatus` a je možné poměrně rychle přidat nové. Ve třídě `Config` lze aktivovat a nastavit filtrování na základě těchto statusů.

```

from enum import Enum
import aoi as _

class Models(Enum):
    """
    Stores paths to trained models.
    Framework prefixes:
    | D2 = detectron2
    | PYCLS = pycls
    """
    D2_LOCATOR = "path\\to\\model\\LOCATOR"
    D2_WIRES = "path\\to\\model\\WIRES"
    PYCLS_PORT = "path\\to\\model\\PORT"
    D2_ANOMALIES = "path\\to\\model\\ANOMALIES"

class Config:
    """
    Global configuration for this AOI system.
    """
    def __init__(self):
        self.INPUT_IMGS_PATH = "path\\to\\folder\\with\\input\\images"
        self.AOI_OUTPUT_PATH = "path\\to\\output\\folder"
        self.FILE_WILDCARD_FILTER = "*.jpg"
        self.LIMITS_YAML_PATH = ".\\limits.yaml"
        self.PCB_PARAMETERS_YAML_PATH = ".\\pcb_parameters.yaml"

        self.MODELS_ENABLED = {
            "D2_LOCATOR": True,
            "D2_WIRES": True,
            "PYCLS_PORT": True,
            "D2_ANOMALIES": True
        }

        self.SCORE_THRESHOLDS = {
            "D2_LOCATOR": 0.6,
            "D2_WIRES": 0.6,
            "PYCLS_PORT": None,
            "D2_ANOMALIES": 0.6
        }

        self.STATUS_FILTER_ENABLED = True
        self.STATUS_FILTER = [
            _AOIRunStatus.LOCATOR_MISSING_CAPACITORS,
            _AOIRunStatus.WIRES_NOK_PORT_WIRES,
            _AOIRunStatus.ANOMALIES_FOUND_FIBER
        ]
        self.STATUS_FILTER_MATCHED_OUTPUT_FOLDER = "correct"
        self.STATUS_FILTER_UNMATCHED_OUTPUT_FOLDER = "wrong"

if __name__ == '__main__':
    _AOI().run()

```

Obr. 5.2 – Kód pro konfiguraci a spuštění celého systému

Základní proces systému je implementován třídou AOI. Tato třída se inicializuje na základě inicializačních vlastností třídy Config. Po spuštění postupně prochází obrazová data ze vstupní složky. Pro každý vstup provede inferenci jednotlivých modelů a dodatečná měření metodami popsány v rámci kapitoly 4.2. Výsledky jsou vizualizovány a vizualizace uložena do výstupní složky. Na obr. 5.3 je ukázka této vizualizace výsledků. Naměřené parametry všech vstupů jsou po projití všech dat uloženy do JSON souboru, který je možné využít například pro následnou statistickou analýzu. Hodnoty limitů jednotlivých parametrů jsou pouze demonstrační. Stanovení vhodných limitů by vyžadovalo statistickou analýzu na velkém množství dat, což není součástí této práce.



Obr. 5.3 – Vizualizace výsledků vytvořeným softwarem

6 HODNOCENÍ MODELŮ

Obsahem této kapitoly jsou výsledky testování jednotlivých modelů systému a jejich hodnocení. Jednotlivé testy jsou zaměřené na specifické schopnosti modelu, které jsou relevantní pro úlohu, kterou mají v rámci systému plnit. Každý model má vlastní testovací dataset sestavený tak, aby obsahoval relevantní inspekční fotografie.

6.1 LOCATOR

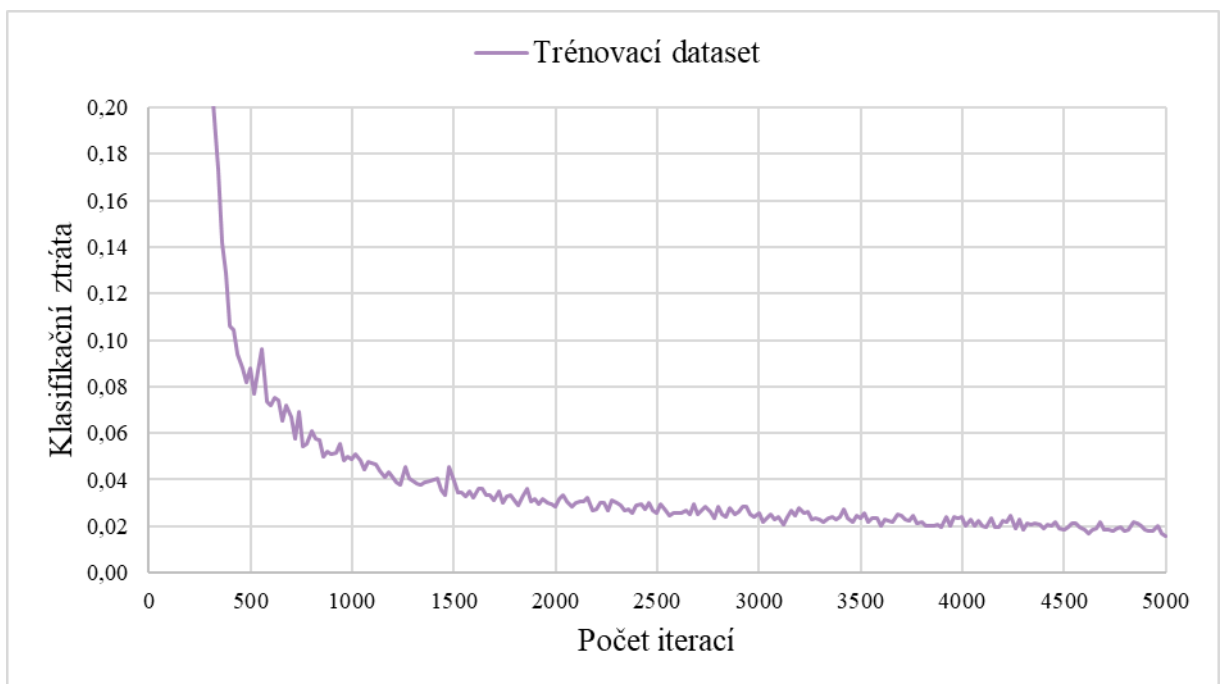
V tab. 6.1 jsou uvedeny základní konfigurované parametry modelu Locator. Všechny parametry jsou dostupné v příloze A této práce. Trénovací dataset modelu je tvořen 55 fotografiemi a testovací dataset 3820 fotografiemi.

Tab. 6.1 – Základní konfigurované parametry modelu Locator

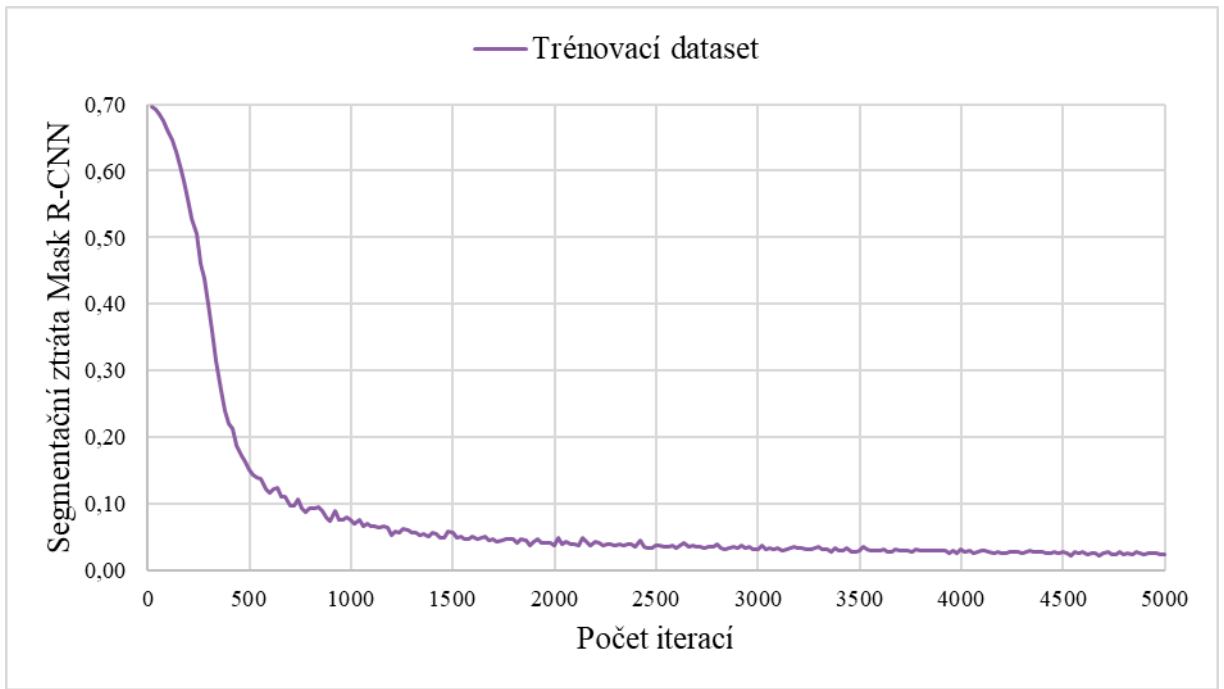
Parametr	Hodnota
SOLVER.IMS_PER_BATCH	1
SOLVER.BASE_LR	0,00025
SOLVER.MOMENTUM	0,9
SOLVER.CLIP_GRADIENTS.ENABLED	False
SOLVER.MAX_ITER	5000
INPUT.MAX_SIZE_TEST	1280
INPUT.MAX_SIZE_TRAIN	1280
INPUT.MIN_SIZE_TEST	1024
INPUT.MIN_SIZE_TRAIN	1024
INPUT.RANDOM_FLIP	horizontal
MODEL.BACKBONE.FREEZE_AT	2
MODEL.ANCHOR_GENERATOR.ASPECT_RATIOS	0,5; 1,0; 2,0
MODEL.ANCHOR_GENERATOR.SIZES	32; 64; 128; 256; 512
MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE	512
MODEL.POINT_HEAD.FC_DIM	1024

Model byl trénován s optimalizační metodou Stochastic Gradient Descent (SGD), která je výchozí metodou pro všechny architektury v Detectron2 a se kterou byly trénovány i předtrénované váhy. Chyba klasifikace je vypočtena křížovou entropií, rozměrově-poziční chyba boxů je vypočtena metodou Smooth L1 a chyba segmentace je pro standardní Mask R-CNN masku i PointRend masku vypočtena binární křížovou entropií. Jedinou augmentací aplikovanou na trénovací data bylo horizontální překlopení. Ostatní augmentace nebyly vhodné, protože deformovaly obraz do stavu, se kterým by se model reálně nikdy neseťkal.

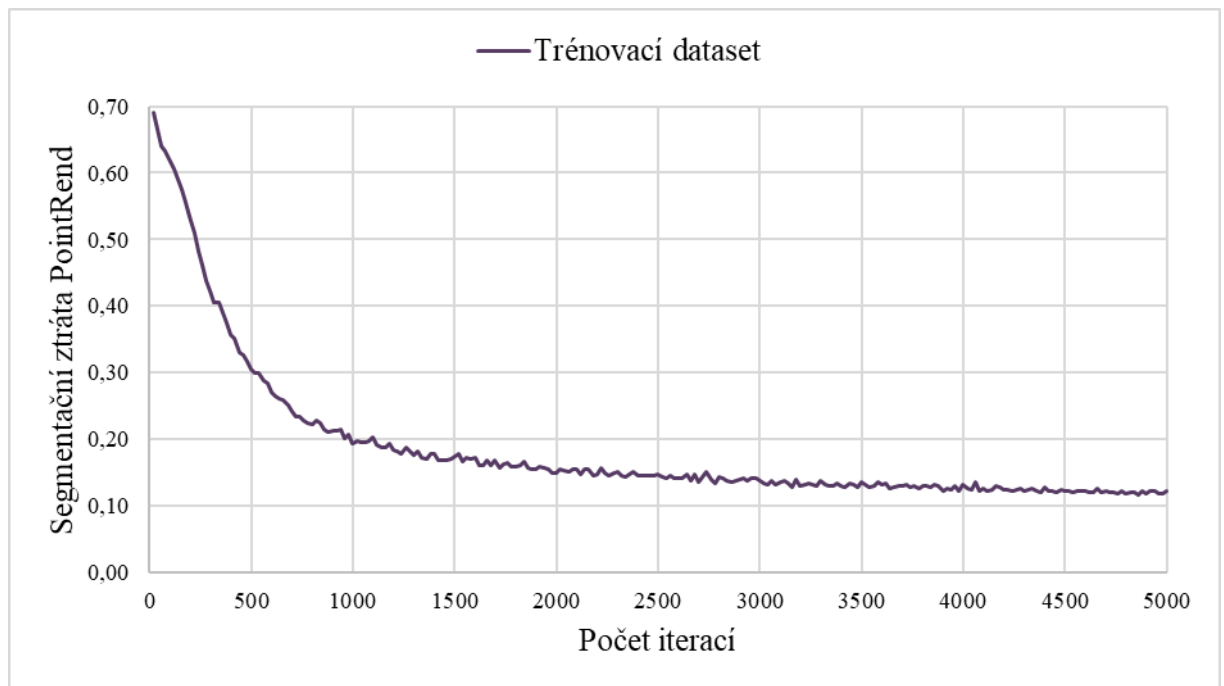
Na obr. 6.1 je graf průběhu klasifikační ztráty na trénovacím datasetu. Z grafů na obr. 6.2 a 6.3 je zřejmé, že model dosahuje poměrně vysoké přesnosti segmentace, což je vidět i na výstupu z modelu na obr. 6.4. Pro účely přesné lokalizace jednotlivých prvků PCB a měření jejich parametrů se tedy zdá být dostačující. V budoucnu, až bude systém nasazen do výroby, bude na místě jemnější optimalizace na základě konkrétních nedostatků vycházejících z měření.



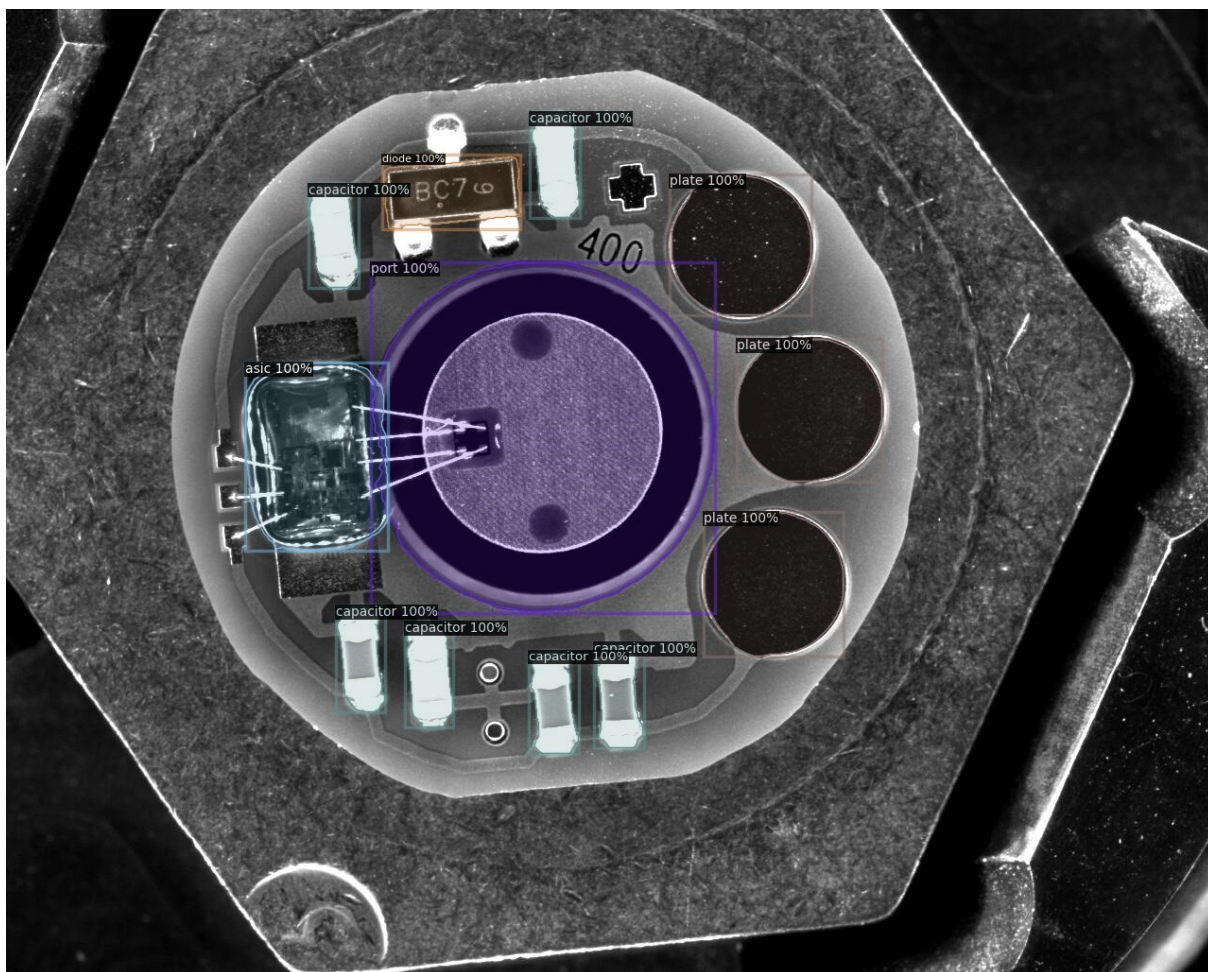
Obr. 6.1 – Průběh klasifikační ztráty modelu Locator během tréninku



Obr. 6.2 – Průběh segmentační ztráty Mask R-CNN modelu Locator během tréninku



Obr. 6.3 – Průběh segmentační ztráty PointRend modelu Locator během tréninku



Obr. 6.4 – Vizualizace výstupu modelu Locator

6.1.1 Test nalezení všech objektů

Model Locator musí korektně nalézt všechny důležité prvky PCB, protože nenalezení některého z nich by znamenalo chybnou klasifikaci celé inspekční fotografie. Na základě lokalizace těchto prvků jsou postaveny také metody měření. Dále jsou informace o lokaci ASIC a portu potřebné pro vyříznutí oblasti, která dále putuje do modelu Wires a Port. Z těchto důvodů je prováděn test nalezení všech objektů. Jelikož nenalezení správného počtu komponent znamená chybnou klasifikaci celé inspekční fotografie, byl takto vyhodnocován i tento test. Výsledky testu jsou uvedeny v tab. 6.2. Sloupec Detekováno ukazuje na kolika procentech inspekčních fotografií byl detekován správný počet instancí dané třídy a sloupec Falešné detekce zase ukazuje procento fotografií s falešnou detekcí.

Tab. 6.2 – Výsledek testu nalezení všech objektů pro model Locator

Třída	Detekováno, %	Falešné detekce, %
asic	100	0,03
capacitor	99,90	0,18
diode	99,97	0,05
port	100	0
plate	100	3,04

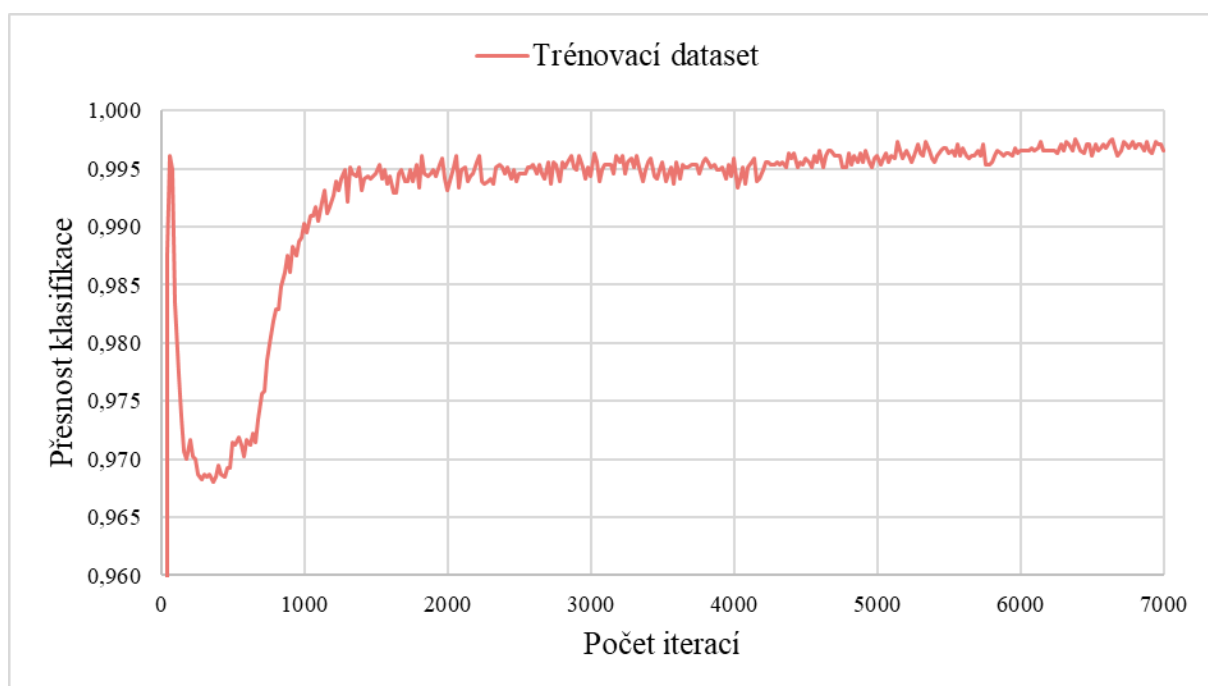
Z výsledků testu je zřejmé, že model Locator je schopen detekovat všechny objekty u téměř 100 % inspekčních fotografií testovacího datasetu, což je pozoruhodné. Hůře na tom je s falešnými detekcemi, kde zhruba u 3 % inspekčních fotografií falešně detekoval třídu plate. Tento nedostatek ovšem půjde vyřešit rozšířením trénovacího datasetu. Falešných detekcí u ostatních tříd je minimum. Model, s výjimkou třídy plate, dosahuje dostatečné přesnosti pro zkušební nasazení ve výrobě a následnou optimalizaci.

6.2 WIRES

V tab. 6.5 jsou uvedeny základní konfigurované parametry modelu Wires. Všechny parametry jsou dostupné v příloze A této práce. Trénovací dataset modelu je tvořen 645 fotografiemi a testovací dataset 3365 fotografiemi. Model byl trénován s optimalizační metodou SGD, která je výchozí metodou pro všechny architektury v Detectron2 a se kterou byly trénovány také předtrénované váhy. Chyba klasifikace je vypočtena křížovou entropií a rozměrově-poziční chyba boxů je vypočtena metodou Smooth L1. Na trénovací data nebyly, kromě změny velikosti na vstupní velikost modelu, aplikovány žádné augmentace. V úvahu by připadalo překlopení nebo rotace, což by pomohlo s učením lokalizace objektů. Model ale i bez augmentací neměl výraznější problémy s lokalizací objektů. Na obr. 6.5 je průběh přesnosti klasifikace objektů na trénovacím datasetu během tréninku.

Tab. 6.5 – Základní konfigurované parametry modelu Wires

Parametr	Hodnota
SOLVER.IMS_PER_BATCH	4
SOLVER.BASE_LR	0,00025
SOLVER.MOMENTUM	0,9
SOLVER.CLIP_GRADIENTS.ENABLED	False
SOLVER.MAX_ITER	7000
INPUT.MAX_SIZE_TEST	400
INPUT.MAX_SIZE_TRAIN	400
INPUT.MIN_SIZE_TEST	400
INPUT.MIN_SIZE_TRAIN	400
INPUT.RANDOM_FLIP	none
MODEL.BACKBONE.FREEZE_AT	2
MODEL.ANCHOR_GENERATOR.ASPECT_RATIOS	0,25; 0,5; 1,0; 2,0; 4,0
MODEL.ANCHOR_GENERATOR.SIZES	16; 32; 64; 128; 256



Obr. 6.5 – Průběh přesnosti klasifikace objektů modelu Wires během tréninku

6.2.1 Test přesnosti klasifikace

Účelem modelu Wires je klasifikace drátků. Nalezení jejich přesné lokace má sice vliv i na přesnost klasifikace, ale pokud je uvnitř lokalizované oblasti většina objektu, není tento vliv významný. Na obr 6.6 je vizualizace výstupu z modelu na vstupní fotografii. Z těchto důvodů byl proveden důkladný test přesnosti klasifikace a přesnost lokalizace byla vyhodnocena pouze osobní kontrolou výstupů modelu. Výsledky testu byly zpracovány do matice záměn, která je uvedena v tab. 6.6. Tab. 6.7 uvádí procentuální zastoupení správně predikovaných tříd.

Tab. 6.6 – Matice záměn z výsledků testu přesnosti klasifikace modelu Wires

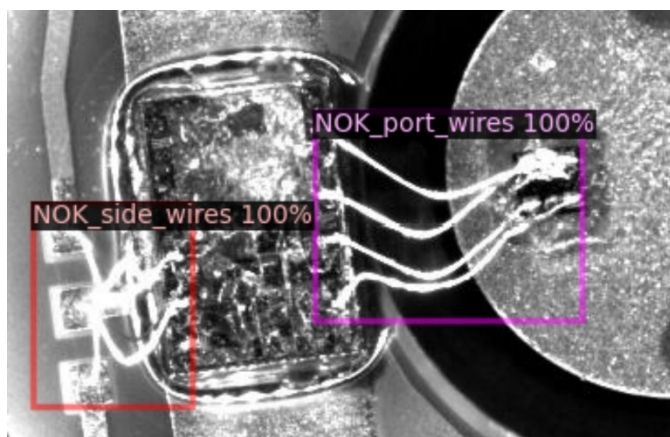
	OK_port_wires	NOK_port_wires	OK_side_wires	NOK_side_wires	Nedetkováno
OK_port_wires	578	4	0	0	0
NOK_port_wires	102	2464	0	0	0
OK_side_wires	0	0	2070	167	4
NOK_side_wires	0	0	31	972	0

Tab. 6.7 – Procentuální přesnosti klasifikace jednotlivých tříd modelu Wires

Třída	True Positive, %
OK_port_wires	99,31
NOK_port_wires.	96,02
OK_side_wires.	92,55
NOK_side_wires.	96,91

Z výsledků testu je zřejmé, že model zaostává hlavně v klasifikaci třídy OK_side_wires, u které dokonce 4 z 2241 instancí nebyly detekovány vůbec. Klasifikace tříd NOK_port_wires a NOK_side_wires dosahuje podobné přesnosti, která ale stále není dostatečná pro nasazení do výroby. Nejvyšší přesnosti dosahuje model při klasifikaci třídy OK_port_wires. Tato přesnost už by mohla být považována za dostatečnou, vzhledem k tomu, že označení drátků, které jsou v pořádku za špatné, nepředstavuje riziko selhání kontrolovaného zařízení.

Obecně je model velmi přesný v nalezení objektů, ale zaostává v klasifikaci. Dosažené přesnosti klasifikace jsou ale dostatečně vysoké na to, aby se dalo předpokládat, že metoda detekce vad bondovaných drátků byla zvolena správně. Dalšího zlepšení přesnosti klasifikace je možné dosáhnout rozšířením trénovacího datasetu. Každopádně se předpokládá další optimalizace modelu po nasazení do výroby, kde bude vystaven mnohonásobně vyššímu objemu obrazových dat, což povede k odhalení dalších potenciálních nedostatků.



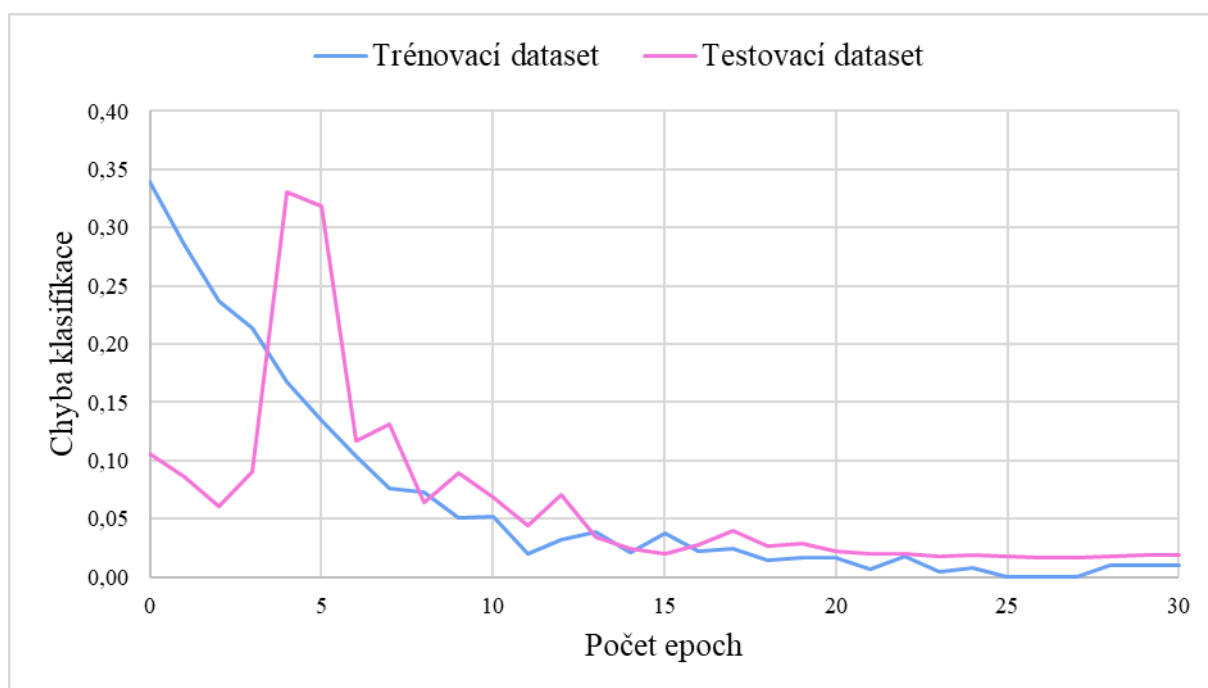
Obr. 6.6 – Vizualizace výstupu modelu Wires na vstupní fotografii

6.3 PORT

V tab. 6.8 jsou uvedeny základní konfigurované parametry modelu Port. Všechny parametry jsou dostupné v příloze A této práce. Trénovací dataset modelu je tvořen 617 fotografiemi a testovací dataset 3438 fotografiemi. Model byl trénován s optimalizační metodou SGD, se kterou byly trénovány předtrénované váhy. Chyba klasifikace je počítána křížovou entropií. Tento model klasifikuje na základě změny světlosti portu a přítomnosti kontaminací, jejichž vzhled je velmi variabilní. Samotné nasvícení scény dostatečně zvýrazňuje zmíněné klasifikační aspekty a nebyla nalezena vhodná forma augmentace, která by nemožifikovala obraz nereálným způsobem. Z těchto důvodů byla jedinou aplikovanou augmentací změna velikosti na vstupní velikost modelu. Na obr. 6.7 je průběh přesnosti klasifikace na trénovacím a testovacím datasetu během tréninku.

Tab. 6.8 – Základní konfigurované parametry modelu Port

Parametr	Hodnota
TRAIN.BATCH_SIZE	8
OPTIMIZER	sgd
OPTIM.BASE_LR	0,025
OPTIM.MOMENTUM	0,15
OPTIM.MAX_EPOCH	27
OPTIM.LR_POLICY	cos
TEST.IM_SIZE	434
TRAIN.IM_SIZE	380



Obr. 6.7 – Průběh přesnosti klasifikace modelu Port během tréninku

6.3.1 Test přesnosti klasifikace

Model Port je čistě klasifikační a rozhoduje pouze mezi dvěma třídami. Výsledky testu přesnosti klasifikace jsou zpracovány do matice záměn v tab. 6.9. V tab. 6.10 jsou uvedeny procentuální zastoupení správně predikovaných tříd.

Tab. 6.9 – Matice záměn z výsledků testu přesnosti klasifikace modelu Port

	OK	NOK
OK	2985	24
NOK	35	394

Tab. 6.10 – Procentuální přesnosti klasifikace jednotlivých tříd modelu Port

Třída	True Positive, %
OK	99,20
NOK	91,84

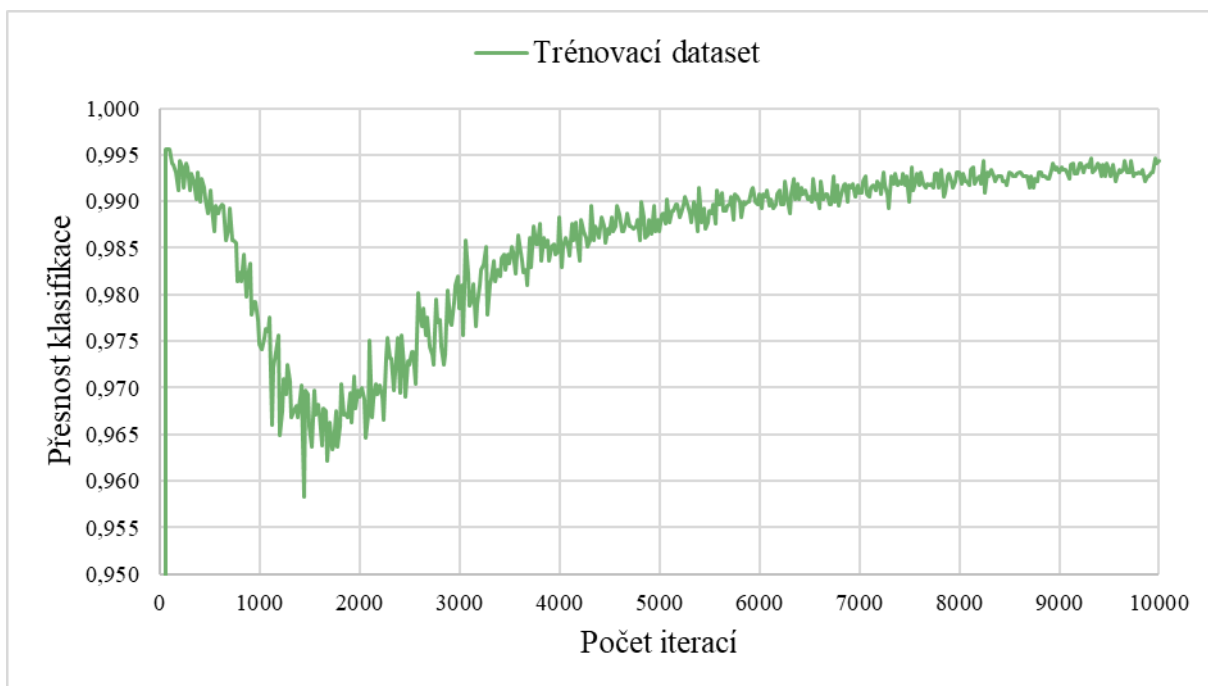
Model dosahuje dostatečné přesnosti klasifikace třídy OK. Přesnost klasifikace třídy NOK je nedostatečná. Podstatné ovšem je, jaké konkrétní vady obsahovaly fotografie, které byly špatně klasifikovány. V tomto případě se totiž převážně jednalo o vady, které by neměly kritický dopad na fungování kontrolovaného zařízení. Dosažená přesnost klasifikace třídy NOK tedy není tak špatná, jak by se mohlo na první pohled zdát. Model ale určitě bude vyžadovat další optimalizaci. Spolehlivost modelu bude možné jednoznačně určit až po zkušebním nasazení do výroby, kde jím projde mnohem vyšší objem dat.

6.4 ANOMALIES

Základní konfigurované parametry modelu Anomalies jsou uvedeny v tab. 6.11. Všechny parametry jsou dostupné v příloze A této práce. Trénovací dataset modelu je tvořen 178 fotografiemi a testovací dataset pouze 163 fotografiemi, protože silnější kontaminace se nevyskytují příliš často. Model byl trénován s optimalizační metodou SGD, která je výchozí metodou pro všechny architektury v Detectron2 a se kterou byly trénovány i předtrénované váhy. Chyba klasifikace je vypočtena křížovou entropií a rozměrově-poziční chyba boxů je vypočtena metodou Smooth L1. Trénovací data byla augmentována horizontálním překlopením, za účelem rozšíření počtu instancí kategorií detekovaných objektů. Na obr. 6.8 je průběh přesnosti klasifikace objektů na trénovacím datasetu během tréninku.

Tab. 6.11 – Základní konfigurované parametry modelu Anomalies

Parametr	Hodnota
SOLVER.IMS_PER_BATCH	4
SOLVER.BASE_LR	0,00025
SOLVER.MOMENTUM	0,9
SOLVER.CLIP_GRADIENTS.ENABLED	False
SOLVER.MAX_ITER	10000
INPUT.MAX_SIZE_TEST	1280
INPUT.MAX_SIZE_TRAIN	1280
INPUT.MIN_SIZE_TEST	1000
INPUT.MIN_SIZE_TRAIN	1000
INPUT.RANDOM_FLIP	horizontal
MODEL.BACKBONE.FREEZE_AT	2
MODEL.ANCHOR_GENERATOR.ASPECT_RATIOS	0,25; 0,5; 1,0; 2,0; 4,0
MODEL.ANCHOR_GENERATOR.SIZES	32; 64; 128; 256; 512



Obr. 6.8 – Průběh přesnosti klasifikace objektů modelu Anomalies během tréninku

6.4.1 Test nalezení všech objektů

Podobně jako u modelu Wires, není u modelu Anomalies příliš důležitá přesnost lokalizace. Protože se pozice lokalizovaných objektů nevyužívá k výpočtům, které by vyžadovaly určitou míru přesnosti. Detekované objekty jsou navíc silně vizuálně variabilní a ani člověku nemusí být vždy úplně jasné, kde přesně objekt začíná a kde končí. Z toho vyplývá, že podstatná je hlavně schopnost tyto objekty nalézt, na což je zaměřen test nalezení všech objektů. Výsledky tohoto testu jsou uvedeny v tab. 6.12.

Tab. 6.12 – Výsledek testu nalezení všech objektů pro model Anomalies

Třída	Detekováno, %
fiber	95,08
silicon	93,34
eca	73,53
particles	71,70

Dosažené přesnosti detekce jsou od pohledu nedostatečné, to se týká hlavně tříd eca a particles. To je způsobeno nedostatečným množstvím dat pro sestavení trénovacího datasetu dostatečně pokrývajícího rozmanitost detekovaných vad. Dalším faktorem, který nemůže být opomenut, je vizuální splývání detekovaných tříd. Třídy silicon, eca a particles jsou někdy vizuálně velmi podobné a v některých případech není možné z fotografie jednoznačně určit o jakou z těchto tříd se jedná. Pro detekci těchto tříd postačí přítomnost alespoň jedné z nich, proto se dá předpokládat lehce vyšší přesnost modelu, než udávají výsledky. Přesto je přesnost tohoto modelu aktuálně nedostatečná a bude potřeba výrazné rozšíření trénovacího datasetu, jakmile bude více dat k dispozici.

7 ZÁVĚR

V rámci teoretické části práce jsou obecně popsány vady osazených PCB, které mají vizuální charakter. Zahrnuty jsou i vady bondování, protože jsou součástí detekční úlohy. Dále jsou rozebrány klíčové vlastnosti obrazu pořizovaného pro vizuální inspekci osazených PCB. Následuje rozbor principu lidského zraku a jeho nedokonalostí v kontextu vizuální inspekce, ze kterého vyplývá význam automatizace vizuální inspekce. Zohledněn je také dopad míry pozornosti na úspěšnost operátorů a situace, ve kterých jsou lidští operátoři zatím nenahraditelní. Závěrem teoretické části je podrobný rozbor architektur konvolučních neuronových sítí vhodných pro detekci vad na osazených PCB. Rozbor je zaměřen na menší množství již vybraných architektur, což umožňuje jejich detailnější popis v rámci práce. Cílem tohoto rozboru je poskytnout ucelený materiál pro nastudování jejich funkce pro studenty, kteří jsou do budoucna nejpravděpodobnější skupinou čtenářů této práce.

Z obrazových dat poskytnutých firmou Vitesco Technologies byly identifikovány a kategorizovány různé typy vad osazených PCB. Na základě těchto kategorií byl navržen komplexní detekční systém kombinující vhodné CNN architektury a metody pro dodatečné výpočty z jejich výstupů. Cílem systému je klasifikace celé inspekční fotografie na základě výsledků jednotlivých modelů a metod.

Byl vytvořen software pro trénink jednotlivých modelů navrženého systému a software pro testování těchto modelů i celého navrženého systému. K tomu bylo využito existujících platforem implementujících vybrané architektury.

Pro navržený systém byly natrénovány všechny modely dle možností daných poskytnutými obrazovými daty. Modely Locator, Wires a Port dosahují poměrně vysokých přesností a jejich nedostatky pravděpodobně půjdou vyřešit mírným rozšířením jejich trénovacích datasetů. Přesnost modelu Anomalies je nízká. Důvodem je nedostatečné množství dat pro sestavení trénovacího datasetu dostatečně pokrývajícího rozmanitost detekovaných vad.

U všech modelů se předpokládá další optimalizace na mnohem rozsáhlejší množství dat, která odhalí skryté nedostatky. V případě závažnějších nedostatků bude na místě změna architektury modelu nebo dokonce úplná změna v přístupu k detekci konkrétní vady. Například klasifikace bondovaných drátků modelem Wires by mohla být nahrazena segmentací doplněnou o měření přímosti drátků. Což je přístup, který nebyl preferován z důvodu časově mnohem vyšší náročnosti na implementaci.

Navržené metody pro měření parametrů z výstupů modelů vyžadují korektně stanovené limity, jejichž nalezení není součástí této práce, protože vyžaduje statistickou analýzu většího

objemu dat s již důkladně optimalizovanými modely systému. Pro demonstrační účely byly zvoleny orientační hodnoty těchto limitů.

POUŽITÁ LITERATURA

- 13 Types of PCB Soldering Defect and Tips*. 2021. Online. PCBONLINE. Dostupné z: <https://www.pcbonline.com/blog/types-of-pcb-soldering-defects.html>. [cit. 2024-04-22].
- 3 Most Common PCB Assembly Defects*. 2020. Online. DigiSource. Dostupné z: <https://blog.thedigisource.com/3-common-pcba-defects>. [cit. 2024-04-22].
- Automated Optical Inspection (AOI) – An Overview for Beginners*. 2022. Online. IBE Electronics. Dostupné z: <https://www.pcbaaa.com/automated-optical-inspection/>. [cit. 2024-05-03].
- Common PCB Manufacturing Defects: All you need to know*. 2023. Online. MorePCB. Dostupné z: <https://morepcb.com/common-pcb-manufacturing-defects/>. [cit. 2024-04-22].
- Computer Vision*. 2024. Online. Papers with Code. Dostupné z: <https://paperswithcode.com/area/computer-vision>. [cit. 2024-05-06].
- ELFWING, S.; UCHIBE, E.; DOYA, K. *Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning*. v3. 2017. Online. arXiv. Dostupné z: <https://arxiv.org/abs/1702.03118v3>. [cit. 2024-05-03].
- GONG, Y.; LIU, G.; XUE, Y.; LI, R.; MENG, L. 2023. A survey on dataset quality in machine learning. *Information and Software Technology*. Volume 162. Online. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0950584923001222>. [cit. 2024-05-08].
- GREEN, T. A Critical Review of Wirebond Visual Inspection Criteria. International Symposium on Microelectronics. 2017. Online. Bethlehem, Pennsylvania: TJ Green Associates LLC. Dostupné z: <https://meridian.allenpress.com/ism/article/2017/1/000451/35917/A-Critical-Review-of-Wirebond-Visual-Inspection>. [cit. 2024-04-24].
- HE, K.; GKIOXARI, G.; DOLLÁR, P.; GIRSHICK, R. *Mask R-CNN*. v3. 2018. Online. arXiv. Dostupné z: <https://arxiv.org/abs/1703.06870v3>. [cit. 2024-05-03].
- HE, K.; ZHANG, X.; REN, S.; SUN, J. *Deep Residual Learning for Image Recognition*. 2015. Online. arXiv. Dostupné z: <https://arxiv.org/abs/1512.03385>. [cit. 2024-05-03].
- KIRILLOV, A.; WU, Y.; HE, K.; GIRSHICK, R. *PointRend: Image Segmentation as Rendering*. v2. 2020. Online. arXiv. Dostupné z: <https://arxiv.org/abs/1912.08193v2>. [cit. 2024-05-03].

- KUJAWIŃSKA, A.; VOGT, K. 2015. Human Factors in Visual Quality Control. *Management and Production Engineering Review*. no. 2. Online. Dostupné z: https://www.researchgate.net/publication/279248156_Human_Factors_in_Visual_Quality_Control. [cit. 2024-05-03].
- Label Studio Docs*. 2024. Online. HumanSignal. Dostupné z: <https://labelstud.io/guide/>. [cit. 2024-05-08].
- LIN, T.; DOLLÁR, P.; GIRSHICK, R.; HE, K.; HARIHARAN, B.; BELONGIE, S. *Feature Pyramid Networks for Object Detection*. v2. 2017. Online. arXiv. Dostupné z: <https://arxiv.org/abs/1612.03144v2>. [cit. 2024-05-03].
- OpenCV: Documentation*. Version 4.x. 2024. Online. OpenCV. Dostupné z: <https://docs.opencv.org/4.x/index.html>. [cit. 2024-05-08].
- PCB Inspection: Diffuse, Coaxial, and Low-Angle Lighting Techniques*. 2021. Online. Advanced illumination. Dostupné z: <https://www.advancedillumination.com/application-notes/pcb-inspection/>. [cit. 2024-04-26].
- RADOSAVOVIC, I.; JOHNSON, J.; LO, W.; DOLLÁR, P.; GIRSHICK, R. et al. *pycls*. 2021. Online, GitHub repozitář. Meta Research. Dostupné z: <https://github.com/facebookresearch/pycls>. [cit. 2024-05-03].
- REN, S.; HE, K.; GIRSHICK, R.; SUN, J. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. v3. 2016. Online. arXiv. Dostupné z: <https://arxiv.org/abs/1506.01497v3>. [cit. 2024-05-03].
- RUSS, J. C.; NEAL, F. B. 2016. *The Image Processing Handbook*. 7th ed. Boca Raton, Florida: CRC Press. ISBN 978-1-4987-4026-5
- SAINT-EXUPÉRY, Antoine de. 2014. *Malý princ*. 14. vyd. Richard PODANÝ (překladař) Praha: Albatros. ISBN 978-80-00-03424-9
- TAN, M.; LE, Q. V. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. v5. 2020. Online. arXiv. Dostupné z: <https://arxiv.org/abs/1905.11946v5>. [cit. 2024-05-03].
- Top 4 most important factors to consider when selecting a camera for an AOI system*. 2021. Online. JAI. Dostupné z: <https://news.jai.com/blog/factors-to-consider-for-aoi-system>. [cit. 2024-04-26].

- VALENTA, V.; SPRENG, T.; YUAN, S.; WINKLER, W.; ZIEGLER, V. et al. 2015. Design and experimental evaluation of compensated bondwire interconnects above 100 GHz. *International Journal of Microwave and Wireless Technologies*, 7 (3 – 4), pp. 261 – 270. Online. Cambridge, England: Cambridge University Press. doi:10.1017/S1759078715000070. Dostupné z: <https://www.cambridge.org/core/journals/international-journal-of-microwave-and-wireless-technologies/article/design-and-experimental-evaluation-of-compensated-bondwire-interconnects-above-100-ghz/CFA7E0494D1F9740B641B7BBB6F54A2A>. [cit. 2024-04-24].
- WEISBERG, S. 2013. *Applied Linear Regression*. 4th ed. Hoboken, New Jersey: John Wilkey & Sons, Inc. ISBN 978-1-118-38608-8
- WU, Y.; KIRILLOV, A.; MASSA, F.; LO, W.; GIRSHICK, R. *Detectron2*. 2019. Online, GitHub repozitář. Meta Research. Dostupné z: <https://github.com/facebookresearch/detectron2>. [cit. 2024-05-03].

PŘÍLOHY

A – Komprimovaná složka

Příloha k diplomové práci

Vývoj systému detekce vad na osazených PCB

s využitím konvolučních neuronových sítí

Kamil Špicar

Komprimovaná složka

OBSAH

- 1 Text diplomové práce ve formátu PDF.
- 2 Zdrojové kódy softwaru implementujícího navržený systém.
- 3 Natrénované modely, jejich konfigurační soubory, protokoly tréninku a startovní váhy.
- 4 Trénovací a testovací datasety všech modelů.