



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

VIZUÁLNÍ DETEKCE VAD V SÉRIOVÉ VÝROBĚ KONEKTORŮ PRO AUTOMOBILOVÝ PRŮMYSL

VISUAL FAULT DETECTION IN SERIAL PRODUCTION OF CONNECTORS FOR AUTOMOTIVE INDUSTRY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jaroslav Kilian

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Brablc

BRNO 2022

Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Jaroslav Kilian**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Mechatronika
Vedoucí práce: **Ing. Martin Brabc**
Akademický rok: 2021/22

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

Vizuální detekce vad v sériové výrobě konektorů pro automobilový průmysl

Stručná charakteristika problematiky úkolu:

Kontrola kvality je nedílnou součástí sériové výroby a v automobilovém průmyslu je na ni kladen velký důraz napříč dlouhými dodavatelskými řetězci. Smyslem tohoto zadání je prozkoumat možnosti detekce různých druhů vad vyskytujících se na několika typech výrobků, které jsou vyráběny na strojích firmy MD Solutions, která poskytla jejich vzorky.

Cíle bakalářské práce:

- 1) Proveďte rešerši metod a technologií, které se pro tuto aplikaci běžně používají. Zaměřte se na tvarové (chybějící nebo přebývající materiál, špatně umístěný díl) a povrchové (oděrky). Popište možnosti Deep learning toolboxu pro Matlab.
- 2) Zopakujte vzorové experimenty se zadaným software.
- 3) Navrhněte přístup pro detekci vad s využitím strojového učení.
- 4) Navrhněte přístup pro detekci vad na libovolném jiném principu.
- 5) Experimentálně porovnejte navržené přístupy.

Seznam doporučené literatury:

NELLES, Oliver. Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Berlin: Springer, 2011. ISBN 978-364-2086-748.

LJUNG, Lennart. System identification: theory for the user. 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 1999. ISBN 978-0136566953.

NOSKIEVIČ, Petr. Modelování a identifikace systémů. Ostrava: Montanex, 1999. ISBN 80-722-50-0-2.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

V této práci jsou popsány způsoby detekce vad, se zaměřením na vizuální detekci, tedy odhalování vad z fotek. Jsou popsány její základní komponenty a metody, které se pro detekci vad z fotek využívají. Dva přístupy jsou navrženy na výrobcích z firmy Mechatronic Design & Solutions, jeden s využitím hlubokého učení a druhý na základě exaktních metod. Tyto přístupy jsou potom experimentálně porovnány.

Summary

In this thesis, the methods of defect detection are described, focusing on visual detection, i.e. detection from photos. Its basic components and methods used for defect detection from photos are described. Two approaches are proposed on products from Mechatronic Design & Solutions, one using deep learning and the other based on exact methods. These approaches are then experimentally compared.

Klíčová slova

Vizuální detekce vad, strojové vidění, hluboké učení, konvoluční neuronová síť, předzpracování, segmentace, prahování, MATLAB

Keywords

Visual defect detection, machine vision, deep learning, convolutional neural network, pre-processing, segmentation, thresholding, MATLAB

Bibliografická Citace

KILIAN, Jaroslav. *Vizuální detekce vad v sériové výrobě konektorů pro automobilový průmysl*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/140092>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Martin Brablec.

Čestně prohlašuji, že bakalářskou práci na téma Vizuální detekce vad v sériové výrobě konektorů pro automobilový průmysl jsem vypracoval samostatně a s využitím literatury uvedené v příloze této práce.

Jaroslav Kilian

Brno

.

Nejprve bych chtěl poděkovat vedoucímu práce Ing. Martinu Brablcovi za jeho rady a připomínky. Dále bych chtěl poděkovat firmě Mechatronic Design & Solutions za poskytnutí výrobků a pomoc s experimenty.

Jaroslav Kilian

Obsah

1	Úvod	9
2	Rešerše	10
2.1	Druhy detekce vad	10
2.1.1	Ultrazvuk	10
2.1.2	Rentgen	10
2.1.3	Vířivé proudy	10
2.1.4	Akustické emise	11
2.1.5	Magnetický prášek	11
2.1.6	Strojové vidění	11
2.2	Pořízení snímku	11
2.2.1	Osvětlení	12
2.3	Předzpracování	12
2.3.1	Transformace jasu	12
2.3.2	Geometrické transformace	14
2.3.3	Filtrace	14
2.3.4	Segmentace	14
2.4	Detekce	15
2.4.1	Tradiční strojové vidění	16
2.4.2	Strojové učení	16
2.5	Hluboké učení	17
2.5.1	Neuron	17
2.5.2	Umělé neuronové sítě	18
2.5.3	Ztrátová funkce	18
2.5.4	Gradientní sestup	19
2.5.5	Konvoluční neuronová síť	19
2.6	Deep learning toolbox	20
3	Postup a výsledky řešení	23
3.1	Použitá data	23
3.1.1	Fotografování	24
3.1.2	Předzpracování	27
3.2	Vzorové experimenty	28
3.3	Detekce exaktními metodami	28
3.4	Detekce neuronovými sítěmi	36
4	Závěr	41

Literatura	43
Seznam obrázků	46
Seznam tabulek	47
Seznam zkratek	48

1 Úvod

Ačkoliv technologie výroby pokročila neskutečně dopředu a spolehlivost strojů ve všech odvětvích se neustále zvyšuje, stále není možné dosáhnout stavu, kdy by 100 % produktů odpovídalo požadavkům výroby a nevyskytl se žádný zmetek neboli NOK (not okay) kus. Velice důležitá je tedy schopnost zmetky závčas odhalit. Konkrétně v automobilovém průmyslu se očekává, že budou odhaleny všechny zmetky. Každý, který by prošel, by mohl způsobit větší a větší problémy dále ve výrobě a ohrozit reputaci firmy, kdyby se dostal ven.

Odhalit všechny zmetky není jednoduché, ale samozřejmě i technologie kontroly kvality šly velice rychle dopředu a existuje velké množství způsobů, jak kontrolu v sériové výrobě provádět. Jejich volba vždy závisí na druhu výrobků a očekávaném typu vad. Samozřejmě je nutné používat nedestruktivní metody. Je to například detekce pomocí ultrazvuku, rentgenu či vířivých proudů. Tyto a další metody jsou popsány v kapitole 2. Tato práce se zabývá primárně metodami spadajícími pod strojové vidění, tedy metodami hledajícími vady na výrobku z fotek. V této práci jsou popsány důležité součásti a postupy při detekci vad z fotek. Nejprve je nutné zvolit vhodný způsob snímání fotky a jemu odpovídající způsob osvětlení. Následuje předzpracování, kdy je snímek zpracován dle potřeby pro další použití.

Poté již následuje samotná detekce. Obecně existují dva hlavní přístupy k této problematice. První je za pomoci exaktních metod, kdy pomocí konkrétních kritérií nastavíme požadované hodnoty, které by výrobek měl splňovat a potom u každého výrobku stanovujeme, zda dané limity splňuje. Do této metody by se již daly zahrnout některé techniky předzpracování, jako třeba segmentace či hledání hran. Druhý je aplikace strojového učení. Existuje více způsobů jak jej využít, ale v této práci byla využito hluboké učení, což je podoblast strojového učení, který využívá neuronové sítě, struktury modelované podle neuronů v lidském mozku. Pro detekci vad z fotek se využívají zejména konvoluční neuronové sítě.

Oba přístupy mají své výhody a nevýhody. Klasický přístup se nejlépe hodí, když je možné zajistit konzistentní podmínky pro snímání a očekávané vady jsou jednoduše měřitelné. Neuronové sítě díky schopnosti se učit zvládnou detekovat i hůře kvantifikovatelné vady, které by klasickým přístupem nebylo možné odhalit. Potřebují ovšem velké množství dat na učení, které občas není jednoduché získat, zejména pak NOK kusy. Problém získávání dostatečného množství dat pro neuronové sítě je v práci také adresován.

V této práci jsou navrženy a porovnány 2 metody, jedna exaktní a druhá za pomoci hlubokého učení. Jako vzorky pro detekci vad byly použity výrobky ze strojů firmy Mechatronic Design & Solutions, konkrétně konektory pro nabíjení elektroaut. Použity byly fotky hotových kusů již roztříděných na OK a NOK, neroztříděné fotky přímo z výroby a fotky kusů foceny autorem práce ve světelném boxu. Obě metody byly navrhovány a implementovány v MATLABU, konkrétně v Image processing toolboxu a Deep learning toolboxu.

2 Rešerše

2.1 Druhy detekce vad

Existuje mnoho různých metod, které se používají pro detekci vad. Při sériové výrobě je samozřejmě nutné používat metody nedestruktivní (Non-destructive testing - NDT). Je to například testování pomocí ultrazvuku, rentgenu, vířivých proudů, akustických emisí, magnetického prášku a vizuální detekce. Každá metoda má své výhody a nevýhody a používá se na jiné druhy výrobků a vad. Tyto metody jsou popsány níže, se zaměřením na vizuální detekci.

2.1.1 Ultrazvuk

Tato metoda využívá k detekci vad zvukové vlny o vysokých frekvencích. Pulzátor generuje vysokonapěťový elektrický impuls, který je měničem přeměněn na vysokofrekvenční ultrazvukovou (zvukovou) energii.[8] Pro detekci se používají dva způsoby uspořádání. Nejčastěji se měří ozvěna pulzu, kdy vysílač i přijímač jsou na stejné straně testovaného výrobku a jsou přiloženy přímo na povrchu testovaného vzorku. Ta se nejvíce hodí pro testování homogenních materiálů. Je možné měřit i průchod pulzu skrz výrobek, kdy vysílač je na jedné straně a přijímač na druhé. Ta se nejvíce hodí pro komplexnější vzorky s komplexnější geometrií, u kterých nemůže být vysílač ani přijímač přímo na povrchu.

Výhody používání ultrazvuku jsou vysoká schopnost penetrace umožňující hledání vad hluboko ve vzorku, vysoká citlivost umožňující detekci velmi malých a vad vysoká rychlost skenu.

Nevýhody jsou náročnost na správné nastavení a náročnost na schopnosti a znalosti pracovníka pro vyhodnocení signálu. Také není možné testovat touto metodou příliš malé či příliš složité části.[16]

2.1.2 Rentgen

Tato metoda využívá rentgenové záření. Na rotující vzorek jsou vyzářeny rentgenové paprsky. Poté je měřena míra absorpce rentgenového záření a z toho je poskládán výsledný obraz vzorku s jasně viditelnými vnitřními detaily díky vysoké penetrační schopnosti rentgenových paprsků.[9]

Výhodami jsou právě vysoká schopnost penetrace paprsků a to, že na funkčnost této metody nemá vliv struktura a vzhled materiálu. Je také jednoduchá na obsluhu.

Největší nevýhodou je riziko ozáření přítomných pracovníků. [16]

2.1.3 Vířivé proudy

Tato metoda se často využívá na detekci vad ve vodivých materiálech, zejména na vady způsobené korozi a napětím. Pomocí cívky jsou v materiálu vyvolány vířivé proudy. Následně sledujeme poruchy v dráhách například pomocí Hallových senzorů, magnetorezistorů a nebo snímacích cívek. Pomocí těchto poruch jsme schopni lokalizovat a charak-

terizovat vady na vzorku.[10]

Výhodou této metody je rychlost a přesnost, možnost detekce bez dotyku vzorku a možnost použití v prostředí s vysokými teplotami.

Velkou nevýhodou této metody je to, že velikost a tvar vady není možno vizualizovat. Také je složité touto metodou detekovat defekty hlouběji ve vzorku a množství materiálů, na které jde metoda aplikovat, je omezené. [16]

2.1.4 Akustické emise

Tato metoda funguje na principu měření zvuku, které vydávají defekty jako například mikrotrhliny. Zvukové vlny způsobené těmito vadami se šíří soustředně z místa vzniku a jsou detekovány polem vysoce citlivých piezoelektrických snímačů.[13]

Velkou výhodou této metody je schopnost rozlišovat mezi rozvíjejícími se defekty a stagnujícími defekty. Tato metoda je také rychlá a přesná.

Nevýhodou jsou vysoké nároky na schopnosti a znalosti potřebné k určení konkrétní vady dle naměřených signálů. [16]

2.1.5 Magnetický prášek

Tato metoda je vhodná na detekci vad na povrchu či v jeho blízkosti. Kovový vzorek je nejprve magnetizován, čímž se v něm vytvoří magnetický tok. Pokud magnetický tok při průchodu vzorkem narazí na defekt, začne "unikat ven", v důsledku čehož vytváří sekundární magnetické póly. Na vzorek se poté umístí magnetické částičky (magnetický prášek), které jsou přitahovány k místu úniku magnetického toku. Ty jasně ukážou místo a velikost vady.[11]

Výhodou této metody je vysoká přesnost a nízká cena a je použitelná na vzorky téměř jakékoliv velikosti.

Nevýhodou této metody je omezení pouze na feromagnetické materiály. Výsledky detekce hodně závisí na geometrii vzorku a je velice náročné tento proces zautomatizovat. [16]

2.1.6 Strojové vidění

Strojové vidění v sobě zahrnuje všechny aplikace, ve kterých je kombinován software a hardware za účelem provádění úkonů na základě pořizování a zpracování fotek, kam spadá i vizuální detekce vad.[17]

Pro vizuální detekci vad je důležité jak zvolit osvětlení a způsob pořízení fotky, jak fotky zpracovat a jakým způsobem vyhodnotit vady na vzorku. Tyto aspekty se volí v závislosti na typu vyráběných dílů a na typu očekávaných vad a jsou postupně popsány v dalších kapitolách práce.

Výhodou vizuální detekce je široké spektrum využití a vysoká přesnost. Výhodou je také, že u vizuální detekce nezáleží na struktuře povrchu testovaného vzorku a je jednoduché provádět detekci automaticky.

Hlavní nevýhodou je možnost detekovat pouze vady na povrchu vzorku. [16]

2.2 Pořízení snímku

V následujících kapitolách budou podrobněji popsány jednotlivé aspekty strojového vidění, počínaje pořízením snímku.

Snímání se dá rozdělit na 1D, 2D. U 1D systémů je analyzována vždy pouze jedna

linie. Často se využívá při detekci vad na materiálech vyráběných kontinuálním procesem, jako papír, kovy, plasty a další pláty.

Nejčastěji používané jsou 2D systémy, kdy kamera vytvoří snímek požadované velikosti rozlišení. V některých případech, například když je potřeba ověřit stav válcového předmětu, je však lepší vytvořit 2D snímek rotováním předmětu před 1D scannerem. [17]

2.2.1 Osvětlení

Dobré osvětlení je klíčem k pořízení kvalitního snímku pro detekci vad. Systémy strojového vidění vytváří snímky analýzou světla odraženého od objektu, nikoliv analýzou objektu samotného. Správně zvolené osvětlení může zvýraznit důležité vlastnosti snímku a naopak potlačit ty nedůležité. Často používaným zdrojem světla jsou LED, protože mohou být použita v mnoha uspořádáních pro dosažení kýženého efektu. V aplikacích strojového vidění je nejpoužívanější uspořádání LED do kruhu. Toto uspořádání umožňuje dosažení vysokého jasů a dá se pohodlně nainstalovat. Používá se často na kontrolu čipů, desek plošných spojů nebo pro osvětlení v mikroskopu.

Druhým často používaným uspořádáním LED je uspořádání do řady. To má dobrý odvod tepla a flexibilitu použití. Používá se například pro detekci vad u velkých konstrukčních dílů, jako měděných pásků a ocelových plátů.

Podle pozice vzorku, zdroje světla a kamery rozlišujeme přední a zadní osvětlení. Nejčastěji používané je přední, kdy zdroj světla i kamera je na stejné straně předmětu. Toto osvětlení se hodí pro detekci povrchových defektů, škrábnutí a důležitých detailů předmětu, hlavně vlastností povrchové textury. Osvětlení závisí na úhlu dopadu paprsků na předmět. Podle toho, zda se světlo odráží přímo do kamery, se přední osvětlení dělí na osvětlení ve světlém nebo tmavém poli. Malý úhel osvětlení může zvýraznit hrany a výšku předmětu a topologickou strukturu.

Koaxiální osvětlení je speciální způsob předního osvětlení. Světlo prochází skrz polo-propustné zrcadlo, čímž vzniknou paprsky koaxiální s kamerou. Koaxiální osvětlení poskytuje rovnoměrnější osvětlení a omezuje přílišný odraz na předmětu.

Využívá se také rozptýlení světla pomocí kupolové konstrukce, což se využívá například při kontrole pájených spojů nebo pinů na čipu.

V případě zadního osvětlení je zdroj světla za objektem. Umožňuje zvýraznit stín u průhledného předmětu nebo pozorovat jeho vnitřek. Výhodou je, že jasně ukazuje obrys předmětu, hodí se tedy při detekci tvaru předmětu a měření rozměrů. [12]

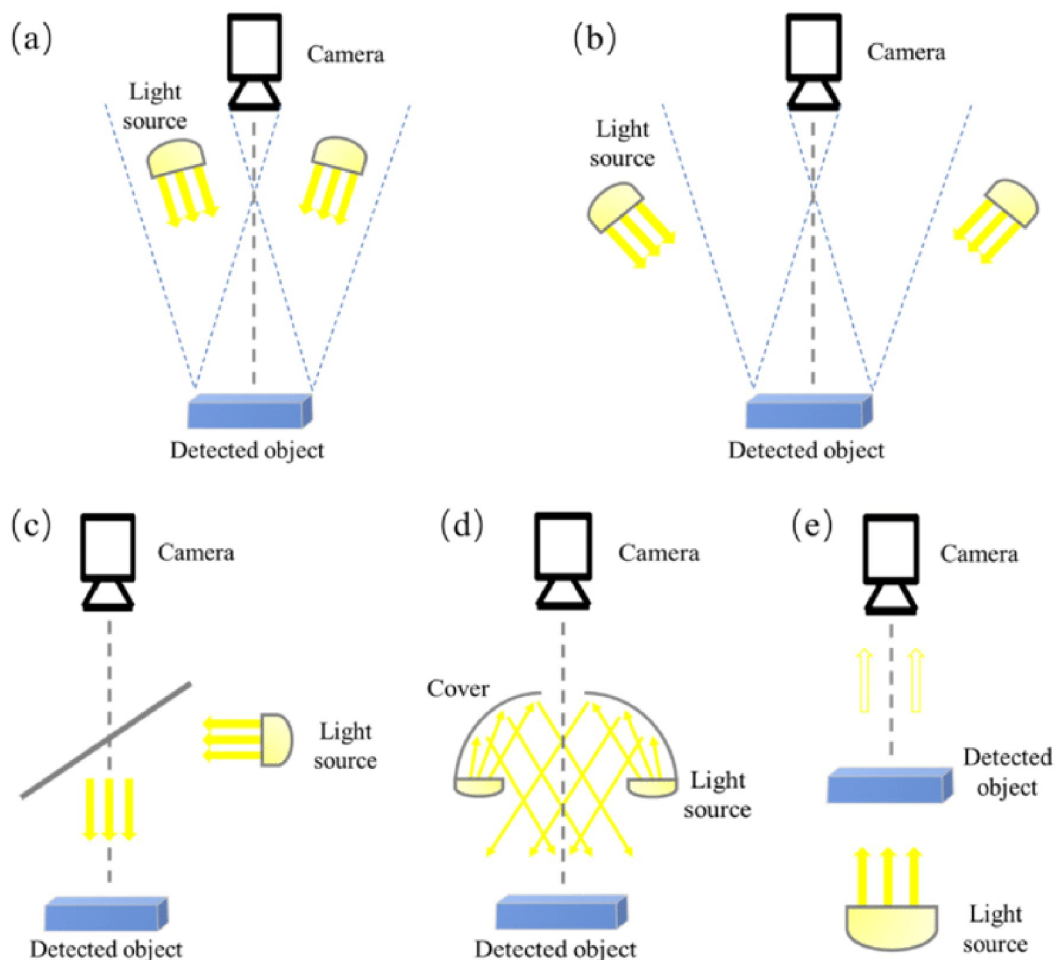
2.3 Předzpracování

Předzpracování je souhrnný název pro techniky, které se aplikují na nezpracovaná data za účelem odstranění šumu a jiných nedostatků a zvýraznění určitých vlastností důležitých pro další zpracování, například právě pro detekci vad.

2.3.1 Transformace jasů

Prvním druhem předzpracování jsou úpravy spojené z transformací jasů pixelů. Dělí se na dva základní druhy, závislá na pozici pixelů a nezávislá neboli gray-scale. Transformace závislá na pozici upraví hodnotu jasů pixelu podle jeho originálního jasů a umístění na fotce. Gray-scale transformace upraví jas pixelu bez ohledu na jeho pozici. Můžeme je dělit na lineární, logaritmické a mocninné.[14]

Základním druhem lineární transformace je transformace snímku z RGB na grayscale.



Obrázek 2.1: Druhy osvětlení a) Přední ve světlém poli b) Přední ve tmavém poli c) Koaxiální d) Rozptýlení kopulí e) Zadní [12]

Jednou z možností je jednoduše sečíst a zprůměrovat jednotlivé složky RGB, ale vzhledem k tomu, že lidské oko vnímá každou složku jinak, výsledný snímek není příliš kvalitní. používá se proto vážená metoda. Ta bere v potaz rozdílné citlivosti oka na jednotlivé složky RGB. Výsledná rovnice pak vypadá následovně: $Greyscale = 0.299R + 0.587G + 0.114B$ [15]. Dalším druhem je tzv. negativní transformace. Při ní je odečtena hodnota pixelu od maximální hodnoty. Tím vznikne invertovaný snímek, kde jsou zvýrazněny detaily, které byly na originálním snímku příliš tmavé.

Logaritmické transformace se používají pro zvýšení jasu na snímku. Hodnotu každého pixelu nahradí jeho logaritmem. Díky tomu jsou pixely s nízkou hodnotou zobrazeny na rozšířeném intervalu, například pixely s hodnotami 0-50 ve vstupním snímku budou nyní na intervalu 0-100. Rozsah u vyšších hodnot je naopak stlačen. Tím je možno lépe zobrazit snímky, kde je většina informací ve tmavých částech. Inverzní logaritmická transformace dělá opak, tedy stlačí rozsah u nižších hodnot a rozšíří ho u vyšších.

Mocninné transformace jsou určeny ke zvýšení kontrastu na fotce. Nazývají se také gamma korekce. Jeden ze způsobů zvyšování kontrastu se nazývá natahování kontrastu, nebo také normalizace. Tato metoda vezme libovolný interval intenzit na fotce a rozšíří a napasuje ho na libovolný jiný interval.

Druhý způsob zvýšení kontrastu se nazývá ekvalizace histogramu. Při něm jsou rozší-

řeny oblasti histogramu s vysokými intenzitami a ztenčeny oblasti s nízkými intenzitami. [14]

2.3.2 Geometrické transformace

Geometrické transformace jsou techniky sloužící k přeskupení pixelů na fotce. Souřadnice vstupní fotky jsou transformovány na souřadnice výstupní fotky pomocí vybrané transformační funkce. Tedy je to mapování jednoho systému souřadnic na druhý. Mapování může být dopředné (mapování pixelů vstupní fotky na výstupní) a nebo zpětná (mapování pixelů výstupní fotky na vstupní). Tato transformace zahrnuje většinou nějaký druh lineárního mapování jako translace, rotace nebo změny měřítka. Případně je možno použít i nelineární mapování jako zkroucení, zvlnění a sférický efekt. [14]

2.3.3 Filtrace

Filtrace je obecně používána ke zvýšení detailu fotky. Existují dva základní druhy, prostorová a frekvenční. V prostorové filtraci záleží hodnota zpracovaného pixelu nejen na sobě samém, ale i na okolních pixelech. Provádí se tak, že hodnota libovolného pixelu na výstupní fotce je vypočítána aplikováním určitého algoritmu na okolní pixely odpovídajícího pixelu na vstupní fotce. Existují různé druhy prostorových filtrů, jako například vyhlazovací filtr, který slouží k rozmazání fotky nebo zaostřovací filtr, který slouží ke zvýraznění rozmazaných detailů.

Existuje také Band Pass filtr, který nepropustí velmi vysoké a velmi nízké frekvence, ale propustí frekvence v intervalu (pásu) uprostřed. Band Pass filtrace může být použita k zvýraznění hran (zeslabením nízkých frekvencí) a redukci šumu (zeslabením vysokých frekvencí).

Frekvenční filtry slouží ke zpracování fotky ve frekvenční oblasti. Jsou to například low-pass filtry, sloužící k rozmazání fotky nebo high-pass filtry, sloužící ke zvýraznění hran a zostření detailů fotky. [14]

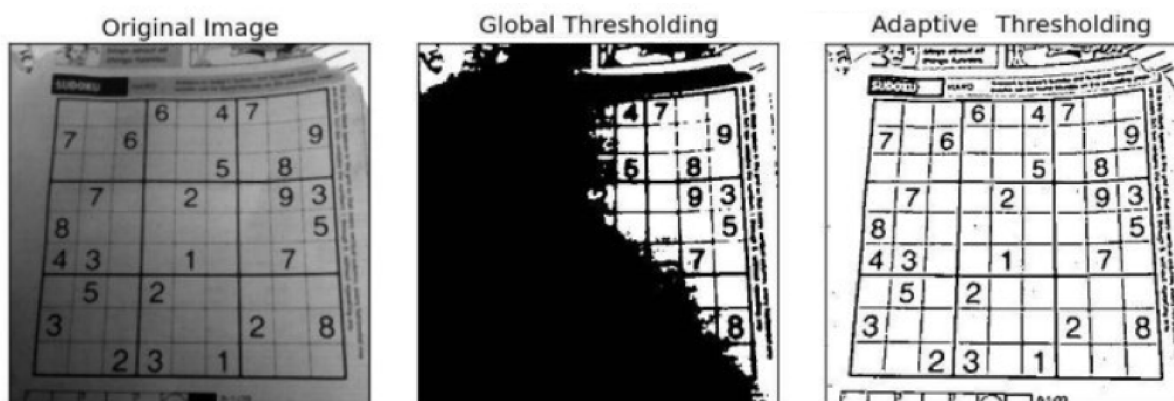
2.3.4 Segmentace

Segmentace slouží k rozdělení fotky na části, což se většinou používá k nalezení důležitých objektů na fotce. Mezi metody segmentace patří například prahování a detekce hran.

Prahování

Jednou ze základních technik pro segmentaci je prahování. Je to proces transformace vstupní černobílé fotky na binární fotku nebo na fotku s různými stupni šedé pomocí určitého prahu. Cílem prahování je vytáhnout z fotky určité pixely a odstranit jiné, neboli označit pixely, které patří do popředí se stejnou intenzitou a které do pozadí s různými intenzitami. Práh může být globální, tedy jedna hodnota pro celou fotku a nebo lokální, tzv. adaptivní prahování. Ten se používá, například když je nerovnoměrné osvětlení nebo jiná komplikace, kvůli které by bylo náročné fotku segmentovat pomocí jednoho prahu. Původní fotka je rozdělena na části a pro každou část je použita jiná hodnota prahu pro segmentaci.

Jedna z metod prahování je založena na tvaru histogramu. Pokud máme například jeden tmavý objekt na světlém pozadí, bude mít histogram fotky pouze dva módy, tzv. bimodální histogram. Poté stačí jen vybrat takový práh, který tyto dva módy rozdělí.



Obrázek 2.2: Efekt globálního a adaptivního prahování[14]

Detekce hran

Detekce hran je velice důležitý nástroj pro segmentaci fotek. Funguje na základě hledání lokálních variací v hodnotách pixelů na fotce. Zobrazení hran snižuje množství dat, které je nutné zpracovat, ale zároveň zachová důležité informace ohledně tvarů objektů na fotce. Existují různé druhy detekce hran, například Sobel, Canny a Prewitt.

Většina metod detekce hran funguje na základě předpokladu, že hrana se objevuje na místě, kde je nespojitost ve funkci intenzity a nebo velmi prudký gradient na fotce. Sobel i Prewitt, využívají dvojici filtrů o rozměrech 3×3 (Kernely). Ty jsou oba stejné, ale jeden z nich je natočený o 90° , tedy jeden pro osu hrany v ose x a druhý pro hrany v ose y. Liší se v konkrétním tvaru Kernelů. Jejich výhodou je v jednoduchých a rychlých výpočtech a detekované hrany jsou i s jejich orientací. Nevýhodou je, že jsou citlivější na šum a nejsou tak přesné.

Cannyho nejprve vyfiltruje fotku, aby se zbavil šumu, který by detekci hran mohl negativně ovlivnit. Výhodou Cannyho filtru je vyšší přesnost a menší ovlivnění šumem, právě díky filtraci. Nevýhodou je pomalá rychlost a vysoká náročnost na výpočty. [14]



Obrázek 2.3: a) Originální snímek b) Sobel c) Prewitt d) Canny[14]

2.4 Detekce

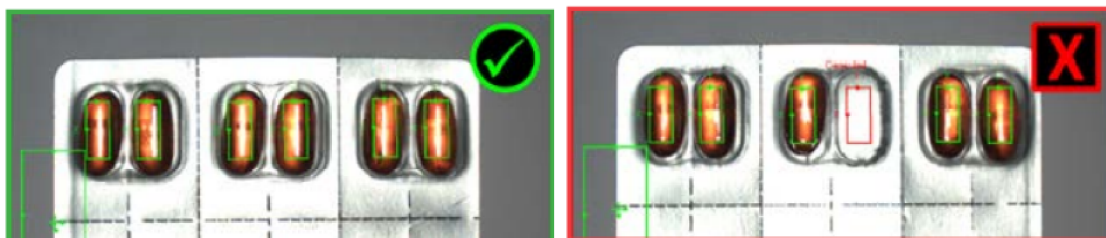
Nejdůležitější součástí soustavy je samotná metoda detekce vady a klasifikace vadných kusů. Obecně existují dvě hlavní metody. Tradiční, založená na exaktních měřeních požadovaných vlastností, a stále více využívané strojové učení, v čele s hlubokým učením.

2.4.1 Tradiční strojové vidění

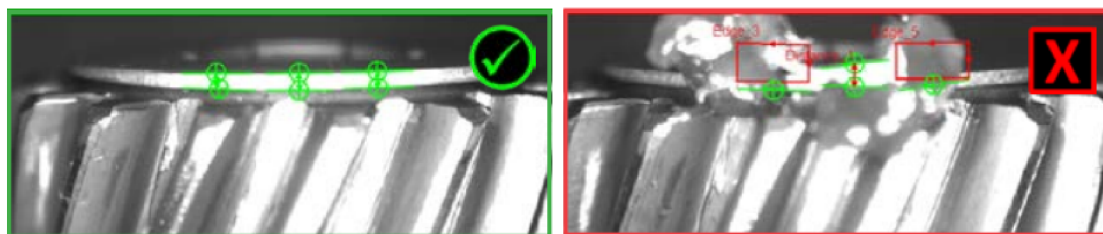
Tradiční metody detekce vad fungují nejlépe na konzistentních výrobcích z jasně danými a měřitelnými vadami. Fungují na principu přesně definovaných parametrů a pravidel. U každého vzorku jsou postupně algoritmicky ověřeny všechny předem naprogramované parametry. [17] Například zda na určitém místě nechybí konkrétní díl nebo zda vzdálenost mezi dvěma díly na vzorku je taková, jaká má být.

Prvním a nejdůležitějším krokem v detekci vady je vždy najít oblasti zájmu, k čemuž slouží metody segmentace. Tím už je možno detekovat některé zmetky, kterým chybí důležité části. Jakmile jsou nalezeny oblasti zájmu, jsou ověřeny exaktní charakteristiky, podle kterých se určí, jestli je vzorek OK anebo NOK.

Vždy záleží na charakteru konkrétního vzorku a konkrétní hledané vady, ale je možno ověřit například jas v dané oblasti, kontrast, barvu pixelů a nebo velikost či vzdálenost daných dílů na výrobku.[22]



Obrázek 2.4: Kontrola jasů v dané oblasti[22]



Obrázek 2.5: Detekce hran[22]

2.4.2 Strojové učení

Strojové učení je odnož umělé inteligence a počítačových věd, která se zabývá použitím algoritmů k imitaci způsobu, kterým se učí lidé. Na rozdíl od exaktních metod nevyužívá přesně definovaných parametrů, ale na příkladech OK a NOK vzorků se naučí, jak mají vzorky vypadat, a podle toho určuje, co je zmetek a co ne. Strojové učení se hodí zejména ve chvílích, kdy je příliš složité naprogramovat algoritmus s exaktními metodami, nebo když vzorek obsahuje příliš mnoho proměnlivých částí či rušivých vlivů na pozadí. Velkou výhodou oproti tradičním metodám je možnost přeučení na nové fotky vzorků a adaptace na nové vzorky, bez nutnosti vše znovu přeprogramovat.[18]

Pro účely detekce vad je většinou používána metoda učení s učitelem (supervised learning). V ní jsou použita označená data pro naučení algoritmů ke správné klasifikaci těchto dat, tedy v případě detekce vad jsou to fotky kusů, které jsou označeny OK a ty

které jsou označeny NOK.

Existuje mnoho různých algoritmů strojového učení s učitelem a jedním z nich je tzv. hluboké učení využívající neuronové sítě.

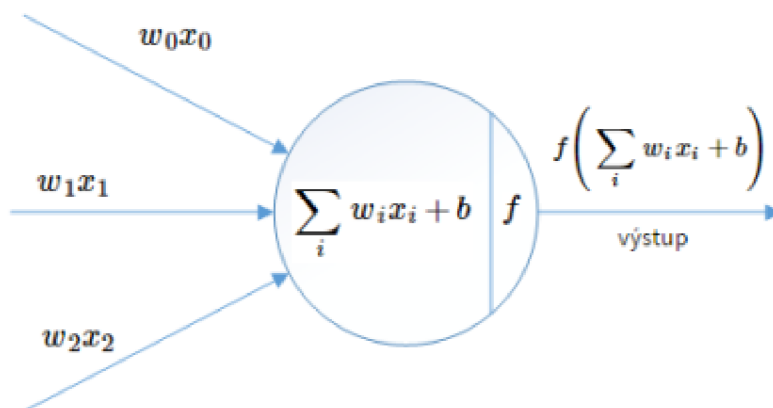
2.5 Hluboké učení

Hluboké učení je oblast spadající pod strojové učení. Zabývá se vytvářením neuronových sítí, které se snaží napodobovat strukturu a funkce lidského mozku.

2.5.1 Neuron

Stejně jako u lidského mozku, základním stavebním kamenem neuronové sítě je neuron. První jednoduchý matematický model neuronu vytvořili již v roce 1943 Warren McCulloch a Walter Pitts. [1]

Každý neuron má n reálných vstupů x_1, \dots, x_n . Tyto vstupy jsou ohodnoceny reálnými váhami w_1, \dots, w_n . Uvnitř neuronu je sumační funkce, která sečte všechny vstupy vynásobené jejich váhami a bias, který umožňuje posunout výsledek funkce o konstantní hodnotu. Na výstupu neuronu je výsledná suma zpracována aktivační funkcí. Výsledná hodnota je poté výstup, který slouží jako vstup pro další neurony.



Obrázek 2.6: Model neuronu[23]

Na obrázku je vidět jednoduchý model neuronu, kde:

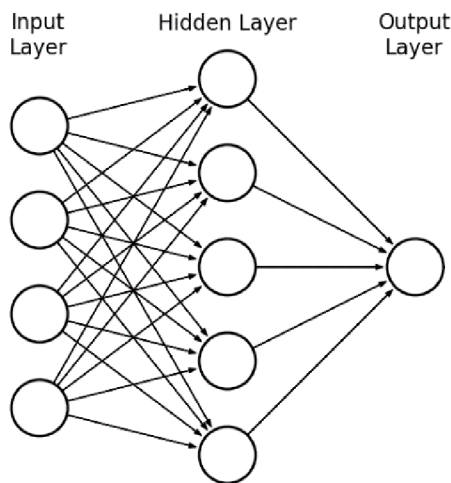
- x jsou vstupy do neuronu
- w jsou váhy
- b je bias
- f je aktivační funkce

Aktivační funkce definuje vlastnosti neuronu a může být jakákoliv matematická funkce. V minulosti se nejvíce využívala funkce sigmoid, ale kvůli jejím limitacím se dnes nejvíce využívá tzv. ReLu (usměrněná lineární jednotka). Pro každý záporný vstup tato funkce vrací nulu a pro každý kladný vstup vrátí hodnotu vstupu. Rovnice ReLu se dá jednoduše matematicky vyjádřit jako $f(x) = \max(0, x)$. [3]

2.5.2 Umělé neuronové sítě

Samotný jeden neuron nemá v podstatě žádné využití, ale pokud spojíme dohromady dva a více, dostaneme umělou neuronovou síť, která je schopna řešit komplexní problémy, jako například detekci vad. Dvě základní architektury pro umělou neuronovou síť jsou acyklická (neboli dopředná), kde informace proudí pouze v jednom směru, anebo cyklická (rekurentní), kde informace mohou proudit i v opačném směru. Pro klasifikaci snímku se používají dopředné sítě.

Jak již bylo zmíněno, dopředné neuronové sítě jsou takové, kdy informace mohou proudit pouze v jednom směru. Neurony jsou organizovány do vrstev s libovolným počtem neuronů. Každý neuron je propojen se všemi neurony v sousedních vrstvách, ale není propojen s žádným ve svojí vrstvě. Jednotlivé vrstvy se často zjednodušují na vstupní vrstvu, skryté vrstvy, jichž může být libovolný počet, a výstupní vrstvu. [2]



Obrázek 2.7: Dopředná neuronová síť[24]

2.5.3 Ztrátová funkce

Proces učení neuronové sítě znamená využití algoritmu, který nalezne vhodné váhy a biasy tak, aby na výstupu sítě bylo správné určení všech vstupů. Problém je, že počet správných výstupů není spojitá funkce a drobné změny vah a biasů se tak většinou nepromítnou na celkovém počtu správných výstupů. Proto se pro kvantifikaci výsledků v průběhu učení používá takzvaná ztrátová funkce (v anglické literatuře Cost function nebo Loss function),

$$C(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2$$

kde w jsou všechny váhy v síti, b jsou všechny biasy, n je celkové množství vstupů a a je vektor výstupů, když x je vstup. Tato funkce $C(w, b)$ se nazývá střední kvadratická chyba. Hodnota ztrátové funkce je nízká, tedy $C(w, b) \approx 0$ právě ve chvíli, kdy $y(x)$ se přibližně rovná správnému výstupu a . To znamená, že učební algoritmus odvedl dobrou práci ve chvíli, kdy je schopen najít takovou kombinaci vah a biasů, že $C(w, b) \approx 0$. Naopak nefungoval dobře, když hodnota $C(w, b)$ je vysoká. Znamená to, že $y(x)$ se neblíží

ke správnému výstupu a pro velké množství vstupů.

Cílem učebního algoritmu je najít takovou kombinaci vah a biasů, aby byla dosažena co nejmenší hodnota ztrátové funkce $C(w, b)$.

2.5.4 Gradientní sestup

Základní algoritmus pro učení neuronových sítí se nazývá gradientní sestup. Ten nejprve spočítá pomocí zpětné propagace gradient ztrátové funkce $C(w, b)$. Zpětná se nazývá proto, že algoritmus počítá gradient od výstupu směrem ke vstupu. Algoritmus poté vezme spočtený gradient a posune se v opačném směru.

Gradientní sestup používá pro výpočet gradientu celou várku vstupních dat, což při velkém množství vstupů může zabrat velké množství času a učení je potom velmi pomalé. V praxi se proto používá tzv. stochastický gradientní sestup. Ten pro každý krok odhadne gradient pouze z malého náhodně vybraného vzorku dat, nazývaného *mini-batch*. Pokud je množství vstupních dat dostatečně velké, odhadnutý průměrný gradient z malé várky se bude téměř rovnat průměrnému gradientu ze všech vstupů. Tím se proces učení značně urychlí.

Stochastický gradientní sestup tedy náhodně vybere malou várku vstupních dat, na kterých se učí. Poté náhodně vybere další várku a učí se na ní. Jakmile algoritmus projde všechna vstupní data, říkáme, že skončila epocha a začíná nová, kdy znovu projde všechna data.

Jak bylo řečeno, princip stochastického gradientního sestupu je posunutí v opačném směru, než je spočítaný gradient. To, o kolik se algoritmus posune, určuje zvolený hyperparametr tzv. learning rate. Je nutné správně zvolit jeho velikost. Pokud bude příliš velký, algoritmus nebude schopný správně najít minimum ztrátové funkce a bude kolem něj neustále oscilovat. Naopak když bude příliš malý, změny vah a biasů budou příliš malé a učení potrvá příliš dlouho.

Existují způsoby jak ještě více zlepšit učení sítí. Jedním z nich je přidat ke stochastickému gradientnímu sestupu hybnost (Stochastic gradient descent with momentum - SGDM). Algoritmus si potom pamatuje, jakým směrem se posunoval v minulém kroku a zahrne to při posunutí v dalším kroku. [4]

2.5.5 Konvoluční neuronová síť

Konvoluční neuronové sítě jsou nejčastěji využívaným typem sítí na klasifikaci z fotek. Jsou velice schopné v zachycení vzorů na fotce. Jejich výhoda spočívá v redukci propojení neuronů a tím pádem ke značné redukci vah, které je nutno počítat. Obsahuje 3 hlavní vrstvy: konvoluční vrstvu, max-pooling a plně propojenou vrstvu.

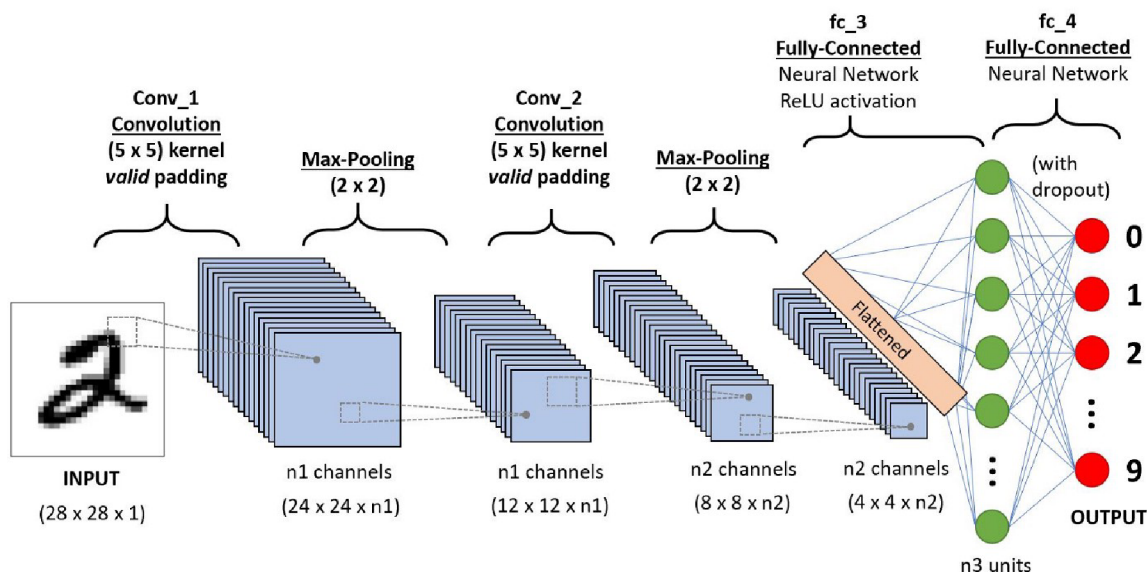
Hlavní vrstvou pro funkčnost sítě je vrstva konvoluční. Obsahuje určité množství filtrů (kernelů), které se postupně přesouvají přes celou vstupní fotku. Každý filtr může hledat na fotce jiný vzor, například hrany, rohy či části kruhů. Pro každý filtr vznikne nová fotka obsahující konkrétní informace o původní fotce podle toho, na co byl který filtr nastaven. Takto je možné postupně v hlubších vrstvách identifikovat složitější a složitější objekty, jako například tváře anebo konkrétní vady na výrobku. Díky tomu, že každý filtr projede celou fotku, nezáleží na poloze hledaných objektů. Stejně jako u klasické umělé neuronové sítě, jakmile fotka projde konvoluční vrstvou, je na ni použita aktivační funkce, jako například ReLu.

Pooling vrstva slouží primárně ke zjednodušení dat pro další vrstvy srovnatelné se snížením rozlišení fotky. Nejčastější metodou je tzv. Max-pooling. Ten rozdělí fotku na

čtverce či obdélníky a v každém zapíše jen nejvyšší hodnotu.

Plně propojená vrstva je podobná tomu, jak jsou aranžovány neurony v klasických umělých neuronových sítích, kde každý neuron je propojen s každým neuronem v sousedních vrstvách. Výstup z této vrstvy již není fotka, ale vektor.

Jako poslední aktivační funkce často bývá použita tzv. softmax funkce, která normalizuje výstup z neuronové sítě do rozložení pravděpodobností jednotlivých vstupních tříd. [6] V případě detekce vadných kusů ve výrobě jsou většinou třídy pouze 2 a to OK a NOK.



Obrázek 2.8: Konvoluční neuronová síť [26]

Při učení neuronových sítí je potřeba rozdělit dostupná data do 3 setů: učební (trénovací), validační a testovací. Na učebním setu probíhá samotné učení sítě. Validační set slouží k objektivnímu ohodnocení výkonu sítě při probíhajícímu učení. Testovací set slouží k objektivnímu ohodnocení výkonu finální podoby naučené sítě na datech, která dosud neviděla.[7]

Jedním ze způsobů vizualizace výsledků klasifikace neuronové sítě je chybová matice. Jedná se o matici o rozměrech $n \times n$, kdy n je počet tříd, tedy při detekci vadných kusů $n = 2$. V řádcích chybové matice jsou sítě predikované třídy a ve sloupcích jsou skutečné třídy. Pomocí chybové matice můžeme jednoduše vidět, jak dobře naučená síť je.[5] Je taky velice důležitá pro určení druhu chyby, kterou síť dělá. Právě u detekce vadných kusů je mnohem horší pokud síť prohlásí zmetek za OK kus (tzv. false positive) než když prohlásí zmetek za OK kus (tzv. false negative).

2.6 Deep learning toolbox

Deep learning toolbox v Matlabu umožňuje návrh a implementaci řešení problémů pomocí neuronových sítí. Je možné použít konvoluční neuronové sítě a Long short-term memory sítě na klasifikaci ze snímků, časových řad (Time series) a z textových dat.[21] Hlavní výhody a možnosti v Deep learning toolboxu jsou popsány v této kapitole.

		True Class	
		OK	NOK
Predicted Class	OK	TP	FP
	NOK	FN	TN

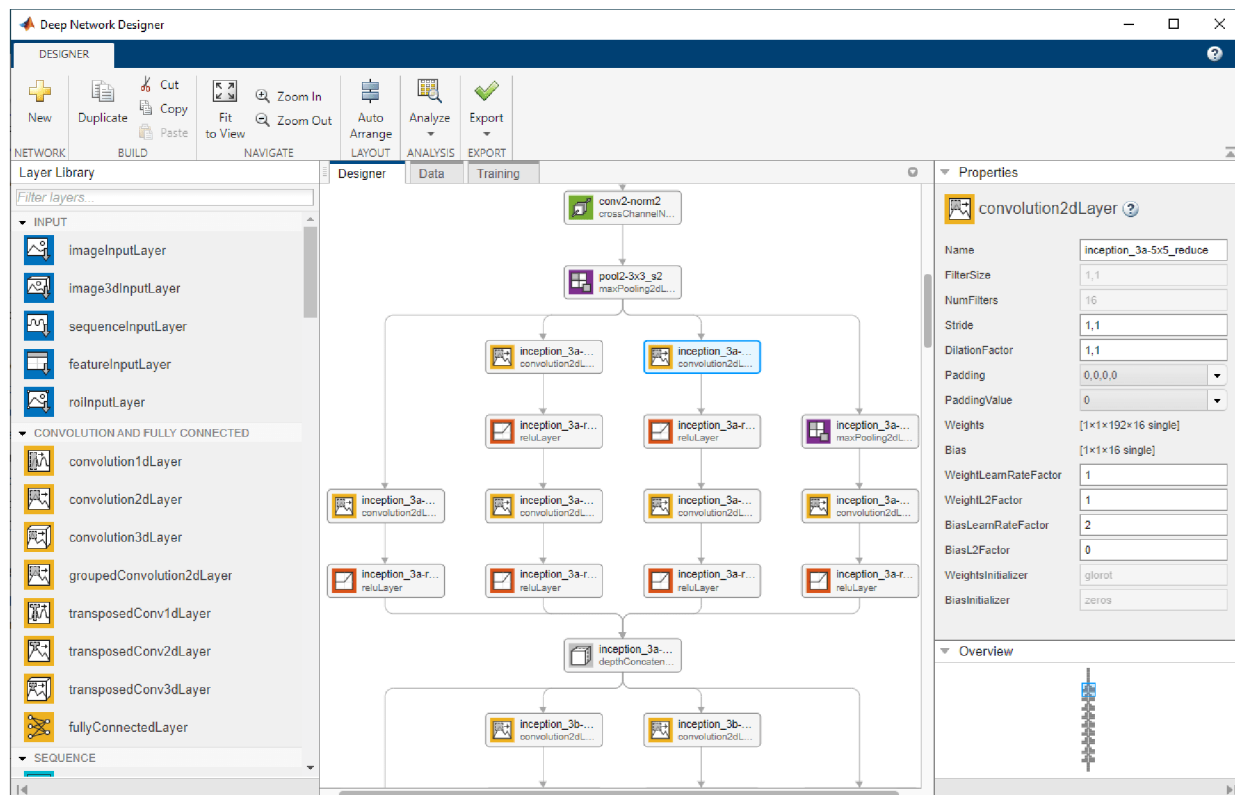
Obrázek 2.9: Chybová matice a) True positive b) False positive c) False negative d) True negative [25]

Deep learning Onramp je interaktivní tutoriál přímo od Mathworks na základy práce s Deep learning toolboxem. Jsou vyžadovány pouze základní znalosti práce s Matlabem, vše důležité pro začátek práce s hlubokým učení se v něm dá naučit. Uživatel se naučí jak používat již naučené sítě na klasifikaci ze snímků, jak pracovat s velkým množstvím dat a i jak učit sítě na nová data. Kromě toho je v průběhu tutoriálu vysvětlován i princip hlubokého učení, struktura neuronových sítí a jak funguje učení i klasifikace.

Deep learning toolbox umožňuje předpřipravit data pro další použití. Základem pro to je tzv. datastore, což je struktura umožňující práci s velkým množstvím dat. Funguje jako odkaz na kolekce dat, která jsou příliš velká na to, aby se vešla do paměti najednou. Základním typem datastore je "ImageDatastore". Pokud chceme provést s daty nějaké základní úpravy, jako například transformace z barevného na greyscale snímek, změnu velikosti nebo natočení či vystřížení části snímku, použijeme "AugmentedImageDatastore". Pro složitější úpravy je již nutno využít jiný Toolbox, například Image Processing Toolbox.[20]

Deep Network Designer je aplikace v Deep learning toolboxu, která umožňuje vytvářet, editovat a učit neuronové sítě bez nutnosti psaní kódu, pouze pomocí bločků. Tato aplikace se velmi hodí pro začátky práce s neuronovými sítěmi, neboť je velice přehledná a uživatelsky přívětivá. Je v ní jednoduše vidět struktura neuronové sítě a dá se jednoduše měnit pro potřeby uživatele. Při učení lze jednoduše sledovat průběh ztrátové funkce a validační úspěšnosti. Poté je možné vyexportovat naučené a upravené sítě do workspace a dále s nimi pracovat.[21]

Důležitou součástí Deep learning toolboxu je možnost provádět tzv. Transfer learning, při kterém je již naučená síť učena klasifikovat nový set snímků. Mathworks nabízí ke stažení množství předučených konvolučních neuronových sítí jako například GoogleNet, Resnet-50 a SqueezeNet. Ty byly trénovány databázi ImageNet nebo Places365. Databáze ImageNet obsahuje 1000 tříd objektů, jako klávesnice, letadlo, různá jídla a zvířata. Databáze Places365 obsahuje 365 kategorií míst, jako jsou pole, park, runway a chodba. K dispozici jsou také Long Short-term memory sítě na klasifikaci ze sekvencí dat.



Obrázek 2.10: Deep Network Designer

Při učení sítí na novém setu dat je nutné nejprve upravit v Deep Network Designeru závěrečné vrstvy sítě podle množství kategorií, na které chceme síť učit. Poté je potřeba upravit data podle toho, jaký vstup síť očekává. Některé pracují s barevnými snímky, jiné z grayscale snímky a každá síť očekává na vstupu jinou velikost snímku (často 224×224 pixelů). [20]

V kombinaci s dalšími toolboxy umožňuje MATLAB další složitější operace, ale ty již nebyly potřeba v této práci a výše popsané jsou ty nejdůležitější základní součásti.

3 Postup a výsledky řešení

V následujících kapitolách budou popsána data, která byla v práci použita, dále navržené metody detekce vad a klasifikace NOK kusů a jejich porovnání.

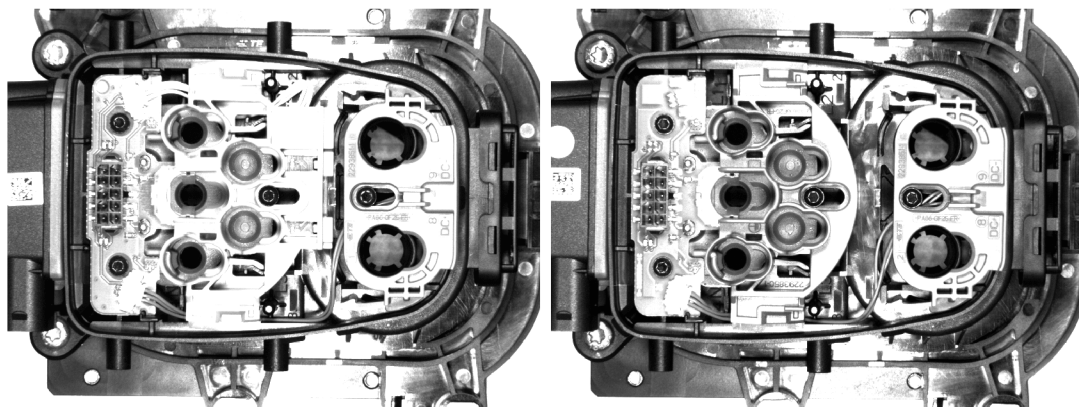
3.1 Použitá data

Metody detekce vad byly navrhovány na konektory pro nabíjení elektromobilů, poskytnuty firmou Mechatronic Design & Solutions. Jedná se o 3 sady dat. První dvě sady byly foceny kamerami na strojích při výrobě, z nichž jedna sada byla již roztríděná na OK a NOK, druhá sada fotografií byla neroztríděná. Třetí sada byla získána fotografováním hotových výrobků dodaných firmou ve světelném boxu.

První sada obsahuje snímky dvou druhů výrobků, tzv. kulatých a hranatých. Ty byly již roztríděny na OK a NOK a transformovány na grayscale.

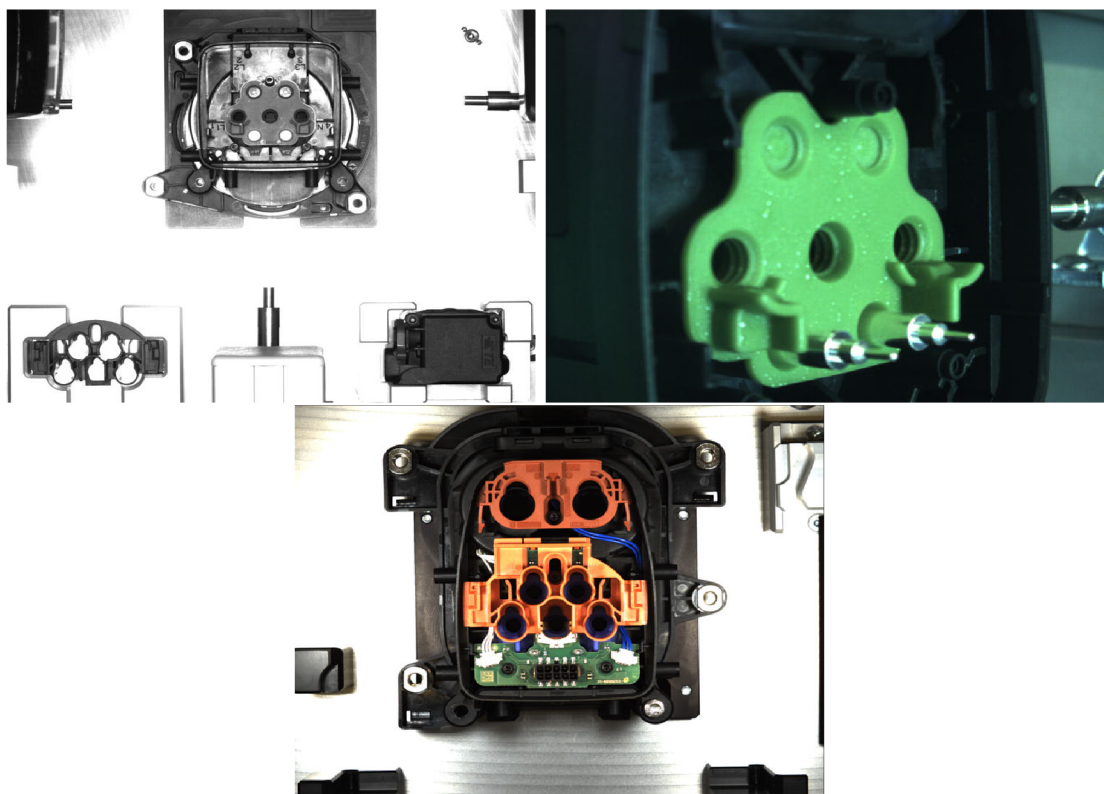
Typ výrobku	OK	NOK
Hranaté	698	76
Kulaté	2719	299

Tabulka 3.1: Počty kusů v první sadě



Obrázek 3.1: Hranatý a kulatý kus

Druhá sada obsahuje mnohem více typů výrobků. Kromě hotových kusů jako v první sadě obsahuje také fotografie detailů a fotografie z průběhu výroby. Tato sada byla určena k roztrídění pomocí profesionálního softwaru In-Sight explorer od firmy Cognex. Fotky v této sadě jsou rozděleny podle linky, na které byly vyráběny, a dále podle konkrétní kamery, která je při výrobě snímala.



Obrázek 3.2: a) Nesložený kus b) Detail z výroby c) Kompletní kus

3.1.1 Fotografování

Třetí sada dat obsahuje fotografie foceně autorem práce ve světelném boxu. Jedná se o 4 typy výrobků, Combo1 a Combo2, což jsou již hotové výrobky a Carrier1 a Carrier2, což jsou základní nosiče pro zbytek dílů. Focení probíhalo bez blesku v režimu makro. Fotoaparát byl upevněn do stativu a za pomoci závaží umístěn na vrchní stranu světelného

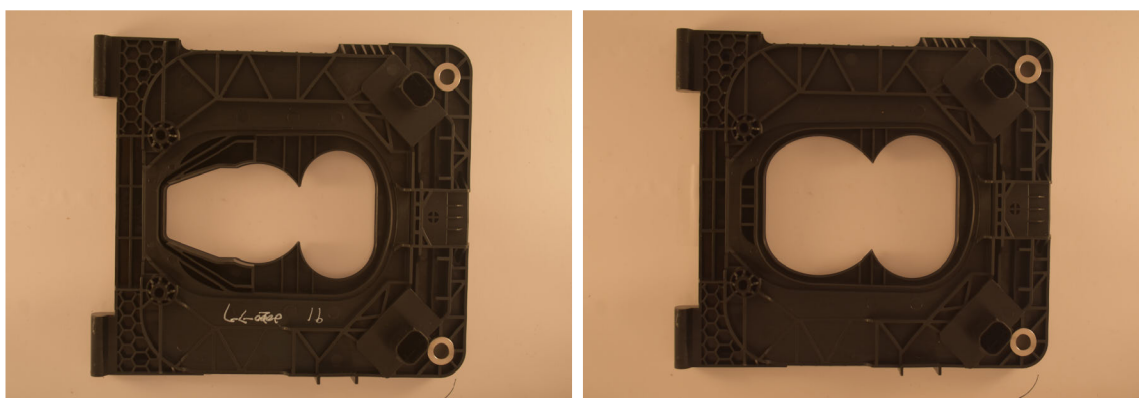
Model fotoaparátu	NIKON D3400
Závěrka clony	f/4.8
Délka expozice	1/400 sec.
Rychlost ISO	ISO-800
Expozice	0 krok
Ohnisková vzdálenost	36 mm
Maximální clona	4.5
Režim expozimetru	Maticové měření

Tabulka 3.2: Parametry fotoaparátu

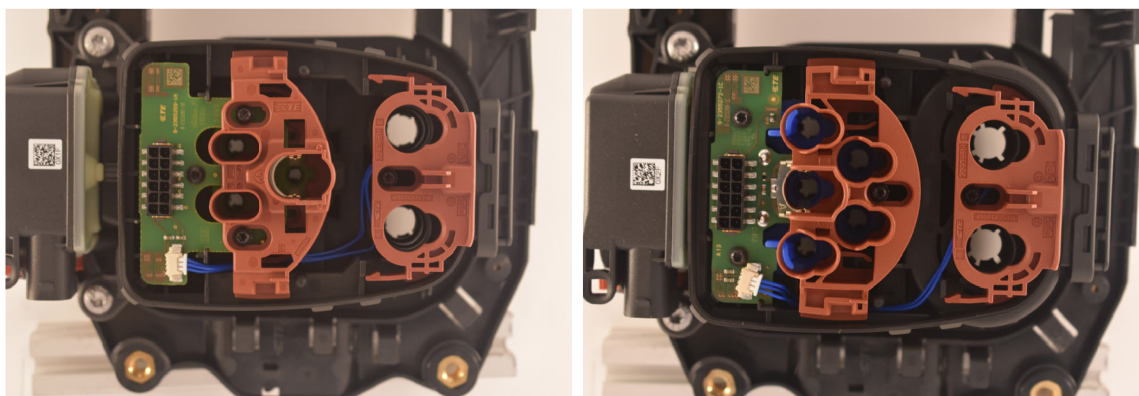
boxu. Osvětlení bylo zajištěno dvěma LED žárovkami o výkonu 8 W. Samotné mačkání spouště bylo řešeno softwarově pomocí programu Capture One Pro. V něm byla následně data i předzpracována.



Obrázek 3.3: Aranžmá při fotografování



Obrázek 3.4: Carrier1 a Carrier 2

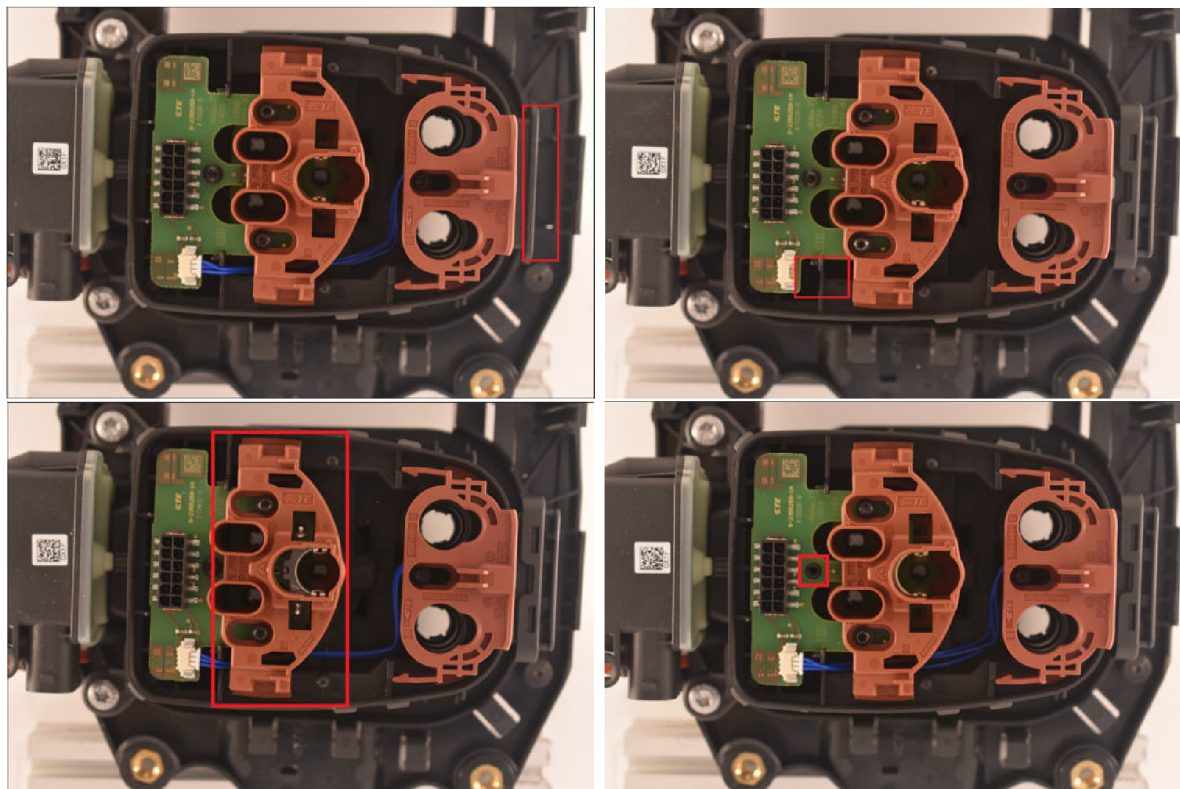


Obrázek 3.5: Combo1 a Combo2

Typ výrobku	OK	NOK
Carrier 1	36	12
Carrier 2	40	12
Combo 1	32	10
Combo 2	30	10

Tabulka 3.3: Počet nafocených kusů

Ukázalo se, že pro detekci vad neuronovými sítěmi je potřeba více NOK kusů. Proto bylo provedeno druhé focení, při kterém byly uměle vytvářeny vady na jednotlivých kusech tak, aby byly k dispozici fotky kusů se všemi kombinacemi vad. Problém s nedostatkem NOK kusů bude více popsán v odpovídající kapitole s neuronovými sítěmi. Jako základ byly použity 3 kusy, jeden OK, jeden s chybějícím drátkem a jeden s chybějícím dílem. Na nich byly postupně kombinovány různé chybějící šroubky a posunuté vrchní kryty. Tyto fotky byly poté přidány k původním. U carrierů nešlo provést žádné vratné vady, tím pádem jejich počet zůstal stejný.



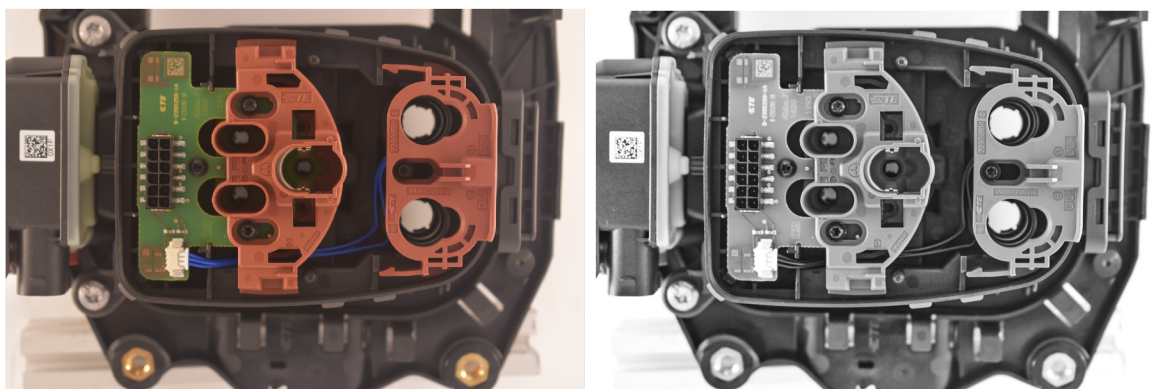
Obrázek 3.6: Vady na Combech. a) Chybějící díl vpravo b) Chybějící kabel c) Posun krytů d) Chybějící šroubky

Typ výrobku	OK	NOK
Carrier 1	36	12
Carrier 2	44	12
Combo 1	32	83
Combo 2	30	80

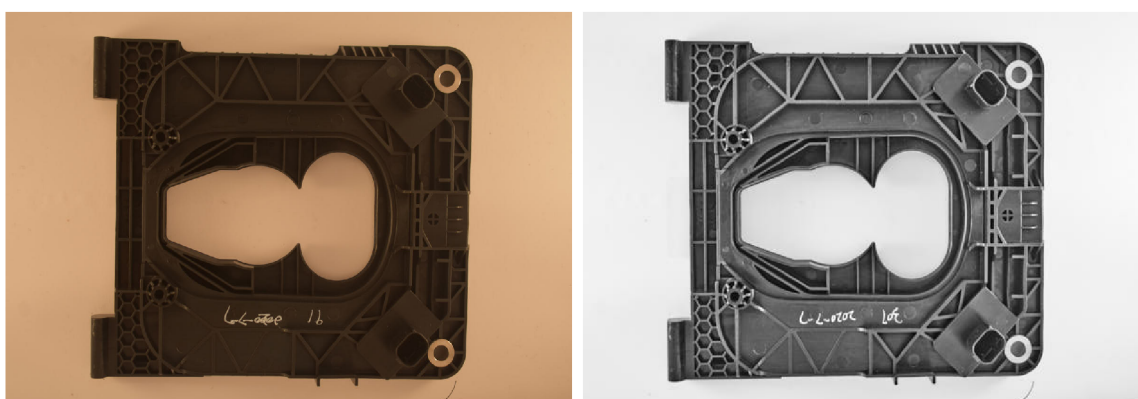
Tabulka 3.4: Počet nafocených kusů s uměle přidanými vadami

3.1.2 Předzpracování

Předzpracování byl stejně jako focení prováděn pomocí programu Capture One. Byly v něm provedeny operace popsané v kapitole 2.3 zejména transformace z RGB na grayscale a zvýšení kontrastu, tedy gamma korekce, za účelem zvýraznění hran a důležitých vlastností pro detekci vad. Úpravy byly provedeny tak, aby se výsledné fotky přiblížily těm ze sady č.1 z kamer z výroby.



Obrázek 3.7: Combo2 před a po zpracování



Obrázek 3.8: Carrier2 před a po zpracování

3.2 Vzorové experimenty

Jedním z cílů práce bylo zopakovat vzorové experimenty se zadaným softwarem, což byl program In-Sight Explorer od společnosti Cognex. Tento software nabízí dvě rozhraní - tabulkové, jež slouží k přesnému naprogramování žádaných parametrů a EasyBuilder, ve kterém lze vytvořit jednoduchý program bez nutnosti programování. Ať už v tabulkovém rozhraní, nebo v EasyBuilderu, pro každý typ výrobku je nutné vytvořit nový tzv. Job, který potom spustíme na požadovanou sadu dat a ten by měl detekovat, který z výrobků je NOK a určit i konkrétní vadu. Software používá metody klasického strojového vidění

INSPECTION MANAGEMENT												
Step	Step for	Enable	Enable in	Enable	Result	New Part	0.000	Result	Mask	Input PLC	Force	Result to PLC
2	Inspector	Inspector	this Step	Plot		Soft 0				No Input		
secondary lock AC	2	1		1	1		secondary lock AC	4	1	0	round	1
secondary lock DC	2	1		1	1		secondary lock DC	7	1	0	<input checked="" type="checkbox"/>	1
screw 1	1	1		1	1		screw 1	8	1	0	<input checked="" type="checkbox"/>	1
screw 2	1	1		1	1		screw 2	7	1	0	<input checked="" type="checkbox"/>	1
screw 3	1	1		1	1		screw 3	9	1	0	<input checked="" type="checkbox"/>	1
screw 4	1	1		1	1		screw 4	9	1	0	<input checked="" type="checkbox"/>	1
actuator f	2	0		0	1		actuator f	6	0	0	<input type="checkbox"/>	0
actuator 90	2	1		1	1		actuator 90	4	1	0	<input checked="" type="checkbox"/>	1
frame	1	0		0	1		frame	0	0	0	<input type="checkbox"/>	0
cap	1	1		1	1		cap	9	1	0	preset	1
part status							part status	1				
screw_5,6	1	1		1	1		screw_5,6	5	1	0	<input checked="" type="checkbox"/>	1
connector 1	2	1		1	1		connector 1	7	1	0	<input checked="" type="checkbox"/>	1
connector 2	2	0		0	1		connector 2	4	0	0	<input type="checkbox"/>	0
not used							not used	0				
counter					1		counter	178		178		
							no Input		1			
							no Input or Force		1			
							0.000					

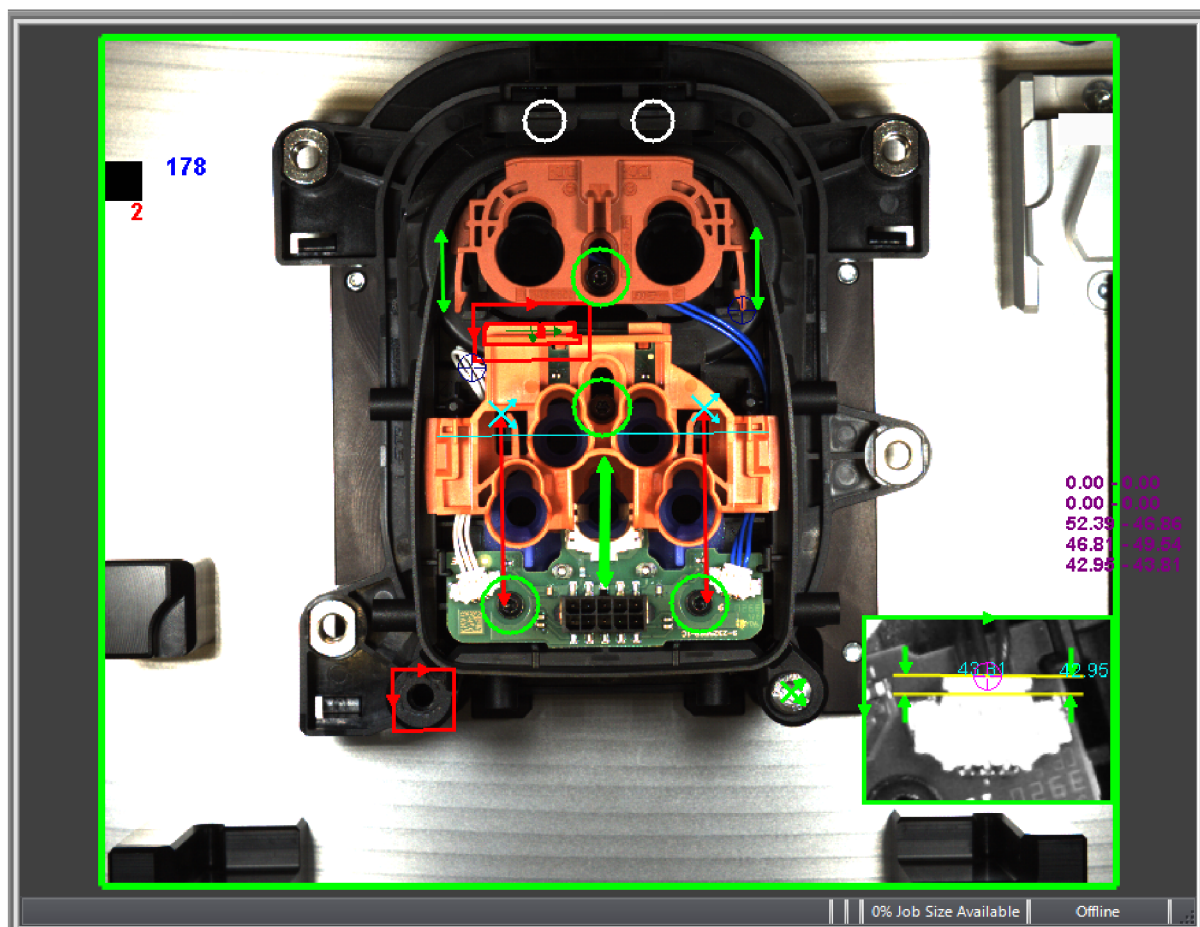
Obrázek 3.9: Ukázka tabulkového zobrazení

popsané výše. Původní myšlenkou bylo použít software k rozřídění druhé sady dat na OK a NOK, aby je bylo možno využít následně při dalším návrhu metod detekce vad. Software nefunguje bez připojené kamery, což bylo řešeno emulátorem. Poté bylo nutné vložit správnou sadu dat a k nim vložit odpovídající Job. Poté Job projde vybranou sadu dat a program jej rozřídí na OK a NOK (V terminologii Cognexu "good" a "bad").

Co ovšem nebylo dopředu oznámeno je, že firma MDS používá tento software pouze pro nalezení a spočítání parametrů. Ty jsou potom posílány na PLC, kde pak dochází k samotnému rozhodnutí, zda je výrobek v pořádku, či ne. Samotný program tedy je schopen pouze najít a vypočítat zadané hodnoty, už není schopen určit, který výrobek je OK a který je NOK. Stroje, na kterých byly tyto výrobky vyráběny již opustily MDS, a tak nebylo možné se k nim dostat. Z tohoto důvodu nebylo možné druhou sadu dat rozřídít a dále využít.

3.3 Detekce exaktními metodami

Prvním navrženým přístupem detekce vad je pomocí exaktních metod. Byly využity postupy klasického strojového vidění, stejně jako u vzorového softwaru. Detekce probíhala v programu MATLAB, zejména za pomoci Image processing Toolboxu. Tato metoda byla navrhována na první set dat, tedy rozříděné fotky kulatých a hranatých výrobků. Nejprve bylo potřeba určit, jaké vady se na výrobcích vyskytují a dle toho navrhnout, jakým



Obrázek 3.10: Ukázka zobrazení EasyBuilder

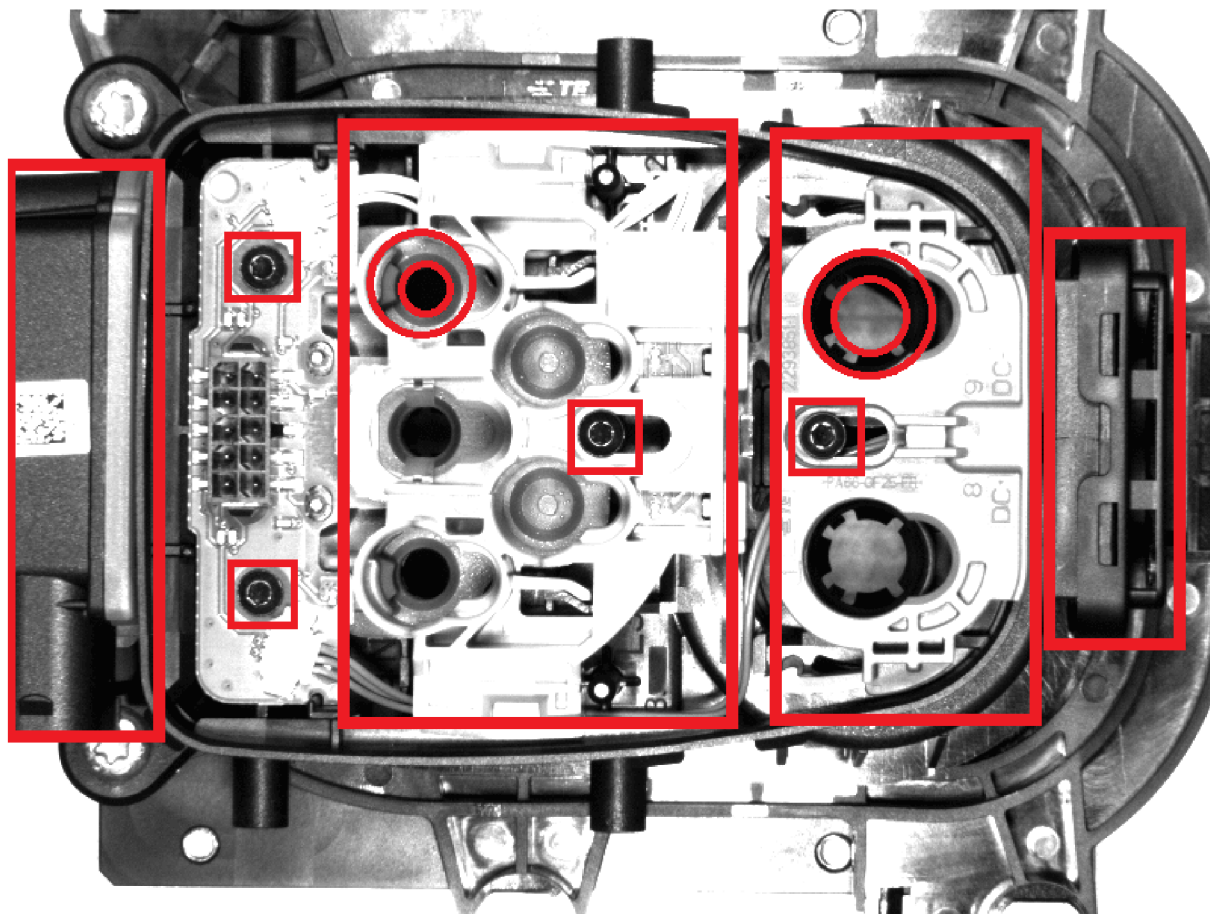
způsobem kontrolovat které části výrobku.

Na kulatých i hranatých kusech se vyskytovaly tyto vady: chybějící šroubky, posunutý pravý či levý horní kryt a chybějící díly na obou stranách výrobku. U každé fotky tedy muselo být ověřeno, zda se na ní nenachází jedna z těchto vad.

V MATLABU byly vytvořeny 3 funkce, jedna pro detekci chybějících šroubků, jedna pro detekci posunutých krytů a jedna pro detekci chybějících dílů. Vstupem do funkce je vždy datastore obsahující odkazy na všechny fotky v konkrétní sadě. Vzhledem k tomu, že data již byla předem rozdělená, tak sady byly OK a NOK. Výstupem každé funkce je vektor o velikosti odpovídající počtu fotek v datastoru. Na pozici odpovídající číslu výrobku, u kterého byla vada odhalena obsahuje jedničku, na těch, kde daná vada odhalena nebyla obsahuje nulu. Poté byly výsledné vektory zkombinovány a byl určen konečný počet NOK kusů v dané sadě. Hlavním cílem bylo nejprve správně označit všechny NOK výrobky a poté redukovat množství false positive, tedy OK označených jako NOK, na co nejmenší hodnotu.

Detekce chybějících šroubků

První popsaná funkce je na detekci chybějících šroubků. Na výrobcích buď byly všechny šroubky, nebo ani jeden, stačilo tedy ověřovat pouze jeden šroubek. Nejprve byla vystřižena část fotky, kde se kontrolováný šroubek nachází. Poté byl použit příkaz na hledání



Obrázek 3.11: Sledované vlastnosti na OK kusu



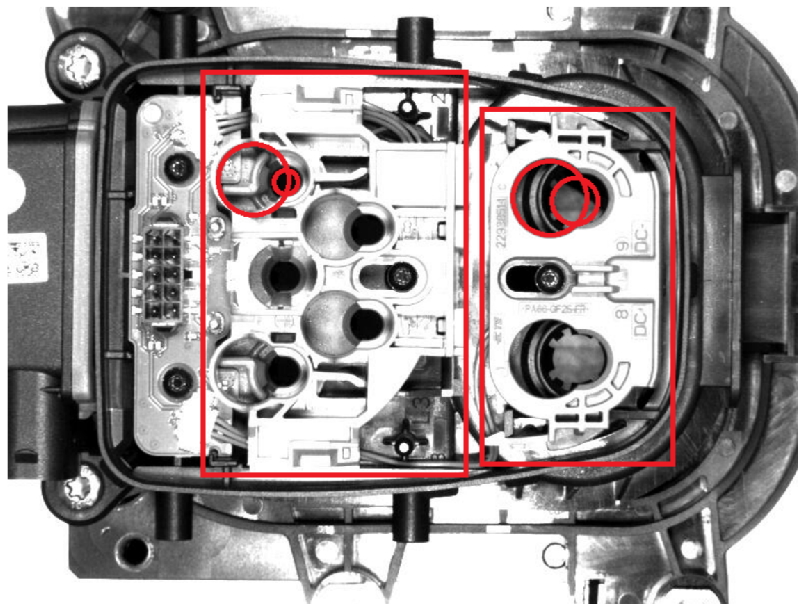
Obrázek 3.12: Chybějící šroubek

kruhů na fotce. Ten za pomoci prahování a hledání hran hledá na fotce kruhy dle nastavených parametrů. Ty byly nastaveny tak, aby hledaly díru po chybějícím šroubku, nikoliv samotný šroubek. Vstupem do příkazu je vystřížená část fotky a výstupem je poloha středu a poloměr nalezeného kruhu. Pokud příkaz našel kruh, tedy díru, chybí na kusu šroubek a kus je tedy označen jako NOK. Takto funkce projde v cyklu všechny fotky a tam, kde díru po šroubku najde, je zapsána jednička do výsledného vektoru.

Detekce posunutých krytů

Druhá funkce kontrolovala posun horních krytů. Fotka byla rozdělena na dvě části a v každé byl kontrolován posun jednoho krytu. Posunutí byla kontrolována opět pomocí příkazu na hledání kruhů. V každém z krytů se nachází otvory. Nejprve byla nalezena poloha všech otvorů u všech OK i NOK kusů. Konstrukce výrobku umožňovala významný posun pouze v ose x, tím pádem se otvory pohybovaly spolu téměř stejně a bylo potřeba ověřit polohu pouze jednoho z nich u pravého i levého krytu (Otvory vyznačené na obrázku

2.13). Všechny x souřadnice polohy byly vyneseny do grafu a podle toho byla určena hranice, za kterou již byly všechny kusy považovány za NOK. Bylo tím potřeba vyloučit i některé OK kusy, ale s tím se při výrobě počítá. NOK nad hranicí jsou ty, u kterých byly detekovány jiné vady než posun krytů.

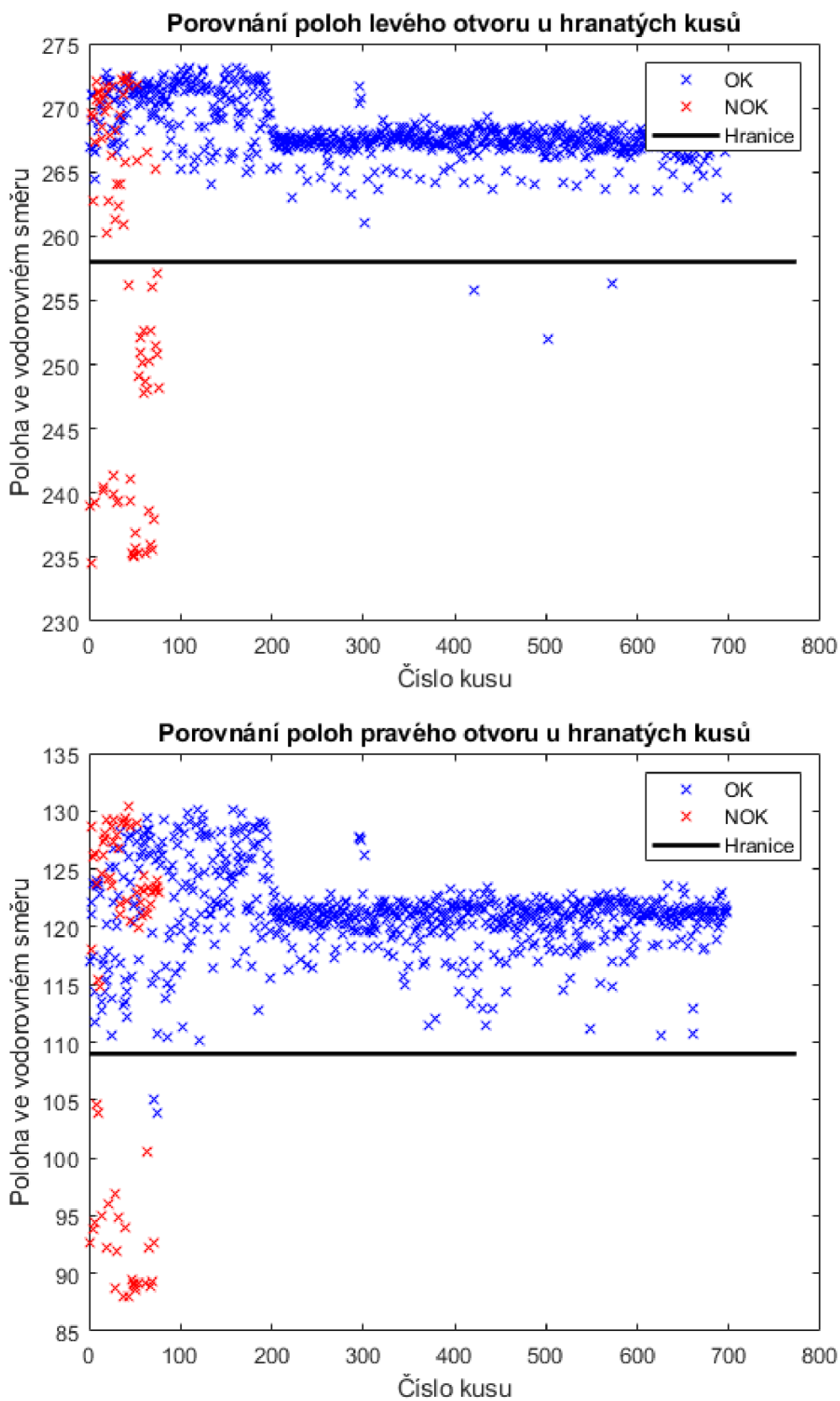


Obrázek 3.13: Posunuté kryty

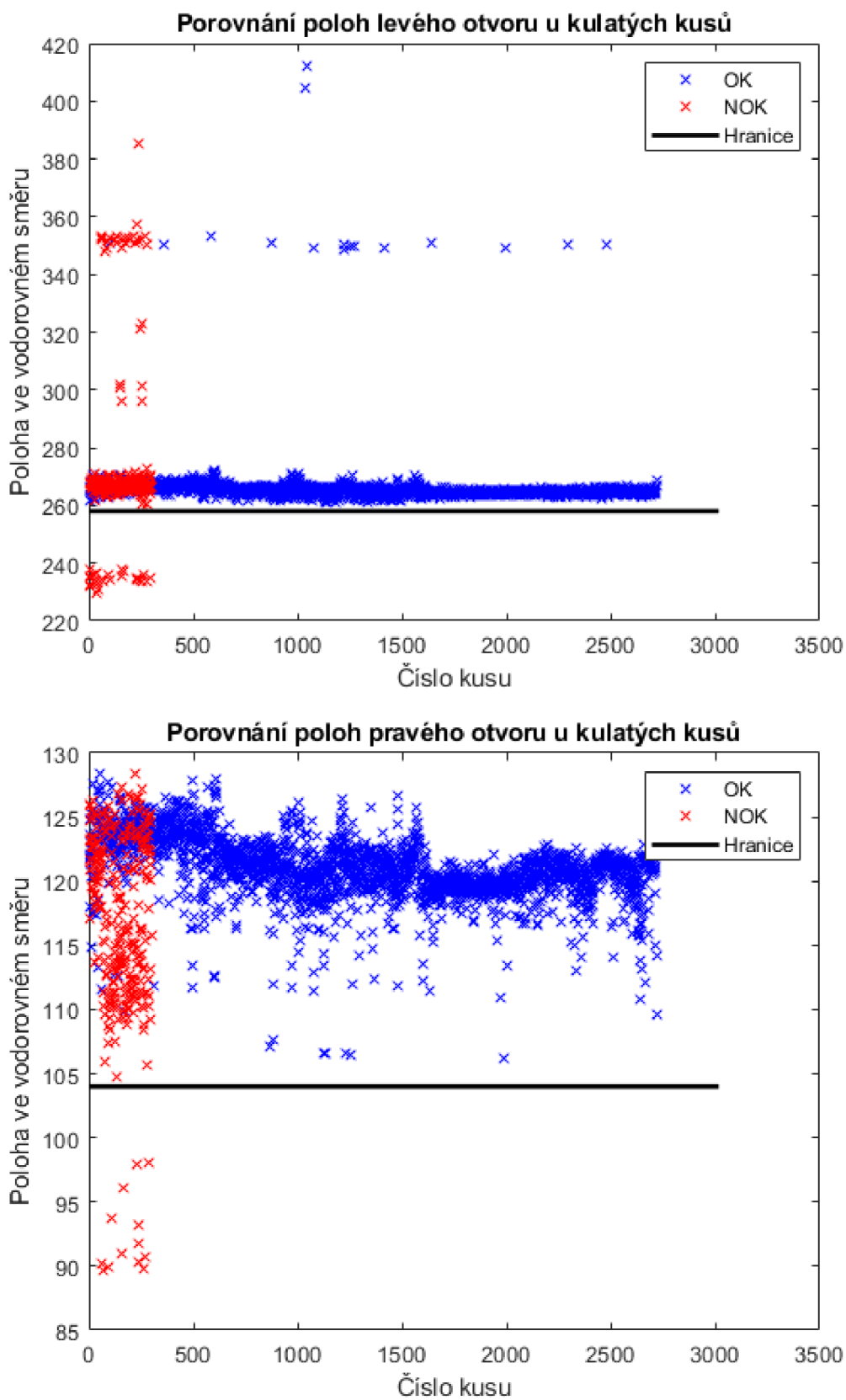
Detekce chybějících dílů

Poslední funkce detekovala chybějící díly na pravé i levé straně. Detekce na levé straně byla jednoduchá. Pokud díl na výrobku chyběl, výsledná intenzita pixelů v oblasti byla a nebo se blížila maximální hodnotě 255. Pokud díl byl na svém místě, hodnoty se pohybovaly kolem hodnoty 80. Stačilo tedy vybrat malou oblast v místě, kde se díl nachází a zprůměrovat hodnoty intenzity v této oblasti. Pokud průměr přesáhl hranici nastavenou na 150, byl kus označen jako NOK. Díl na pravé straně byl složitější, pod chybějícím dílem byl další s podobnými hodnotami intenzity. Pro detekci vady byl zvolen obdélníkový otvor ve správně umístěném dílu. Ten vytváří stín s nízkými hodnotami intenzity. Pokud díl chybí, žádný stín se v této oblasti neobjeví a hodnoty intenzity jsou tak vyšší.

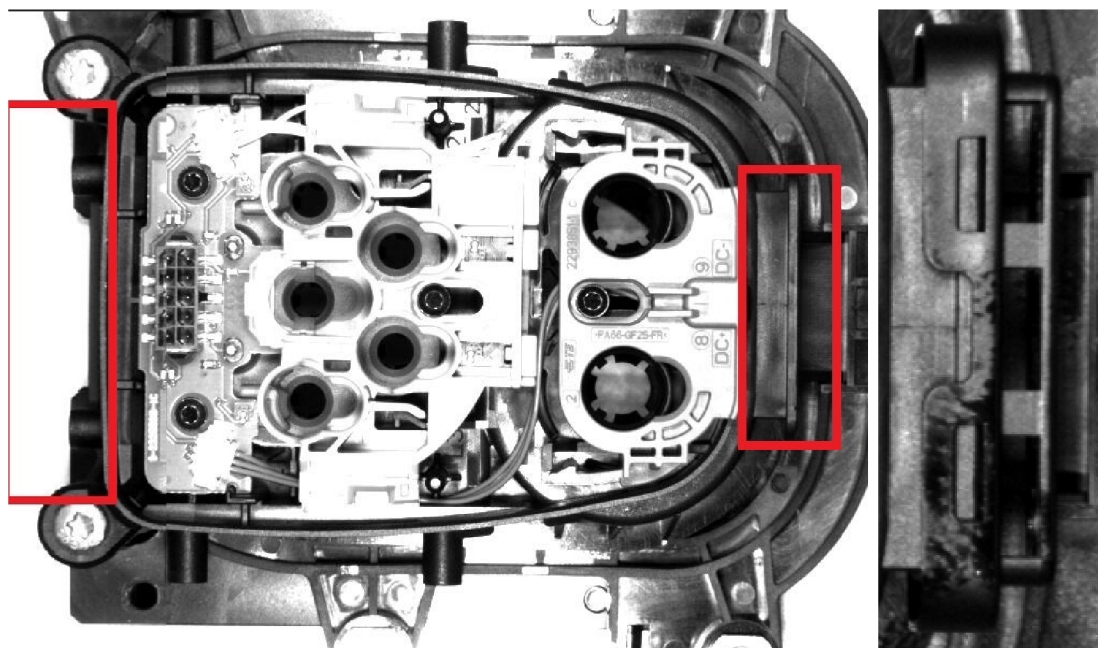
U 4 kulatých kusů se na pravém díle ještě vyskytly oděrky, kvůli čemuž byly taky klasifikovány jako NOK. Tyto oděrky byly tmavší než zbytek dílu, a tak byl spočítán počet pixelů, které mají hodnoty pod stanovenou hranicí 50, a když počet těchto pixelů překročil 34, bylo stanoveno, že je přítomna oděrka a kus je NOK. Toto bylo ověřováno jen pokud nebyl detekován chybějící pravý díl, na kterém se oděrky vyskytovaly, proto je v grafu zobrazeno méně NOK kusů.



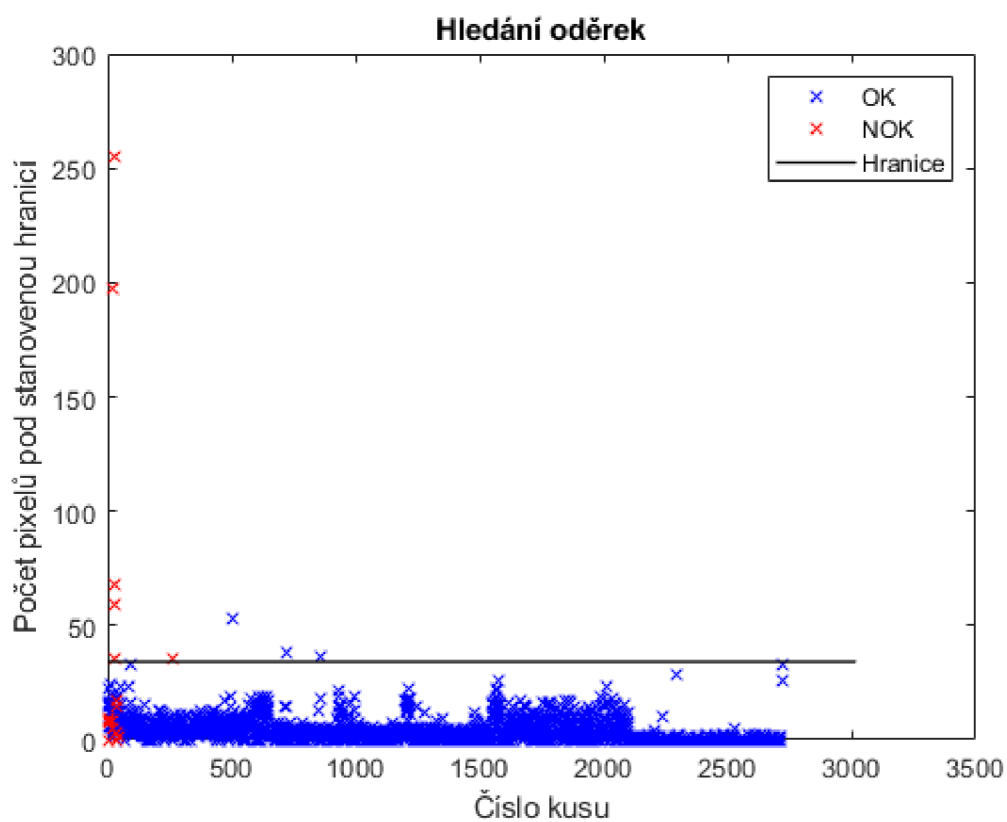
Obrázek 3.14: Grafy pro detekci posunutých krytů-hranaté



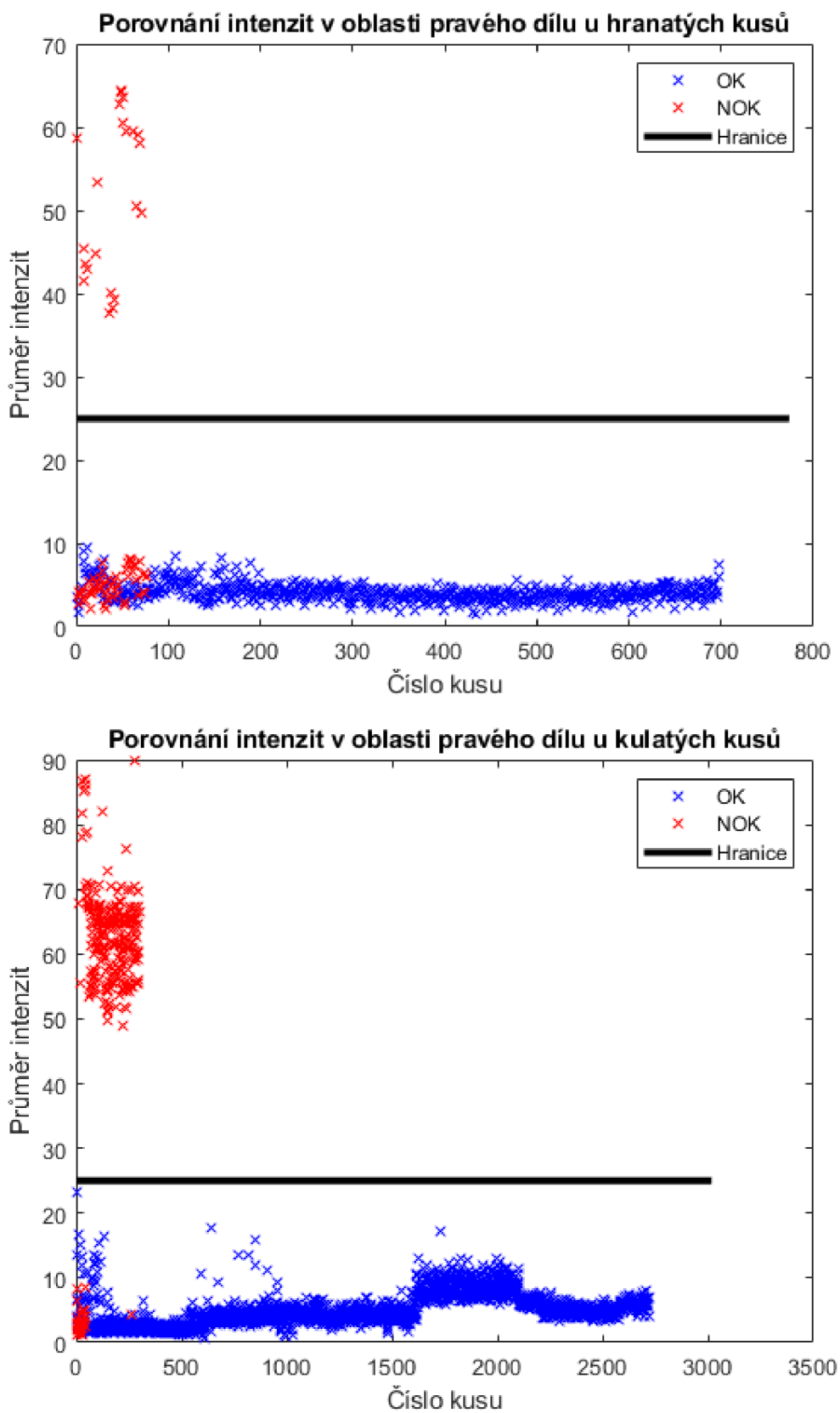
Obrázek 3.15: Grafy pro detekci posunutých krytů-kulaté



Obrázek 3.16: a) Chybějící díly b) Oděrky



Obrázek 3.17: Graf pro hledání oděrek



Obrázek 3.18: Grafy pro chybějící díly

Projít celou sadu kulatých zabralo algoritmu průměrně 360 s (315 sekund OK a 45 sekund NOK), celou sadu hranatých prošel za 107 sekund (95 sekund OK a 12 sekund NOK). Výsledné statistiky jsou zapsány v tabulkách níže. Algoritmus nedokázal odhalit

	Kulaté	Hranaté
Chybějící šroubek	2	11
Posunuté kryty	40	52
Chybějící díl	286	25

Tabulka 3.5: Počty detekovaných vad

	Kulaté	Hranaté
NOK detekováno	299	75
False positive	0	1
NOK detekováno/NOK	1,0000	0,9868
False negative	3	4
False negative/OK	0,0011	0,0115

Tabulka 3.6: Finální zhodnocení

jeden NOK kus u hranatých výrobků. Při bližším prozkoumání u něho nebyla nalezena žádná vada. Je tedy možné, že se jedná o false negative, který byl špatně klasifikován při roztřídění dat na firmě.

3.4 Detekce neuronovými sítěmi

Druhá navržená metoda detekce byla pomocí neuronových sítí. Detekce probíhala v MATLABU primárně využitím deep learning toolboxu. Cílem bylo, aby sítě správně klasifikovaly všechny NOK a OK kusy, jednotlivé vady nebyly rozpoznávány.

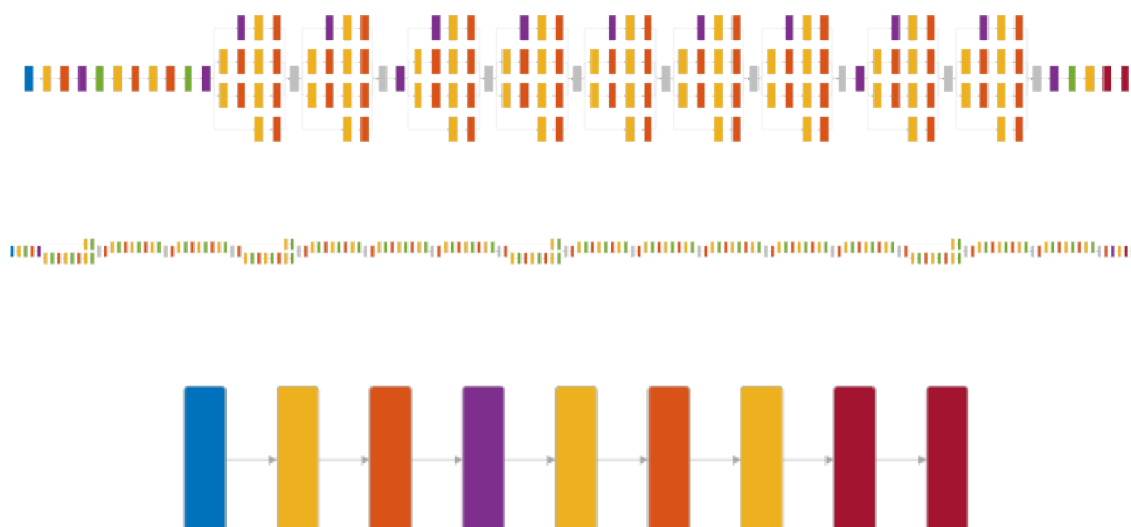
Pro klasifikaci byly použity 3 konvoluční neuronové sítě, "GoogleNet" a "Resnet50", které byly již k dispozici v MATLABU, a v Deep network designeru nově vytvořená síť pracovně nazvaná "BasicNet". Všechny sítě mají velmi odlišnou architekturu. Googlenet obashuje mnoho paralelních vrstev, zatímco Resnet50 je téměř celá v jedné řadě. BasicNet je oproti nim mnohem jednodušší, skládající se pouze z několika vrstev. Jednotlivé vrstvy, ze kterých jsou sítě složeny jsou popsány v kapitole 2.5.6.

Učení sítí probíhalo na stroji Acer Nitro 5 AN515-54 CPU: Intel Core i5-9300H, GPU: NVIDIA GeForce GTX 1650 4GB, RAM: 8GB DDR4. Učení probíhalo na jednotce GPU díky parallel computing toolboxu.

Největší vliv na úspěšnost klasifikace sítě měl hyperparametr Initial Learning Rate, hlavním úkolem tedy bylo najít ideální hodnotu tohoto parametru pro každou síť a pro každou sadu dat. Z celé sady dat bylo odebráno 20 % na konečné testování. Zbylá data byla rozdělena v poměru 0,8 ku 0,2 na trénovací a validační.

Všechny snímky byly upraveny tak, jak to vyžadovaly sítě. Všechny očekávaly velikost snímku 224×224 pixelů. GoogleNet a Resnet50 pracovaly s RGB snímky, BasicNet s grayscale snímky.

Počet epoch učení byl nastaven podle průběhu validační přesnosti a ztrátové funkce. Pokud přesnost stoupala a ztrátová funkce klesala i v momentu, kdy došlo k ukončení



Obrázek 3.19: Struktury použitých sítí a) GoogleNet b) Resnet50 c) BasicNet

učení, počet epoch pro další učení byl zvýšen. Pokud došlo k tomu, že se hodnoty přestaly zlepšovat a naopak se začaly zhoršovat, počet epoch pro další učení byl snížen.

Vždy bylo zvoleno několik hodnot ILR a učení bylo provedeno pro každou hodnotu. Poté byly zaznamenávány vždy: celková přesnost klasifikace, přesnost u NOK kusů, doba učení a doba klasifikace. Podle toho, pro které hodnoty ILR byla síť úspěšná, byly zvoleny hodnoty pro další učení. Takto se postupně přibližovalo k ideálnímu nastavení, dokud nedosáhla síť přesnosti u NOK 100 %, neboť to je průmyslová přesnost. Druhotně se kontrolovala celková přesnost, protože občas síť rozpoznala správně všechny NOK kusy, ale naopak nebyla schopna dobře rozpoznat OK kusy. Za předpokladu správné klasifikace všech NOK kusů je povolena celková přesnost nad 99 %.

ILR	Přesnost	Přesnost NOK	Čas učení	Čas kvalifikace
0,1	0,8957	0	118,3	5,4
0,01	0	0	81,5	3,8
0,001	0,9007	0	77,7	3,3
0,0001	0,9652	0,65	75,5	3,7
0,00001	0,9901	0,9333	81,5	4,1
0,000001	0,9901	1	78,3	4,1
0,0000001	0,9884	0,8833	75,8	3,7

Tabulka 3.7: Příklad opakovaného učení sítě BasicNet na kulaté kusy

Dále budou popsány testy podle dat, na kterých se síť učily.

Kulaté a hranaté kusy

V tomto testu byly síť učeny na kulaté a hranaté díly z prvního setu.

Kulaté	ILR	Přesnost	Přesnost NOK	Čas učení	Čas klasifikace
BasicNet	0,000001	0,9901	1,0000	78,3	4,1
GoogleNet	0,000005	0,9917	0,9167	170,6	6,1
Resnet50	0,0008	0,9934	0,9333	258,4	7,6

Tabulka 3.8: Nejlepší dosažené výsledky na kulatých kusech

Hranaté	ILR	Přesnost	Přesnost NOK	Čas učení	Čas klasifikace
BasicNet	0,000004	0,9935	0,9333	108,6	0,9
GoogleNet	0,00086	0,9935	0,9333	191,1	1,4
Resnet50	0,007	0,9935	1,0000	125,2	2,7

Tabulka 3.9: Nejlepší dosažené výsledky na hranatých kusech

Požadované přesnosti 100% na NOK kusech se podařilo dosáhnout pouze sítí Resnet50 na hranatých dílech a BasicNet na kulatých dílech. Problém byl, že počet NOK kusů na výsledné testování byl u hranatých kusů pouze 15 a pokud tedy nebyl odhalen jeden NOK kus, přesnost šla rapidně dolů. Tento problém byl řešen v dalších testech namnožením dílů.

Namnožení dat

Pro namnožení dat byl použit tzv. transformed datastore. Každý datastore obsahující originální sady byla několikrát nakopírován a na každý snímek byla aplikována úprava, aby se vytvořil defacto nový kus. Tím byl znásoben počet kusů jak pro učení, tak pro závěrečný test, a tak bylo možno přesněji vyhodnotit, jak dobře je síť schopna NOK kusy rozpoznat.

Základní úprava byla drobné posunutí a rotace snímku, simulující různě umístěný díl při výrobě. Pokud by síť byly schopny takto kusy bezpečně klasifikovat, byly by přidány další úpravy, simulující například náhodný stín na snímku, náhodnou změnu jasu či náhodný šum.

Celkově byl počet kusů znásoben dvanáctkrát, tedy závěrečný počet kulatých kusů byl 36216 a hranatých 9288. To samozřejmě značně zvýšilo dobu a výkon, potřebný k učení sítě. Z důvodu nedostatečného výkonu, již nebylo možno použít síť Resnet50.

hranaté	ILR	Přesnost	Přesnost NOK	Čas učení	Čas klasifikace
BasicNet	0,00001	0,9941	0,9890	1150,1	21,5
GoogleNet	0,00059	0,9995	0,9945	1233,7	25,5

Tabulka 3.10: Nejlepší dosažené výsledky na hranatých kusech

Síť GoogleNet dosáhla přesnosti u NOK kusů 99,45 % na hranatých kusech. To už nám dává mnohem přesnější informaci o schopnosti sítě. Vzhledem k tomu, že procentuální rozdělení dat zůstalo stejné, pro závěrečný test bylo použito 180 NOK kusů. Dosažená úspěšnost tedy znamená, že síť nebyla schopna klasifikovat pouze 1 NOK kus. BasicNet nerozeznala 2 NOK kusy. Výsledky u kulatých jsou o poznání horší, ovšem kvůli dlouhému trvání učení a jeho náročnosti nebylo možno provést dostatečné množství testů. Je možné, že by šlo dosáhnout i lepších výsledků.

kulaté,36000,rot,pos	ILR	Přesnost	Přesnost NOK	Čas učení	Čas klasifikace
basicnet	0,00001	0,9977	0,9805	2695,1	111,8
googlenet	0,0001	0,9979	0,9791	3452,7	132,9

Tabulka 3.11: Nejlepší dosažené výsledky na kulatých kusech

Test s nafocenými kusy 1

V tomto testu byly sítě učeny na kusy nafocené ve světelném boxu v první várce, to znamená pouze původní kusy bez uměle vytvořených vad. Hned ze začátku byl objeven problém s Carriery. Každý kus je na povrchu popsán a nejspíš nešťastnou náhodou došlo k tomu, že OK a NOK kusy jsou každý popsány na jiné straně. Sítě tak potom byly schopné bezpečně rozeznat OK a NOK kusy pouze podle popisu a jejich další testování nemělo smysl.

Testování Comb probíhalo 2 způsoby. Oba začaly tím, že byl vytvořen datastore obsahující všechny OK a NOK snímky jednoho druhu výrobku, například Combo1. První způsob: Vytvořený datastore byl nejprve nakopírován, aby bylo dosaženo většího množství snímků. V tomto případě byl nakopírován dvacetpětkrát, výsledný počet snímků byl tedy 1000. Tento datastore byl poté rozdělen na trénovací, validační a testovací sety ve stejném poměru jako v předchozích testech. Druhý způsob: Vytvořený původní datastore byl jako první rozdělen na trénovací, validační a testovací sety. Každý z těchto setů byl poté dvacetpětkrát nakopírován, takže výsledný počet snímků byl opět 1000. Při prvním způsobu byla tedy síť učena na všech původních snímcích, při druhém způsobu při učení neviděla ty původní snímky, které byly odebrány do testovacího setu.

Při prvním způsobu se sítě byly schopny naučit bez problémů všechny vady a poté správně všechny NOK kusy klasifikovat. Sítě byly schopny všechny NOK kusy rozeznat i při dalších úpravách snímků, například přidání čtvercového stínu na náhodné místo, náhodná úprava jasu či náhodný Gaussův šum. Ovšem kvůli malému počtu původních NOK kusů nelze příliš jednoznačně určit reálnou schopnost sítě vady detekovat.

Při druhém způsobu byl problém to, že každý NOK kus obsahuje jinou vadu. Tím, že sítě neviděly některé původní kusy znamená, že vůbec neviděly některé vady a proto nebyly schopny tyto NOK kusy detekovat. Z toho důvodu, byla výsledná přesnost vždy 50 % nebo 75 % podle toho, jak byly zrovna snímky náhodně rozděleny.

Test s nafocenými kusy 2

Problém s množstvím NOK kusů byl řešen umělým vytvořením vad, popsaným v kapitole 3.1. Testování probíhalo opět 2 způsoby, jako v předchozí sekci.

Při testování prvním způsobem (sítě při učení viděly všechny původní snímky) se ukázalo, že obě sítě byly opět schopny rozeznat všechny NOK kusy i při všech úpravách jako v předchozím testu. Díky výrazně většímu množství NOK kusů však je možné lépe hodnotit schopnosti sítí při detekci. Znamená to tedy, že při dostatečném množství snímků s dostatečným zastoupením všech vad, by neuronové sítě měly být schopny bezpečně klasifikovat všechny NOK kusy.

Při testování druhým způsobem (sítě při učení neviděly snímky určené na závěrečné testování) přesnost značně stoupla v porovnání s testem bez uměle vytvořených vad, ale 100 % se dosáhnout nepodařilo. Přesnost kolísala i při stejném parametru ILR a oproti ostatním testům značně kolísala i u OK kusů. Je možné, že to bylo způsobeno nepoměrem

dat ve prospěch NOK kusů.

Ostatní a nepovedené testy

Bylo předpokládáno, že postupné snižování parametru ILR v průběhu učení povede k lepším výsledkům, ale překvapivě nebylo možno nalézt konfiguraci, kdy by snižování parametru vedlo ke srovnatelným, natož lepším výsledkům než při optimálně nastaveném neměnném ILR. Efekt snižování ILR by se nejspíš pozitivně projevil až při řádově větším množství dat a počtu epoch učení.

Dále bylo vyzkoušeno změnit parametr hybnost (momentum) u algoritmu SGDM, jehož základní hodnota v MATLABU je 0.9, ale opět to nevedlo k lepším výsledkům.

Při jednom z testů byly sítě nejprve naučeny na kulaté kusy a poté přeučovány na hranaté. Byly očekávány lepší výsledky, ale jak u GoogleNet, tak u BasicNet bylo dosaženo horších výsledků, než při učení od nuly.

Bylo plánováno vyzkoušení komplexnější sítě InceptionNet, ale kvůli nedostatku výkonu to nebylo možno provést.

4 Závěr

Hlavním cílem této práce bylo navrhnout 2 metody vizuální detekce vad. Jednu s využitím strojového učení a druhou založenou na libovolném jiném přístupu. Měla být experimentálně ověřena jejich schopnost detekce vad a schopnost vyrovnat se profesionálnímu softwaru In-Sight explorer od společnosti Cognex. Další cíl práce byl zopakovat zadané experimenty s tímto softwarem.

Pro přístup s využitím strojového učení byla zvolena jeho odnož, hluboké učení. K detekci vad tak byly použity konvoluční neuronové sítě. Jako druhý přístup byly zvoleny exaktní metody, které detekují vady pomocí zpracování obrazu a přesně nalezených a spočítaných parametrů. Obě metody byly navrhovány a implementovány v MATLABU.

Metody detekce byly navrhovány a testovány na konektorech na nabíjení elektroaut, které poskytla firma Mechatronic Design & Solutions. Konkrétně se jedná o 3 sady snímků, v první jsou již hotové kusy roztríděné na OK a NOK, druhá je neroztríděná a obsahuje i fotky z průběhu výroby. V třetí sadě byly hotové výrobky focené autorem práce ve světelném boxu.

V rámci vzorových experimentů měl být použit software In-Sight Explorer k roztrídění druhé sady dat na OK a NOK. Podařilo se úspěšně detekovat vady na zadaných snímcích, ovšem kvůli způsobu, kterým se program v MDS využívá, neuměl určit, kdy je kus OK a kdy NOK, to obstarávají až PLC. Z toho důvodu nebylo možné porovnat navržené metody přímo oproti softwaru.

V automobilovém průmyslu se očekává, že budou detekovány všechny zmetky a toho se podařilo dosáhnout oběma přístupy. Oba způsoby mají své výhody a nevýhody a výrazně odlišný postup při návrhu a aplikaci.

Základní rozdíl při návrhu přístupů byl ten, že pro implementaci exaktní metody bylo potřeba znát, jak má OK kus vypadat a jaké vady je třeba detekovat. U každé vady bylo potom nutné navrhnout, jakým způsobem bude detekována. Výhodou tedy je, že je možné přesně určit, jaká vada se na výrobku vyskytuje. Neuronové sítě ukáží pouze to, zda se jedná OK či NOK kus.

Neuronové sítě potřebují velké množství dat, aby se mohly dostatečně dobře naučit rozpoznat OK a NOK kusy a zejména u NOK kusů je občas těžké tato data získat. Při návrhu exaktních metod nejsou potřeba téměř žádné hotové kusy, stačí od zákazníka vědět, jak má zadaný výrobek vypadat a jaké jsou potenciální tolerance u určitých vlastností. V této práci byly ovšem hotové kusy potřeba z důvodu nastavení hranic u hledaných vad.

Ukázalo se, že obdržené množství NOK dat není pro použití neuronových sítí dostačující. Na závěrečné testování zůstalo příliš málo NOK kusů, a tak nebylo možné s dostatečnou přesností určit, jak dobře sítě vady detekovaly. Proto bylo potřeba data namnožit. V této práci byla data množena kopírováním celé sady dat s úpravami každé fotky, jako bylo natočení a posunutí, což simulovalo různé umístění výrobku. Takto bylo možné lépe posoudit, jak dobře byly sítě naučené a nejlepší výsledky se pohybovaly nad 98 %. Výsledky učení ovšem byly omezeny dostupným výkonem. S větším výkonem by nejspíš

4 ZÁVĚR

bylo možné dosáhnout lepších výsledků. Bylo by možné využít komplexnější a přesnější sítě, použít ještě větší množství dat a zkrátila by se doba učení. Na stejné sadě dat bylo exaktními metodami dosaženo 100 %. Z toho plyne, že při menším dostupném výkonu se jeví jako vhodnější exaktní metody, které nejsou tak náročné na výkon.

Na hotových výrobcích byly uměle přidávány vady při focení. Na těchto namnožených datech sítě dosáhly 100 % úspěšnosti v rozpoznávání NOK kusů, a dokonce bez jediného false negative. Zdá se tedy, že pokud mají sítě dostatek vstupních dat s každou vadou, jsou schopny se je dobře naučit rozpoznat. Jsou schopny vyřešit například i náhodný stín či šum. S náhodným stínem by si exaktní metody neporadily. Jsou mnohem závislejší na důsledně stejných světelných podmínkách a kvalitě fotografie. U neuronových sítí na kvalitě fotky až tolik nezáleží, většina očekává vstupní snímek o velikosti 224×224 pixelů. Kvůli tomu ale můžou splynout některé důležité detaily či vady, které by potom síť nemusela odhalit.

Vzorové experimenty byly zopakovány, obě požadované metody detekce byly navrženy a experimentálně porovnány. Všechny cíle práce se tedy dají považovat za splněné.

Literatura

- [1] ŠÍMA, Jiří a Roman NERUDA. Teoretické otázky neuronových sítí. Praha: MATFYZ press, 1996, 390 s. ISBN 80-85863-18-9.
- [2] KENJI SUZUKI. Artificial Neural Networks - Methodological Advances and Biomedical Applications. 1. Chicago: IntechOpen, 2011. ISBN 9533072431. Dostupné z: doi:10.5772/644
- [3] BROWNIEE, Jason. A Gentle Introduction to the Rectified Linear Unit (ReLU). Machine Learning Mastery [online]. 2019 [cit. 2022-05-05]. Dostupné z: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>
- [4] NIELSEN, Michael. Neural Networks and deep Learning [online]. Determination Press, 2015 [cit. 2022-05-05]. Dostupné z: <http://neuralnetworksanddeeplearning.com/>
- [5] Confusion Matrix. DeepAI [online]. [cit. 2022-05-06]. Dostupné z: <https://deepai.org/machine-learning-glossary-and-terms/confusion-matrix>
- [6] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
- [7] BROWNIEE, Jason. What is the Difference Between Test and Validation Datasets [online]. 2017 [cit. 2022-05-06]. Dostupné z: <https://machinelearningmastery.com/difference-test-validation-datasets/>
- [8] Nondestructive Flaw Detection in Metallic Components. Quality Magazine [online]. 2015 [cit. 2022-05-08]. Dostupné z: <https://www.qualitymag.com/articles/92780-nondestructive-flaw-detection-in-metallic-components>
- [9] DU PLESSIS, Anton, Igor YADROITSEV, Ina YADROITSAVA a Stephan G. LE ROUX. X-Ray Microcomputed Tomography in Additive Manufacturing: A Review of the Current Technology and Applications. 3D Printing and Additive Manufacturing. 2018, 5(3), 227-247. ISSN 2329-7662. Dostupné z: doi:10.1089/3dp.2018.0060
- [10] ROCHA, Tiago J, Helena Geirinhas RAMOS, A LOPES RIBEIRO a Dário J PASADAS. Magnetic sensors assessment in velocity induced eddy current testing. Sensors and actuators. A. Physical [online]. LAUSANNE: Elsevier B.V, 2015, 228, 55-61 [cit. 2022-05-17]. ISSN 0924-4247. Dostupné z: doi:10.1016/j.sna.2015.02.004

- [11] What Is MPI (Magnetic Particle Inspection)?. The Graduate Engineer [online]. 2021 [cit. 2022-05-17]. Dostupné z: <https://thegraduateengineer.com/what-is-mpi-magnetic-particle-inspection/>
- [12] REN, Zhonghe, Fengzhou FANG, Ning YAN a You WU. State of the Art in Defect Detection Based on Machine Vision. International Journal of Precision Engineering and Manufacturing - Green Technology [online]. Seoul: Korean Society for Precision Engineering, 2021, 9(2), 661-691 [cit. 2022-05-17]. ISSN 2288-6206. Dostupné z: doi:10.1007/s40684-021-00343-6
- [13] CALABRESE, Luigi a Edoardo PROVERBIO. A Review on the Applications of Acoustic Emission Technique in the Study of Stress Corrosion Cracking. Corrosion and Materials Degradation [online]. Melbourne: MDPI, 2021, 2(1), 1 [cit. 2022-05-17]. ISSN 2624-5558. Dostupné z: doi:10.3390/cmd2010001
- [14] CHAKI, Jyotismita a Nilanjan DEY. A Beginner's Guide to Image Preprocessing Techniques. 1. Milton: CRC Press, 2019. ISBN 9781138339316. Dostupné z: doi:10.1201/9780429441134
- [15] Image Processing 101 Chapter 1.3: Color Space Conversion. Dynamsoft [online]. 2019 [cit. 2022-05-17]. Dostupné z: <https://www.dynamsoft.com/blog/insights/image-processing/image-processing-101-color-space-conversion/>
- [16] YANG, Jing, Shaobo LI, Zheng WANG, Hao DONG, Jun WANG a Shihao TANG. Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges. Materials [online]. Switzerland: MDPI, 2020, 13(24), 1-23 [cit. 2022-05-17]. ISSN 1996-1944. Dostupné z: doi:10.3390/ma13245755
- [17] Introduction to Machine Vision: A guide to automating process & quality improvements [online]. Cognex, 2018 [cit. 2022-05-08]. Dostupné z: <https://www.cognex.com/resources/white-papers-articles/introduction-to-machine-vision-17152>
- [18] Deep Learning vs. Machine Vision [online]. Cognex, 2020 [cit. 2022-05-17]. Dostupné z: <https://www.cognex.com/resources/white-papers-articles/deep-learning-vs-machine-vision-ebook>
- [19] Datastores for Deep Learning. MathWorks [online]. [cit. 2022-05-10]. Dostupné z: <https://www.mathworks.com/help/deeplearning/ug/datastores-for-deep-learning.html>
- [20] Deep Network Designer. MathWorks [online]. [cit. 2022-05-10]. Dostupné z: <https://www.mathworks.com/help/deeplearning/ref/deepnetworkdesigner-app.html>
- [21] Deep Learning Toolbox: Design, train, and analyze deep learning networks. MathWorks [online]. [cit. 2022-05-10]. Dostupné z: <https://www.mathworks.com/products/deep-learning.html>
- [22] Introduction to Vision Sensors [online]. Cognex [cit. 2022-05-10]. Dostupné z: <https://www.cognex.com/resources/white-papers-articles/introduction-to-vision-sensors>

- [23] SEDLÁK, Jan. Rozpoznávání vozidel pomocí metod strojového učení [online]. Brno, 2016 [cit. 2022-05-17]. Dostupné z: <https://is.muni.cz/th/lot2e/>. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Tomáš BRÁZDIL.
- [24] Hassan, Hassan Negm, Abdelazim Zahran, Mohamed & Saavedra, Oliver. (2015). ASSESSMENT OF ARTIFICIAL NEURAL NETWORK FOR BATHYMETRY ESTIMATION USING HIGH RESOLUTION SATELLITE IMAGERY IN SHALLOW LAKES: CASE STUDY EL BURULLUS LAKE. *International Water Technology Journal*. 5.
- [25] Confusion Matrix for Your Multi-Class Machine Learning Model. In: *Towards Data Science* [online]. 2020 [cit. 2022-05-18]. Dostupné z: <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>
- [26] A CNN sequence to classify handwritten digits. In: *Towards Data Science* [online]. 2018 [cit. 2022-05-06]. Dostupné z: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Seznam obrázků

2.1	Druhy osvětlení	13
2.2	Kontrola jasu v dané oblasti	15
2.3	Hledání hran	15
2.4	Kontrola jasu v dané oblasti	16
2.5	Detekce hran	16
2.6	Model neuronu	17
2.7	Neuronová síť	18
2.8	Konvoluční neuronová síť	20
2.9	Chybová matice	21
2.10	Deep Network Designer	22
3.1	Hranatý a kulatý kus	23
3.2	Snímky z výroby	24
3.3	Aranžmá při fotografování	25
3.4	Carrier1 a Carrier 2	25
3.5	Combo1 a Combo2	25
3.6	Vady na Combech	26
3.7	Předzpracování Combo	27
3.8	Předzpracování Carrier	27
3.9	In-Sight tabulka	28
3.10	In-Sight EasyBuilder	29
3.11	OK kus	30
3.12	Chybějící šroubek	30
3.13	Posunuté kryty	31
3.14	Grafy-detekce posunutých krytů-hranaté	32
3.15	Grafy-detekce posunutých krytů-kulaté	33
3.16	Chybějící díly a oděrky	34
3.17	Graf-oděrky	34
3.18	Grafy-Chybějící díly	35
3.19	Struktury použitých sítí	37

Seznam tabulek

3.1	Počty kusů v první sadě	23
3.2	Parametry fotoaparátu	24
3.3	Počet nafocených kusů	26
3.4	Počet nafocených kusů s uměle přidanými vadami	27
3.5	Počty detekovaných vad	36
3.6	Finální zhodnocení	36
3.7	Příklad opakovaného učení sítě BasicNet na kulaté kusy	37
3.8	Nejlepší dosažené výsledky na kulatých kusech	38
3.9	Nejlepší dosažené výsledky na hranatých kusech	38
3.10	Nejlepší dosažené výsledky na hranatých kusech	38
3.11	Nejlepší dosažené výsledky na kulatých kusech	39

Seznam zkratek

NDT Non-destructive techniques (Nedestruktivní metody)

CNN Convolutional Neural Network (Konvoluční neuronová síť)

ILR Initial Learning Rate

SGDM Stochastic gradient descent with momentum (Stochastický gradientní sestup se setrvačností)

ReLU Rectified Linear Unit (Usměrněná lineární jednotka)

NOK Not OK

LED Light emitting diode (Elektroluminiscenční dioda)

CPU Central processing unit (Centrální procesorová jednotka)

GPU Graphics processing unit (Centrální grafická jednotka)

MDS Mechatronic Design & Solutions

PLC Programmable logic controller (Programovatelný logický automat)