Časová náročnost nasazení neuronových sítí pro detekci vad

Jiří Slovák

Bakalářská práce
2024

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Jiří Slovák**
Osobní číslo:	**A21396**
Studijní program:	**B0714A150006 Aplikovaná informatika v průmyslové automatizaci**
Specializace:	**Inteligentní systémy s roboty**
Forma studia:	**Prezenční**
Téma práce:	**Časová náročnost nasazení neuronových sítí pro detekci vad**
Téma práce anglicky:	**Time complexity of deploying neural networks for defect detection**

Zásady pro vypracování

Zadání:

1. Popište základní principy neuronových sítí používaných pro detekci vad na základě kamerového obrazu.
2. Vytvořte přehled dostupných neuronových sítí pro detekci vad.
3. Sestavte množinu snímků, která bude sloužit pro testování neuronových sítí.
4. Otestujte vybrané neuronové sítě pro detekci vad. Zaměřte se především na úspěšnost vyhodnocení časové náročnosti.
5. Proveďte celkové zhodnocení vybraných neuronových sítí pro použití při detekci vad.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. KŘIVAN, Miloš. *Úvod do umělých neuronových sítí*. Vyd. 3., přeprac. Praha: Oeconomica, 2014. ISBN 9788024520247.
2. PATTERSON, Josh a GIBSON, Adam. *Deep learning: a practitioner's approach*. Sebastopol, CA: O'Reilly Media, 2017. ISBN 9781491914236. Dostupné také z: http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&AN=1564784&authtype=ip,shib&custid=s3936755
3. GRAUPE, Daniel. *Deep learning neural networks: design and case studies*. New Jersey: World Scientific, [2016]. ISBN 9789813146457.
4. FORSYTH, David a PONCE, Jean. *Computer vision: a modern approach*. Upper Saddle River: Prentice Hall, [2003]. ISBN 0130851981.
5. BUDUMA, Nikhil. *Fundamentals of deep learning: designing next-generation machine intelligence algorithms*. Beijing: O'Reilly, 2017. ISBN 9781491925614.
6. LU, Huimin a LI, Yujie. *Artificial intelligence and computer vision*. Studies in computational intelligence. Cham: Springer, [2016]. ISBN 9783319462455.

Vedoucí bakalářské práce: **Ing. Petr Chalupa, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **8. prosince 2023**
Termín odevzdání bakalářské práce: **27. května 2024**

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
DĚKAN
L.S.

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 8. prosince 2023

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Jiří Slovák, v.r.
podpis studenta

ABSTRAKT

Cílem bakalářské práce je zkoumat a porovnat vlastnosti a funkce několika průmyslových nástrojů využívajících technologii neuronových sítí pro vizuální detekci. V první části se bakalářská práce zabývá teoretickými principy fungování neuronových sítí s přesahem do umělé inteligence. Včetně rozdělení, využití a obecným popisem neuronových sítí. Druhá část se zaměřuje na praktické porovnání konkrétních průmyslových neuronových sítí, v oblastech jako je schopnost učení, časová náročnost učení a přesnost rozpoznávání. U každého testu jsou prezentovány a diskutovány výsledky. Na konec jsou shrnuty získané poznatky a doporučení pro praktické využití v dané oblasti aplikace. Tato práce přispívá k lepšímu porozumění této technologie a ukazuje možnosti a omezení neuronových sítí v praktických aplikacích.

Klíčová slova: neuronové sítě, hluboké učení, detekce vad, strojové učení, strojové vidění.

ABSTRACT

The aim of the bachelor thesis is to investigate and compare the features and functions of several industrial tools using neural network technology for visual detection. The first part of the bachelor thesis deals with the theoretical principles of neural networks with an overlap to artificial intelligence. Including classification, applications and general description of neural networks. The second part focuses on a practical comparison of specific industrial neural networks, in areas such as learning ability, learning time and recognition accuracy. For each test, the results are presented and discussed. Finally, the lessons learned are summarized and recommendations for practical use in the application domain are presented. This work contributes to a better understanding of the technology and demonstrates the capabilities and limitations of neural networks in practical applications.

Keywords: neural networks, deep learning, defect detection, machine learning, machine vision.

Rád bych poděkoval Ing. Vlastimilovi Čablovi a jeho kolegům z firmy AMV Technology s.r.o. za sdílení know-how v oblasti implementace neuronových sítí pro rozpoznávání obrazu a za poskytnutí softwaru a hardwaru potřebného k práci s touto technologií. Dále bych rád poděkoval mému vedoucímu práce, panu Ing. Petru Chalupovi, Ph.D za jeho vedení a cenné rady, které mi poskytoval po celou dobu zpracování této práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

ÚVOD

Neuronové sítě se v posledních letech staly klíčovou komponentou v oblasti umělé inteligence (AI) a strojového učení. Jejich schopnost učit se z dat, adaptovat se na měnící se prostředí a zpracovávat komplexní informace, z nich činí důležitý nástroj s širokou škálou aplikačních možností v moderním technologickém světě. Tato bakalářská práce se zaměřuje na komplexní popis principů fungování neuronových sítí a jejich praktických aplikací, s důrazem na srovnání vybraných nástrojů.

Teoretická část práce poskytuje ucelený přehled základních principů neuronových sítích a jejich fungování. Jsou zde diskutovány základní principy fungování neuronů, struktura a architektura neuronových sítí, algoritmy učení a optimalizace, oblasti využití této technologie a také problematika strojového učení obecně.

Praktická část práce prezentuje výsledky experimentů zaměřených na otestování výkonu a schopností vybraných nástrojů využívajících technologii neuronových sítí a hlubokého učení. Tato srovnání jsou provedena na základě různých metrik a testovacích sad dat, čímž umožňují komplexní hodnocení efektivity a účinnosti jednotlivých architektur při řešení specifických úkolů spjatých s vizuální kontrolou snímků. Praktická část práce tak přináší cenný vhled do praktického využití neuronových sítí pro různé aplikace vizuální kontroly z průmyslu.

Cílem této bakalářské práce je tedy poskytnout ucelený pohled na fungování neuronových sítí a jejich aplikace a přispět k lepšímu porozumění jejich potenciálu a omezení v praxi.

I. TEORETICKÁ ČÁST

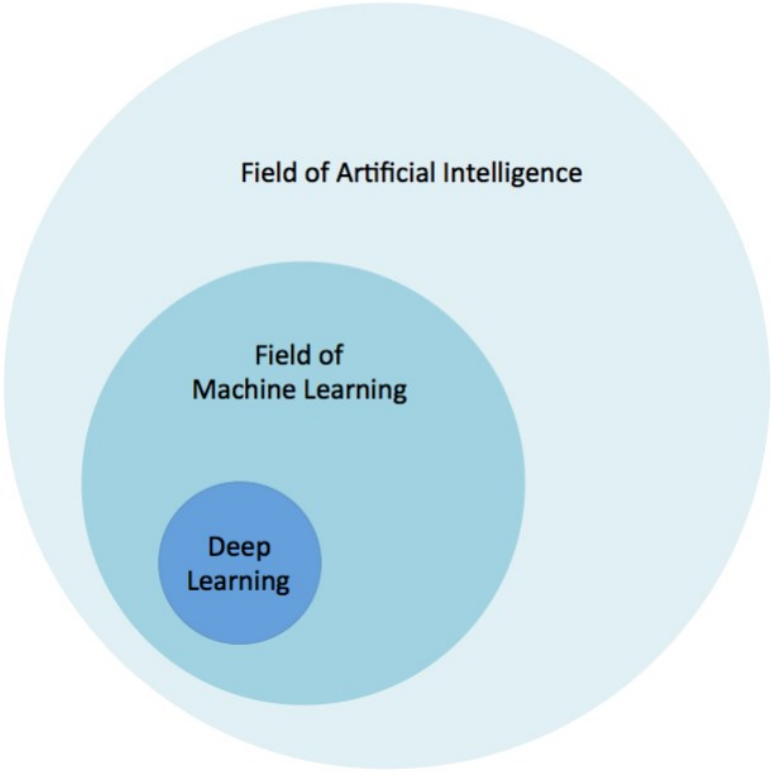
1 UMĚLÁ INTELIGENCE

V posledních letech dochází k nebývalému rozvoji umělé inteligence, poháněnému zejména pokrokem v oblasti hardwaru a dramatickým nárůstem výkonu počítačových procesorů. Tato kapitola se zaměřuje na základní principy funkce umělé inteligence a detailní analýzu jejich klíčových podmnožin, strojového učení a hlubokého učení. Dále budou popsány základní algoritmy a metody, které se v procesu učení AI používají.[1; 2; 3]

AI představuje interdisciplinární vědní obor, jehož primárním cílem je vytvářet systémy, které projevují vlastnosti podobné lidské inteligenci. Zaměřuje se na vývoj technologií a metod, které umožňují konstruovat systémy schopné řešit problémy analogicky k lidskému přístupu. Jednou z hlavních úloh AI je automatizace intelektuálních úkolů, které doposud vyžadovaly lidský zásah, jako je zpracovávání kancelářské agendy, investování na burze nebo psaní kódu. [1; 2; 4]

Dříve byla široce přijímána představa, že lidská mysl může být replikována pomocí explicitně definovaných pravidel, obvykle ve formě rozsáhlých souborů podmínek if-else. Tyto systémy, známé jako expertní systémy, se osvědčily při řešení jednoduchých logických úkolů s omezeným počtem možných řešení. Nicméně, s narůstající složitostí úkolů začaly tyto systémy velmi rychle ztrácet na účinnosti.

Se vzrůstající potřebou řešit problémy v oblasti identifikace obrazu se objevila nutnost hledat nová řešení, neboť aplikace explicitních pravidel se ukázala jako neproveditelná. Proto vznikly moderní koncepty umělé inteligence, které díky svým algoritmům dokážou adaptovat svá vnitřní pravidla rozhodování pro řešení problémů a optimalizovat je tak, aby daný problém dokázaly řešit.[1; 4; 5]



Obrázek 1. Umělá inteligence, strojové a hluboké učení[4]

1.1 Strojové učení

Neuronové sítě jsou úzce spojeny se strojovým učením a představují konkrétní model implementace tohoto paradigmatu, prakticky jsou jeho podmnožinou.

Strojové učení je oblast AI zaměřená na vytváření systémů, které se učí z dat a dokážou se bez explicitního programování zlepšovat ve svých úkonech. Strojové učení zahrnuje různé algoritmy jako je supervizované učení (učení s učitelem), nesupervizované učení (učení bez učitele) a posilující učení.

Strojové učení se zabývá využíváním algoritmů k odvození strukturálních popisů systémů. To znamená, že počítačový systém se snaží získat pochopení o struktuře dat, které jsou mu prezentovány v podobě surových nezpracovaných dat. Strukturální popis je pojmem používaným k označení modelů, které jsou vytvořeny na základě agregace informací z těchto surových dat a poté jsou využity k predikci neznámých hodnot. [4; 5]

Existuje řada různých modelů, z nichž každý aplikuje určitá pravidla na surová data s cílem predikovat neznámé hodnoty. Jedním z takových modelů je rozhodovací strom, který vytváří

hierarchii pravidel ve formě stromové struktury. Dalším příkladem je model lineární regrese, který se zaměřuje na nalezení souboru parametrů, jež nejlépe reprezentují vstupní data.[4]

Je důležité si uvědomit, že u výsledků poskytnutých umělou inteligencí nikdy není stoprocentní jistota správnosti výsledků. K nalezení výsledků se často používá náhodný element a statistika, v případě jednodušších úloh se dá chyba považovat za zanedbatelnou.

1.2 Jak strojové učení funguje

Hlavním principem strojového učení je nalezení modelu pro konkrétní úlohu, díky kterému je pak možno řešit nějaký obecný problém. Pro různé úlohy se hodí různé metody učení. Základní rozdělení učení je na:

1.2.1 Nesupervizované učení

Učení bez učitele se zaměřuje na algoritmy, které se učí z dat, která nejsou označena žádným správným výstupem. Cílem těchto algoritmů je najít v datech skryté struktury a vzorce, které by mohly být užitečné pro další analýzu nebo zpracování.

Jinými slovy, strojový učební model v tomto případě nemá k dispozici žádné "návody", co by měl dělat. Místo toho se musí sám naučit z dat "rozpoznat", co je v nich důležité a jak je organizovat.

Příklad: Algoritmus nesupervizovaného učení by se mohl použít k segmentaci obrázků do různých oblastí. To by fungovalo tak, že by algoritmus nejprve analyzoval vlastnosti jednotlivých pixelů v obrázku (např. barvu, jas, texturu). Na základě těchto vlastností by pak algoritmus rozdělil obrázek do oblastí, které sdílejí podobné vlastnosti.

1.2.2 Supervizované učení

Učení s učitelem se zaměřuje na algoritmy, které se učí z dat, která jsou označena správným výstupem. To znamená, že algoritmus má k dispozici sadu tréninkových dat, která obsahují jak vstupní data (např. obrázky), tak i požadovaný výstup (např. názvy objektů na obrázcích). Algoritmus se pak snaží z těchto dat naučit mapovat vstupní data na požadovaný výstup.

Jinými slovy: Strojový učební model v tomto případě dostává "návod", na to, co má dělat. Na základě tohoto návodů se model učí, jak správně interpretovat vstupní data a generovat požadovaný výstup.

Příklad: Algoritmus supervizovaného učení se může naučit identifikovat objekty na obrázcích tím, že je trénován na sadě obrázků označených správnými názvy objektů. Algoritmus se pak bude snažit naučit, jak na základě vlastností obrázků (např. barvy, tvaru, textury) správně identifikovat zobrazené objekty a zařadit je do jednotlivých tříd.

1.2.3 Posilující učení

Algoritmy se učí interakcí s prostředím a získáváním odměn za dosažení požadovaných cílů. Například algoritmus posilujícího učení se může naučit hrát hru tím, že se pokouší maximalizovat své skóre a zároveň se vyhýbá trestům.[6]

1.3 Optimalizace, minimalizace výstupní chyby

Základem strojového učení je optimalizace, která spočívá v minimalizaci chyby mezi predikovaným a skutečným výstupem modelu. Optimalizace se zaměřuje na změnu čísel ve vektoru parametrů **x**, dokud není nalezena vhodná sada hodnot, která dá co nejbližší výstupní hodnotu.[4; 5; 7]

Klíčovou roli ve strojovém učení hraje proces přiřazování vah parametrům modelu. Váhy představují sílu propojení mezi vstupy a výstupy modelu a odrážejí relativní důležitost jednotlivých vstupů pro daný úkol. Proces přiřazování vah lze chápat jako formulování hypotéz o tom, jak vstupy ovlivňují výstupy. Model porovnává své predikované výstupy s realitou a na základě chyby upravuje váhy (hypotézy) tak, aby dosáhl co nejpřesnějších odhadů. [4; 5; 7]

Každá sada vah reprezentuje konkrétní hypotézu o tom, jak vstupy ovlivňují výstupy, tedy jak jsou hodnoty propojeny s jejich významy. Tyto váhy představují předpoklady o korelacích mezi vstupy neuronové sítě a cílovými hodnotami, které se snaží odhadnout. Při učení s učitelem je důležitou částí tohoto procesu rozhodnutí, které parametry brát v úvahu a které ignorovat. [4; 5; 7]

1.3.1 Regrese

Regrese se zabývá funkcemi, které se snaží predikovat skutečnou výstupní hodnotu. Jinými slovy, odhaduje závislou proměnnou na základě nezávislé proměnné. Mezi nejběžnější typy regrese patří lineární regrese, která se snaží najít lineární funkci, která popisuje vztah mezi proměnnými x a y. Pro známé hodnoty x předpovídá neznámé hodnoty y. [8]

Logistická regrese se používá pro binární klasifikační úkoly, kde je cílem zařadit pozorování do jedné ze dvou kategorií.

Nekonečná regrese umožňuje modelovat složitější nelineární vztahy mezi proměnnými.

Lineární regrese je jedním z nejpoužívanějších typů regrese a je poměrně jednoduchá na pochopení a implementaci. Základní myšlenka spočívá v nalezení přímky, která co nejlépe prochází shlukem bodů reprezentujících data. Tato přímka se nazývá regresní přímka a její rovnice má tvar:

$$y = b0 + b1 * x \quad (1)$$

y je závislá proměnná (výstup)

x je nezávislá proměnná (vstup)

b0 je intercept (bod, kde přímka protíná osu y)

b1 je směrnice přímky (určuje sklon přímky)

Predikce lineárního regresivního modelu se vypočítá jako součet koeficientů z parametrového vektoru **β** a vstupní proměnné ze vstupního vektoru **x**:

$$y_{pred} = \beta 0 + \sum_{0}^{i} \beta_i * x_i \quad (2)$$

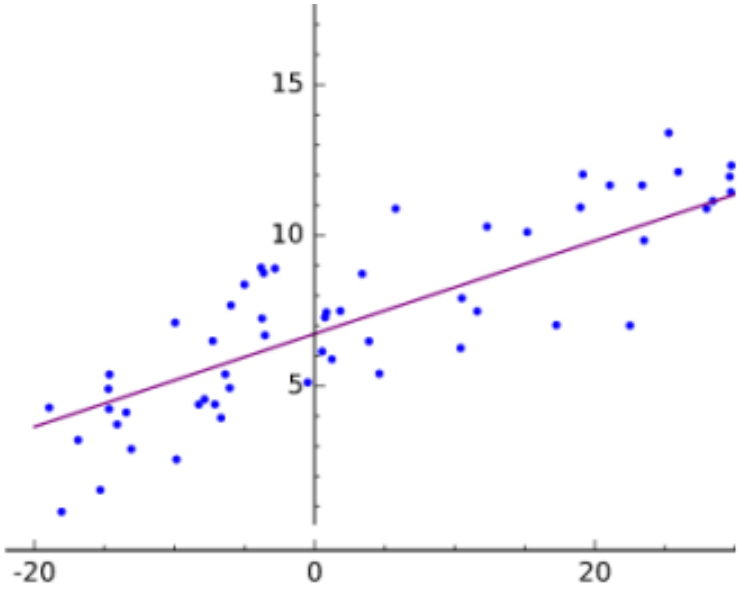
y_pred je predikovaná hodnota závislé proměnné

β_0: Intercept (konstanta) modelu.

β_i: Koeficienty regresního modelu pro i-tou nezávislou proměnnou.

x_i: Hodnoty i-té proměnné pro nový datový bod.

Typickým případem využití lineární regrese je předpověď toho, kolik utratíme za benzín v závislosti na najetých kilometrech. Cena benzínu je funkcí ujeté vzdálenosti. Ujetá vzdálenost je nezávislá proměnná a cena benzínu je závislá proměnná.

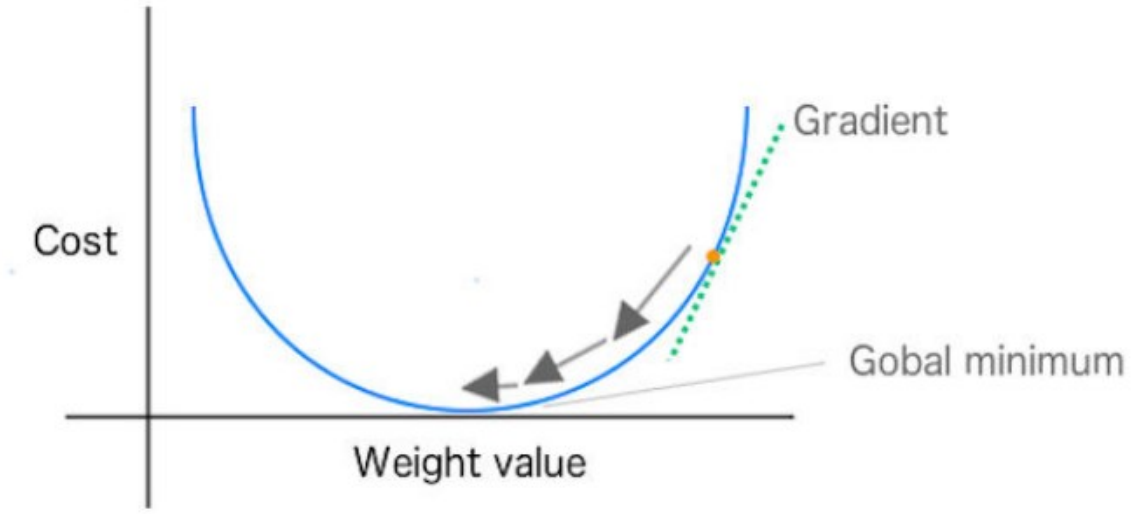


Obrázek 2. Vizuální reprezentace lineární regrese [4]

1.3.2 Gradientní sestup

Při gradientním sestupu je kvalita předpovědí reprezentována konvexní funkcí, hodnoty umístěné vysoko na ose y představují velkou chybu v předpovědi, místa nízko na ose y zase menší predikční chybu. Na parabolu se umisťuje počáteční bod predikce, což předurčí počáteční váhy. Tato počáteční váha může být zvolena na základě znalosti, například pokud víme, že pro detekci vad je třeba hlídat jedno konkrétní místo. Nebo může být celá práce nechána čistě na neuronové síti, v takovém případě může být určeno místo počátečních vah náhodně.[8; 4]

Cílem je upravit váhy neuronové sítě tak, aby byla minimalizována chyba predikce. Toho je dosaženo postupným snižováním hodnoty na ose y. Gradient je získán z derivace ztrátové funkce a udává další krok optimalizačního algoritmu. Obecně je gradientový sestup iterativní optimalizační algoritmus používaný k nalezení minima funkce, v tomto případě chybové funkce. Jedním z problémů, který se u gradientního sestupu musí řešit je, že se postupně dostane do lokálního minima a dále už chybovou funkci neoptimalizuje. Problém je že se jedná pouze o lokální minimum, ale globální minimum může být jiné. [8; 4]



Obrázek 3. Změna vah za účelem dosažení chybového minima pomocí gradientu [4]

Model může generovat různé výstupy na základě nastavení daného modelu. Dva hlavní typy jsou generativní a diskriminační model. Generativní modely se snaží pochopit, jak daná data vznikla, tak, aby mohl vygenerovat příslušný výsledek. Diskriminativní modely neřeší, jak data vznikla, ale jednoduše je zařazuje do kategorií.

1.4 Hluboké učení

Je to podmnožina strojového učení, která využívá umělé neuronové sítě (Artificial neural network) inspirované strukturou lidského mozku. Skládají se z propojených vrstev neuronů, které se učí z dat a dokážou rozpoznávat komplexní vzorce a závislosti.

Hluboké učení je podmnožinou strojového učení, které se zabývá vývojem neuronových sítí s mnoha vrstvami (tzv. hluboké neuronové sítě). Tyto sítě se dokáží učit z velkého množství dat a komplexních vzorců a dosahovat tak pozoruhodných výsledků v široké škále úkolů, jako je rozpoznávání obrazu, zpracování přirozeného jazyka a predikce.

Hluboké neuronové sítě (DNN) se skládají z umělých neuronů, které jsou uspořádány do vrstev. Každý neuron v dané vrstvě je propojen s neurony v následující vrstvě pomocí vah. Během procesu učení se váhy těchto propojení upravují tak, aby síť co nejlépe mapovala vstupní data na požadovaný výstup.

Proces učení v hlubokých neuronových sítích probíhá obvykle iterativně. V každém kroku se síti předloží vstupní data a vypočítá se predikovaný výstup. Následně se porovná

predikovaný výstup se skutečným výstupem a vypočítá se chyba. Na základě chyby se pak upraví váhy propojení mezi neurony tak, aby se v dalším kroku chyba snížila.[4; 2; 1]

Hluboké učení má oproti klasickým metodám strojového učení mnoho výhod:

- Schopnost učit se z velkého množství dat: Hluboké neuronové sítě dokáží zpracovat a analyzovat obrovské objemy dat, čímž získávají cenné informace o komplexních vzorcích a souvislostech.
- Schopnost učit se bez nutnosti ručního značení dat: Mnoho typů hlubokých neuronových sítí se dokáže učit bez nutnosti ručního značení dat, čímž se značně usnadňuje a zrychluje proces učení.
- Schopnost dosahovat vysoké přesnosti: Hluboké neuronové sítě dosahují špičkových výsledků v mnoha úkolech, které dříve byly pro strojové učení obtížné, jako je rozpoznávání obrazu a zpracování přirozeného jazyka.

Hluboké učení má i některé nevýhody:

- Vysoká náročnost na výpočetní zdroje: Trénink hlubokých neuronových sítí může být výpočetně velmi náročný a vyžadovat specializovaný hardware.
- Náchylnost k tzv. overfittingu: Pokud se síť trénuje na příliš malém souboru dat, může dojít k tzv. overfittingu, kdy se síť naučí spíše specifické vlastnosti tréninkových dat než obecné principy.
- Interpretovatelnost modelů: Hluboké neuronové sítě jsou často "černé skříňky", jejichž vnitřní fungování je obtížné interpretovat. To může ztěžovat pochopení toho, jak síť dospěla k danému rozhodnutí.

Hluboké učení má širokou škálu aplikací v různých oblastech, jako jsou:

- Počítačové/Strojové vidění: Rozpoznávání objektů, detekce tváří, sledování objektů, segmentace obrazu, generování obrazu.
- Zpracování přirozeného jazyka: Strojový překlad, rozpoznávání složitých vzorců. [4; 2; 1]

2 CO JE UMĚLÁ NEURONOVÁ SÍŤ

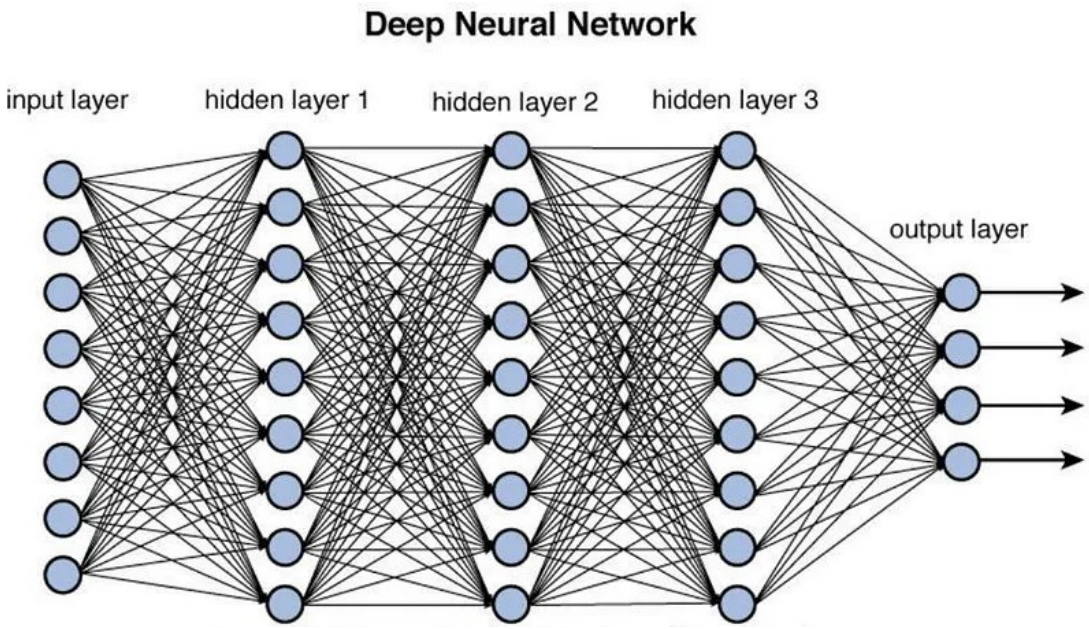
Neuronové sítě jsou podmnožinou strojového učení a také tvoří jádro pokročilých algoritmů hlubokého učení. Jejich název indikuje, že jsou inspirovány funkcí lidského mozku, konkrétně způsobem, jakým biologické neurony komunikují mezi sebou. Nejedná se však o simulaci fungování biologických neuronů, ale spíše o napodobení jejich funkce posílat signály jeden druhému. [5; 4]

Základní jednotkou neuronové sítě je uzel (neuron), který volně vychází z biologického neuronu v mozku savců. Spojení mezi neurony jsou rovněž modelována podle biologického mozku, stejně jako způsob, jakým jsou tato spojení vytvářena. V průběhu času se vyvíjejí pomocí tréninku.

Umělé neuronové sítě se skládají z různých uzlových vrstev. Vstupní vrstva, na tu se přivádí vstupní data, pak jedna nebo více skrytých vrstev, kde se vstupní data různě mění a výstupní vrstva, která tvoří výstup celého algoritmu. Vrstvy jsou podobně jako biologické neurony různě propleteny. Každý z těchto uzlů, nebo umělých neuronů je propojen s minimálně jedním dalším a má přiřazenou váhu a práh. Když bude výstup některého uzlu vyšší než nastavená prahová hodnota, tak se tento uzel aktivuje a odešle data do další vrstvy sítě. V opačném případě se data do další vrstvy neodešlou.[9; 3; 1]

Důležitou vlastností, kterou neuronové sítě simulují, je schopnost učit se. Neuronová síť se učí na základě tréninkových, dat přivedených na vstupní vrstvu. Postupem času také neuronová síť zlepšuje svou přesnost.

Vyladěná neuronová síť představuje mocný nástroj, který umožňuje velmi rychle a přesně klasifikovat a shlukovat velké množství dat. Různé úlohy na rozpoznávání řeči, nebo obrazu trvají neuronové síti minuty oproti hodinám ve srovnání s člověkem. Asi nejznámější umělou neuronovou sítí je vyhledávací algoritmus od Googlu.[4; 1; 2]



Obrázek 4. Propojení jednotlivých uzlových vrstev hluboké neuronové sítě [2]

2.1 Základní architektury neuronových sítí

K různým problémům se hodí různé architektury neuronových sítí. Mohou být využity k jednoduchých klasifikačním úlohám až po komplexní analýzu sekvenčních dat a zpracování přirozeného jazyka.

2.1.1 Feedforward neural networks (FFNN)

Jsou nejjednodušším a základním typem neuronových sítí. Informace v těchto sítích proudí jedním směrem: od vstupní vrstvy přes jednu nebo více skrytých vrstev až k výstupní vrstvě. Architektury: FFNN se skládají z vrstev neuronů, kde každý neuron v jedné vrstvě je spojen se všemi neurony v následující vrstvě. Neurony ve skrytých vrstvách aplikují nelineární aktivační funkce, (popsány v kapitole 2.3.1), tak aby zlepšily schopnost sítě modelovat složité funkce. Používají se pro základní klasifikační a regresní úlohy, jako je rozpoznávání obrazů, hlasová rozpoznávání a predikce číselných hodnot.[10]

2.1.2 Convolutional neural networks (CNN)

Jsou specializovanou architekturou navrženou pro efektivní zpracování a analýzu obrazových dat. CNN využívají konvoluční vrstvy, které automaticky extrahují prostorové rysy z obrazů. Konvoluční vrstvy: aplikují filtry na vstupní obraz a vytvářejí aktivační mapy, které zachycují různé rysy obrazu, jako jsou hrany, textury a složitější struktury. Pooling vrstvy, provádějí subsamplování aktivačních map (např. max-pooling), čímž snižují rozměry dat a zvyšují robustnost vůči posunům a deformacím. Na konci sítě se nacházejí vrstvy podobné těm ve FFNN, které kombinují extrahované rysy a provádějí klasifikaci. Použití: CNN se široce používají v oblastech, jako je rozpoznávání obrazů, detekce objektů, segmentace obrazů a analýza lékařských snímků.[10]

2.1.3 Recurrent neural networks (RNN)

Jsou navrženy pro práci se sekvenčními daty, jako jsou časové řady nebo text. RNN využívají zpětné vazby pro přenos informací mezi časovými kroky, což umožňuje modelovat závislosti v sekvencích. V RNN jsou výstupy neuronů z předchozího časového kroku zpětně připojeny jako vstupy k neuronům v aktuálním časovém kroku. To umožňuje síti uchovávat informace o předchozích stavech.[10]

2.1.4 Transformer Neural Networks

Představují moderní architekturu neuronových sítí, která výrazně zlepšila výkon v úlohách zpracování přirozeného jazyka. Na rozdíl od RNN se Transformers nespoléhají na sekvenční zpracování, ale využívají mechanismus pozornosti (attention mechanism). Mechanismus pozornosti umožňuje síti vážit vstupy různě v závislosti na jejich relevanci pro daný úkol, což usnadňuje modelování dlouhodobých závislostí. Transformery se používají v široké škále aplikací, včetně strojového překladu, shrnutí textu, odpovídání na otázky a generování textu. Modely jako BERT (Bidirectional Encoder Representations from Transformers) a GPT (Generative Pre-trained Transformer) jsou příklady úspěšných implementací této architektury.[10]

2.2 Historie umělých neuronových sítí

První uměle vytvořenou neuronovou síť vytvořili roku 1943 neurofyziologista Warren McCulloch a matematik Walter Pitts. Zabývali se výzkumem biologických neuronů a obecnou funkcí lidského mozku. Aby popsali funkci biologických neuronů, tak vytvořili model jednoduché neuronové sítě pomocí elektrických obvodů.[11]

V roce 1949 zmínil Donald Bebb ve své knize The Organization of Behavior, že neuronové dráhy jsou posilovány pokaždé když jsou použity, což je esenciální koncept pro způsob, kterým se lidé učí.[11]

V roce 1959 Bernard Widrow a Marcian Hoff vytvořili modely nazvané ADALINE a MADELINE. ADALINE byla vyvinuta k rozpoznávání binárních vzorů, takže při čtení datového toku z telefonní linky dokázala předpovídat další bit v pořadí. MADALINE byla první neuronová síť využitá k řešení reálných problémů, konkrétně byla využita jako adaptivní filtr, který eliminoval ozvěny na telefonních linkách. Tento systém se v upravené formě stále využívá.[11]

Navzdory pozdějším úspěchům neuronových sítí, převzala vládu nad výpočetní technikou klasická von Neumannova architektura a konvenční způsoby programování, takže výzkum umělých neuronových sítí zůstal pozadu. Ironií je, že sám von Neumann navrhoval napodobovat funkci neuronů pomocí relé nebo vakuových trubic.[11]

Prvotní úspěchy neuronových sítí vedly k přehánění potenciálu této technologie a s ohledem na tehdejší hardware zůstaly sliby nenaplněny. Také se začaly objevovat filozofické otázky, které vedly k obavám o tzv. myslící stroje a jejich vliv, což zůstává palčivé téma dodnes. Všechny tyto věci vedly ke zpomalení vývoje umělých neuronových sítí.

Představa počítače, který by se v podstatě programoval sám byla velmi lákavá, ovšem velmi těžce implementovatelná. Navíc v tu dobu nabývala Von Neumannova architektura na popularitě a vznikaly na ní komplexní lidsky psané programy, které splňovaly většinu tehdejších požadavků. Naproti tomu na poli neuronových sítí vznikalo jen pár pokroků a většinou šlo o ojedinělé výzkumy.[12]

Výzkum na poli neuronových sítí byl obnoven až roku 1982 Johnem Hopfieldem, který prezentoval přístup obousměrných linek mezi jednotlivými neurony, což byl nový přístup. Tím byla také publikována práce tzv. Hybridní sítě s mnoha vrstvami a každá z nich používala jinou strategii řešení problému.[12]

Hlavní myšlenka, která stála za pozdějšími výzkumy byla, že pokud něco funguje v přírodě, tak to musí fungovat i v počítači. Velký posun nastal až po velkém posunu v hardwaru, ve stejné době, kdy začaly vznikat pokročilé šachové automaty jako Deep Blue.[12]

Vývoj neuronových sítí je obecně pomalý, primárně kvůli hardwarovým omezením, proto učení neuronové sítě trvá až týdny. Technologické firmy se aktuálně snaží vyrábět specifický typ procesoru optimalizovaného pro funkci neuronové sítě. Analogové technologie jsou obecně považovány za překonané, ale právě neurony fungují spíše analogově.

Dnes jsou neuronové sítě výkonnou a důležitou součástí umělé inteligence, ale po mnoho neslavných desetiletí předbíhaly svou dobu. Přestože zpětné šíření (backpropagation) existuje již od 80. let 20. století, teprve v roce 2006 se vědcům podařilo použít jej k úspěšnému trénování hluboké sítě. S nástupem grafických procesorů, které poskytly levnější a rychlejší výpočetní výkon, a s rozvojem internetu, který poskytoval stále větší soubory dat pro trénování, se hluboké sítě vrátily do budoucnosti a začaly pohánět nespočet nových technologií v našem každodenním životě. [13]

Od roku 2012 došlo v oblasti umělé inteligence k významné revoluci díky rozvoji hlubokého učení. Rok 2015 přinesl zásadní průlom, kdy umělá inteligence silně ovlivnila hlavní konferencí o zpracování přirozeného jazyka (NLP). Tento trend se projevil i v dalších disciplínách, jako je počítačové vidění, robotika, zpracování zvuku, umělá inteligence v medicíně a mnoho dalších oblastí.[13]

2.3 Výkonný prvek umělé neuronové sítě

Jak již bylo zmíněno, neuronová síť je biologicky inspirovaný proces. Neurony stejně jako v mozku sbírají informace ze synapsí (v případě umělého neuronu vah). Na základě těchto vstupů rozhoduje neuron, jestli pošle zpracovanou informaci dál (bude aktivní nebo neaktivní) to v případě umělého neuronu reprezentuje aktivační funkce. Umělé neuronové sítě umožňují řešení silně nelineárních problémů.[5]

V umělé neuronové síti je klíčovým prvkem formální neuron. Je to základní stavební kámen sítě, který transformuje prvky vstupního vektoru $\boldsymbol{x}$ na výstupní hodnotu $\boldsymbol{y}$. Tento proces využívá váhy **w**, které určují, jak jednotlivé složky vstupního vektoru ovlivňují výstup, a práh $\boldsymbol{b}$. Výstup neuronu $\boldsymbol{y}$ je poté stanoven prostřednictvím nelineární aktivační funkce:

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{3}$$

2.3.1 Aktivační funkce neuronu

Aktivační funkce neuronu je klíčovým prvkem neuronových sítí, který určuje výstupní signál neuronu na základě jeho vstupů. Tato funkce definuje, jaké vzory vstupů aktivují neuron a jak intenzivně je aktivován. Existuje řada různých typů aktivačních funkcí, které se liší svými vlastnostmi a použitím v různých kontextech.[4]

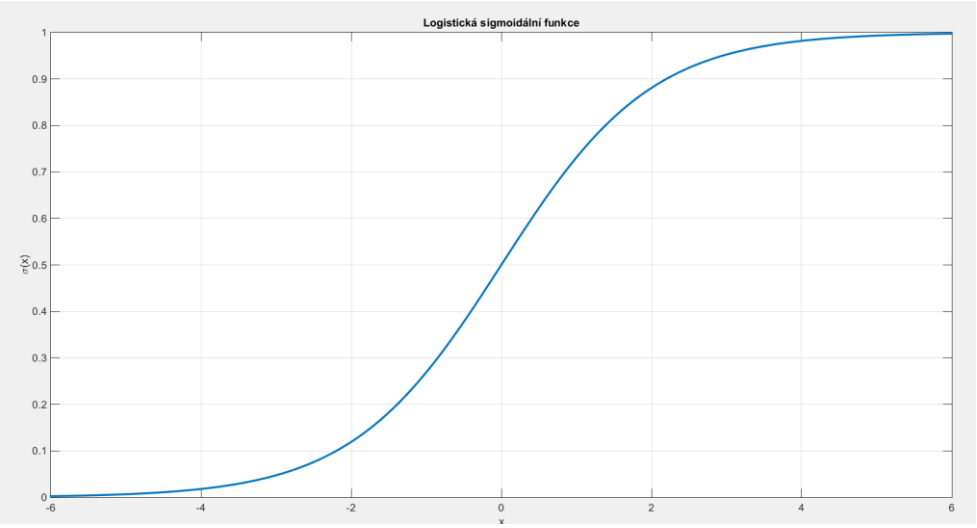
Jedním z nejčastěji používaných typů aktivačních funkcí je sigmoidální funkce, jako například logistická funkce (obrázek 5.), nebo hyperbolický tangens (obrázek 6.). Tyto funkce mají tvar S-křivky a převádějí vstupní hodnoty na rozsah mezi 0 a 1 nebo -1 a 1, což umožňuje modelovat pravděpodobnosti nebo zpracovávat vstupy, které mají omezený rozsah hodnot.

Další často používanou aktivační funkcí je ReLU (Rectified Linear Unit) (obrázek 7.), která vrací nulu pro vstupy menší než nula a samotný vstup pro větší hodnoty. ReLU funkce má jednoduchý tvar a umožňuje efektivní trénování neuronových sítí a zabraňuje problému mizejícího nebo explodujícího gradientu.[4]

V poslední době získává popularitu i další aktivační funkce nazvaná Leaky ReLU(obrázek 8.), která při záporných vstupech vrátí malou nenulovou hodnotu, což pomáhá předcházet problémům s mizením gradientu.

Logistická sigmoidální funkce:

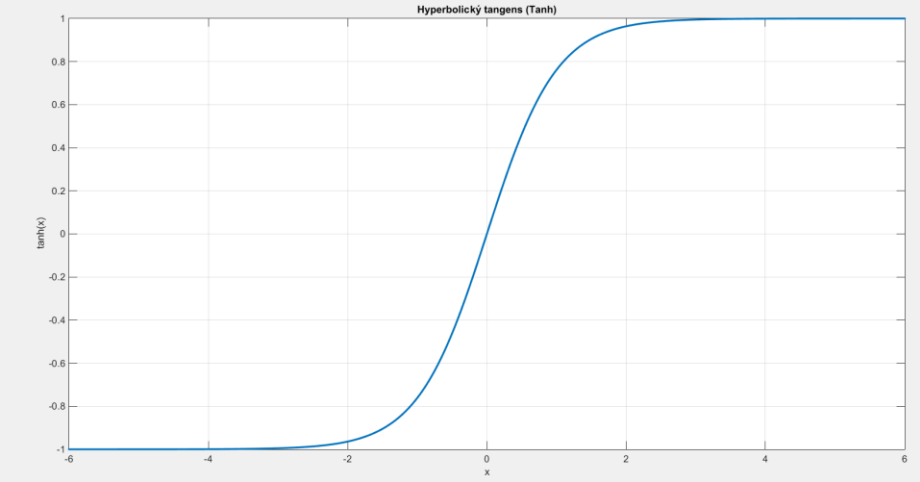
$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$



Obrázek 5.Logistická sigmoidální funkce

Hyperbolický tangens:

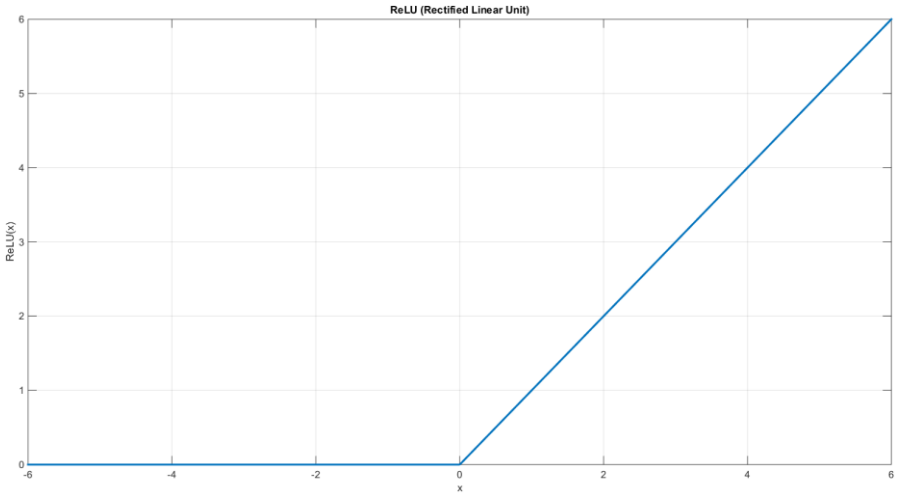
$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{5}$$



Obrázek 6.Hyperbolický tangens

ReLU (Rectified Linear Unit):

$$ReLU(x) = max(0, x) \tag{6}$$

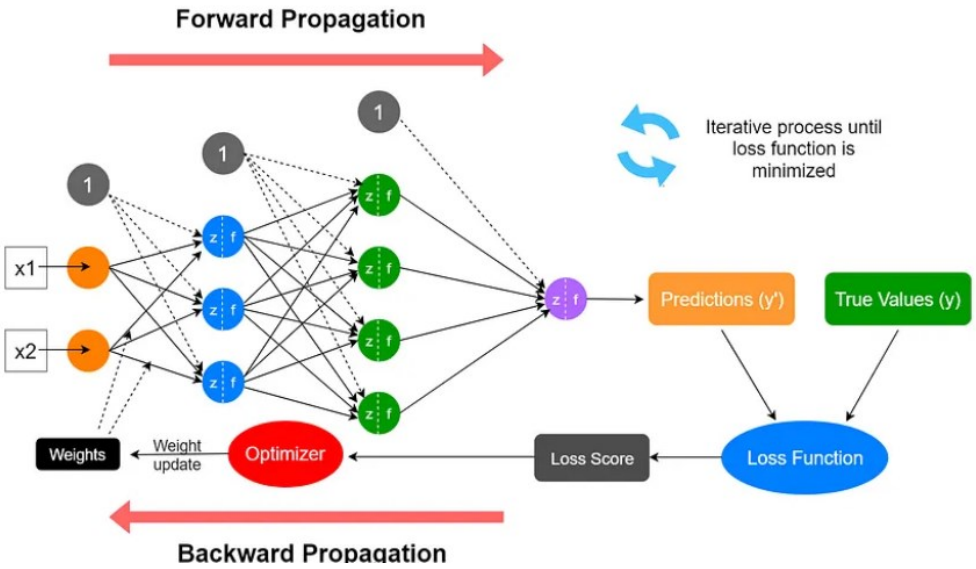


Obrázek 7. ReLU funkce

2.4 Učení neuronové sítě

V procesu učení neuronové sítě dochází ke změnám, kterými se neuronová síť adaptuje na optimální řešení daného problému. Realizuje se to pomocí nastavování synaptických vah a prahů. Nejdříve se nastaví počáteční hodnoty, které jsou buď náhodné nebo jsou nastaveny k rychlejší adaptaci dle reálných poznatků. Základním typem učení je tzv. učení s učitelem. To znamená, že existuje vnější kritérium, které je objektivně správně a v síti se nastavují váhy pomocí zpětné vazby podle toho, jak blízko je výstup kritériu. Po získání rozdílu mezi výstupem sítě a kritériem, se váhy modifikují podle algoritmu, který zajistí snižování chyby mezi výsledkem a kritériem. Poté se síti předloží nový vsup a celý proces se opakuje. Až po provedení velkého množství pokusů se síť naučí stabilně predikovat optimální výstup, který je velmi blízko požadovanému kritériu.[4]

Dalším typem je tzv. učení bez učitele, takže neexistuje objektivní kritérium správnosti výstupu. V tomto případě algoritmus hledá na vstupních datech společné vlastnosti a opakující se vzory a ty pak shlukuje dle podobnosti.



Obrázek 8. Základní princip učení neuronové sítě [7]

Během učení se může stát, že se síť dlouhou dobu zdržuje u lokálních minim chyby, z nichž často není úniku a další prodlužování učení nemá smysl. Může také nastat přerušení sítě, to se projevuje mírným poklesem chyby, následované velkým zvyšováním hodnoty globální chyby. V tomto případě je nutné učení ukončit, případně se vrátit k hodnotám kdy síť dávala nejlepší výsledky. [4]

2.5 Typy neuronových sítí

Uměle vytvořené neuronové sítě se dělí na různé typy podle druhu využití konkrétní sítě.

- Predikce znamená předpovídání výstupní hodnoty, na základě jejího průběhu v minulosti. Jde o to, nalézt průběh známé číselné řady, jejíž hodnoty se mění na základě další nezávislé proměnné sledovaného jevu
- Aproximace spočívá ve funkci přibližně určit hodnotu, kterou není možné nebo výhodné určit přesně.
- Filtrace vyhlazuje průběh vstupního signálu
- Asociace, umělá neuronová síť se naučí rozeznávat nějaký vzor na bezchybných datech a poté dokáže doplňovat data poškozená, asociuje je.
- Optimalizace slouží k určení optimální hodnoty nějaké proměnné, třeba k nalezení nejrychlejšího možného průběhu, či nejkratší možné cesty.

2.6 Princip funkce neuronových sítí

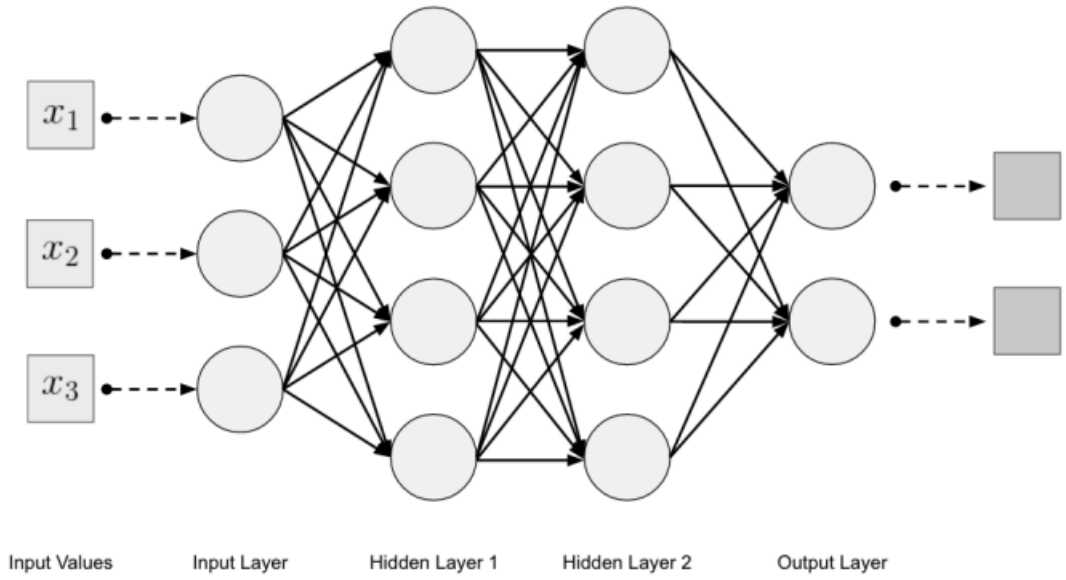
Důležitým pojmem je vektor parametrů, který představuje váhu (sílu) spojení mezi jednotlivými uzly neuronové sítě.

Nejzákladnější jednotkou neuronové sítě je uzel (neuron), a stejně jako biologické neurony tak i ty umělé jsou stimulovány informací. Pokud přijdou neuronu informace od jiného neuronu, tak některé z těchto informací v upravené formě odešle dalším neuronům.

Biologické neurony jsou postupným tréningem naučeny posílat dál jen informace, které jsou užitečné pro dosažení nějakého většího cíle. Stejným způsobem mohou být učeny i neurony v umělé neuronové síti.

Váhy mezi neuronovými spoji vlastně představují dlouhodobou paměť neuronové sítě a změna vah je primární formou učení nové informace.

Nejjednodušším příkladem neuronové sítě je síť s dopředným průchodem více vrstev. Má jednu vstupní vrstvu, několik skrytých vrstev a jednu výstupní vrstvu. Každá vrstva může mít jiný počet neuronů a každá vrstva je propojena s jinou



Obrázek 9. Vícevrstvá neuronová síť [4]

2.7 Adaptace

2.7.1 Hebbovo adaptační pravidlo

Toto adaptační pravidlo je založeno na myšlence, že váhové hodnoty na spojení dvou neuronů, které jsou současně ve stejném stavu (1 nebo 0) budou narůstat a pro dva spojené neurony které současně nemají stejnou hodnotu budou klesat. Takže změna synaptické váhy spoje dvou neuronů je úměrná souhlasné aktivitě součinu jejich stavů. Toto pravidlo platí jen pro neurony s binárním výstupem.

$$R\Delta w_{ij} = a * y_i * y_j \qquad (7)$$

a představuje koeficient učení **y** reprezentuje stavy neuronů. Takže pokud budou oba neurony ve stejném stavu, tak je váhová hodnota zvýšena a naopak.

Hebbovo adaptační pravidlo se ovšem potýká se zásadním problémem. Ten nastane v případě, že vstupní vzory nejsou ortogonální, v takovém případě dochází k rušení a síť se není schopná tuto asociaci naučit.

2.7.2 Delta pravidlo

Je to jedno z nejpoužívanějších adaptačních pravidel. Pro určený vstupní vektor je výstupní vektor porovnán s očekávanou odpovědí. Když se rozdíl rovná nule, tak žádná adaptace neproběhne, v opačném případě jsou hodnoty vah upraveny tak, aby byl tento rozdíl co nejvíce redukován.

Delta pravidlo determinuje změnu váhové hodnoty na spojení vedoucí mezi dvěma neurony $\boldsymbol{u_i}$ *a* $\boldsymbol{u_j}$ dle vztahu:

$$R\Delta \boldsymbol{w_{ij}} = \boldsymbol{a} * \boldsymbol{y_i} * \boldsymbol{e_j} \qquad (8)$$

a je koeficient učení $\boldsymbol{y_i}$ je stav neuronu $\boldsymbol{u_i}$ a $\boldsymbol{e_j}$ je rozdíl mezi očekávanou a reálnou výstupní hodnotou neuronu $\boldsymbol{u_j}$**.** Při adaptaci jsou hledány takové hodnoty, které celkovou chybu minimalizují. Na začátku adaptace jsou hodnoty nastaveny náhodně, což obvykle odpovídá velkým chybám na výstupu. Delta pravidlo posouvá váhový vektor blíže k ideálu a zároveň to činí tím nejefektivnějším způsobem. Graficky se to dá znázornit jako gradientní pohyb váhového vektoru na povrchu paraboloidu směrem dolů k nejnižšímu možnému bodu, který reprezentuje minimální možnou chybu. Tento proces je časově náročný, a to i pro poměrně malé topologie vícevrstvých sítí. [5]

2.8 Klasifikace

Klasifikace se obecně provádí ve dvou krocích:

1.. Výběr klíčových vlastností. Vstupní data jsou nejdříve algoritmicky předzpracována tak, aby byly extrahovány pouze klíčové vlastnosti ze vstupních dat. Primárně se jedná o odstranění vlivů zašumění, posunutí, otočení či poškození. Pro výběr klíčových dat neexistuje žádné jednotné pravidlo. Výběr klíčových vlastností vždy závisí na dané aplikaci a konkrétním typu dat.

2.. Vlastní klasifikace. Extrahované klíčové vlastnosti jsou předány klasifikátoru a ten je poté roztřídí. Klasifikátor tudíž nepracuje přímo se vstupními objekty, ale s jejich obrazy které vznikly během klasifikace.

Kvalitnější proces zpracování vstupních dat umožňuje použití jednoduššího klasifikačního algoritmu. V případě velmi dobře zpracovaných vstupních dat je možné použít i triviální klasifikátor jako je například měření Hammingovy vzdálenosti.

Mějme dvě matice binární matice stejného typu označené A a B. Hammingova vzdálenost těchto matic bude

$$\boldsymbol{d} = \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \sum_{\boldsymbol{j}=1}^{\boldsymbol{n}} \left| \boldsymbol{a_{ij}} - \boldsymbol{b_{ij}} \right| \tag{9}$$

Nejzákladnější formou klasifikace je binární klasifikátor, který má pouze jeden výstup se dvěma hodnotami, typicky 0 a 1.Výstupem může být také desetinné číslo mezi 0,0 a 1,0. V takovém případě musí být určen práh, např. 0,5. Cokoliv nad tuto hodnotu bude bráno jako 1 a cokoliv pod ní bude 0. Příkladem binárního klasifikátoru je například rozeznání vady, výrobek je vadný, nebo je v pořádku. Rozeznání emailu, je to spam nebo běžná pošta atd.

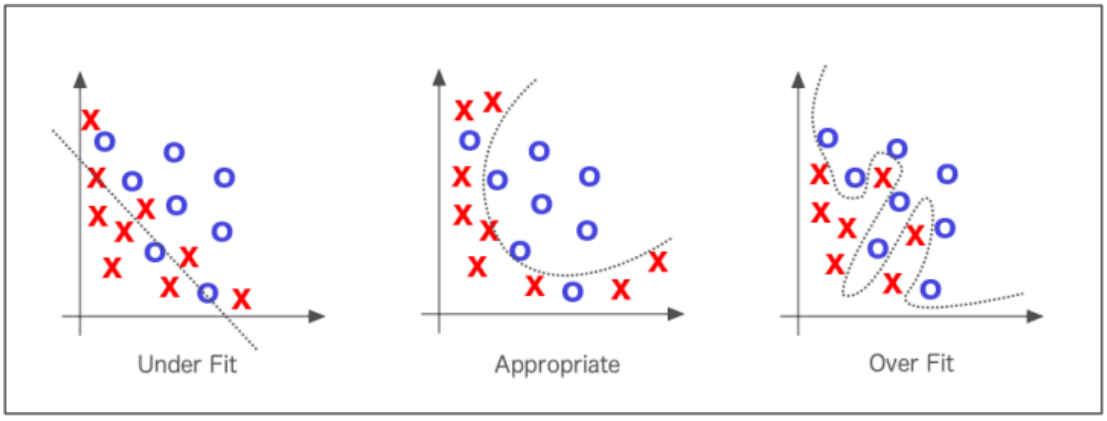
2.8.1 Metoda shlukování

Tato metoda patří mezi metody učení bez učitele, které byly popsány výše. Funguje na principu shlukování jednotlivých položek blíže k sobě, na základě měření vzdálenosti. Na konci procesu jsou položky nejhustěji rozmístěné kolem centroidů, takže jsou zařazeny do konkrétní skupiny.

2.9 Nedostatečné přizpůsobení a přílišné přizpůsobení

Algoritmy se nejprve snaží řešit problém nedostatečného přizpůsobení. Tedy např. vzít přímku, která špatně aproximuje data, a zlepšit její aproximaci dat. Přímka protnutá křivým rozptylovým diagramem by byla dobrým příkladem nedostatečného přizpůsobení.

Opačným problémem, který může nastat je přílišné přizpůsobení. To znamená, že naučený model má minimální chybovost na učených datech, což ale znamená, že tento model není dostatečně obecný k tomu, aby se dal použít na jakákoliv lehce odlišná data.



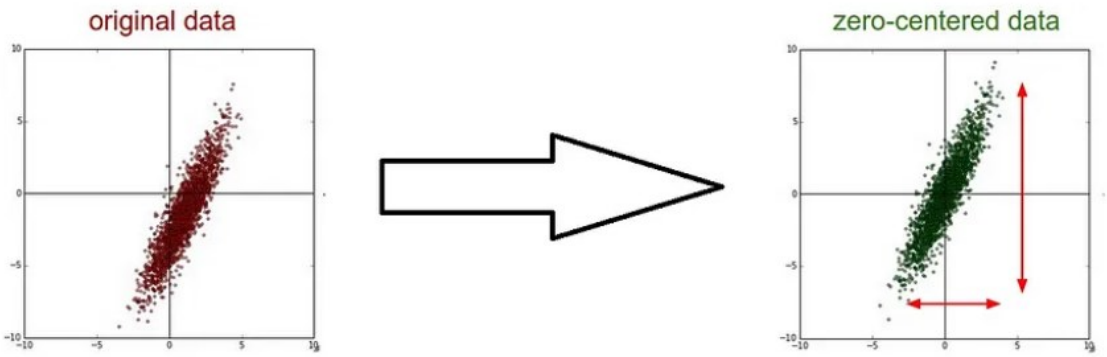
Obrázek 10. Chyby přizpůsobení modelu strojového učení [4]

2.10 Předzpracování dat

Aby se dala neuronová síť použít na aplikace s potřebou vysoké úrovně přesnosti, tak je třeba obrovské množství dat a výpočetního výkonu, aby se danou úlohu naučila. Existují různé metody, které potřebný čas a výkon mohou snížit. Je vhodné použít různé metody úpravy vstupních dat, protože platí, že neuronová síť je jen tak dobrá jako vstupní data, která byla použita k jejímu učení. Pokud nejsou data předzpracována, tak to pravděpodobně negativně ovlivní přesnost a výkon dané neuronové sítě. [2]

2.10.1 Odečítání střední hodnoty-nulové centrování

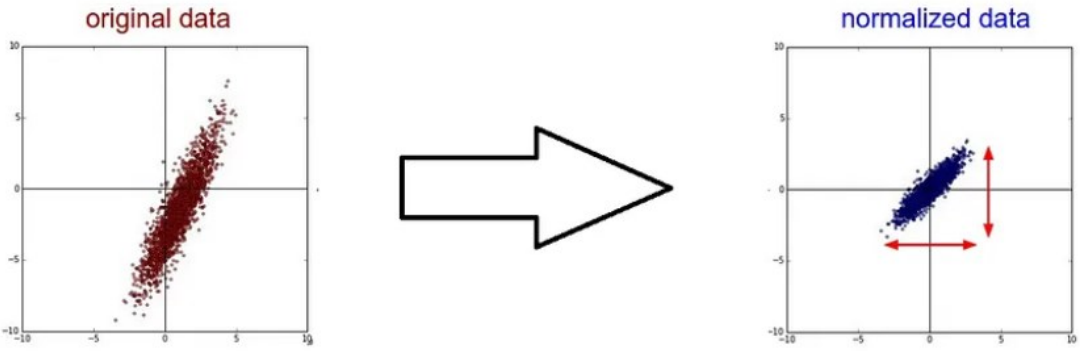
Jedná se o proces odečtení střední hodnoty od každého datového bodu, aby střední hodnota byla nulová a ostatní body byly rozmístěny v jeho okolí. Uvažujme případ, kdy jsou všechny vstupy do neuronu (uzlu) kladné nebo všechny záporné. V takovém případě bude gradient vypočtený při zpětném šíření buď kladný, nebo záporný. [2]



Obrázek 11.Odečtení střední hodnoty[2]

2.10.2 Normalizace dat

Normalizace znamená upravit data tak, aby měla ve všech dimenzích stejnou škálu. Běžný postup spočívá ve vydělení dat směrodatnou odchylkou.



Obrázek 12. Normalizace dat ve dvou dimenzích [2]

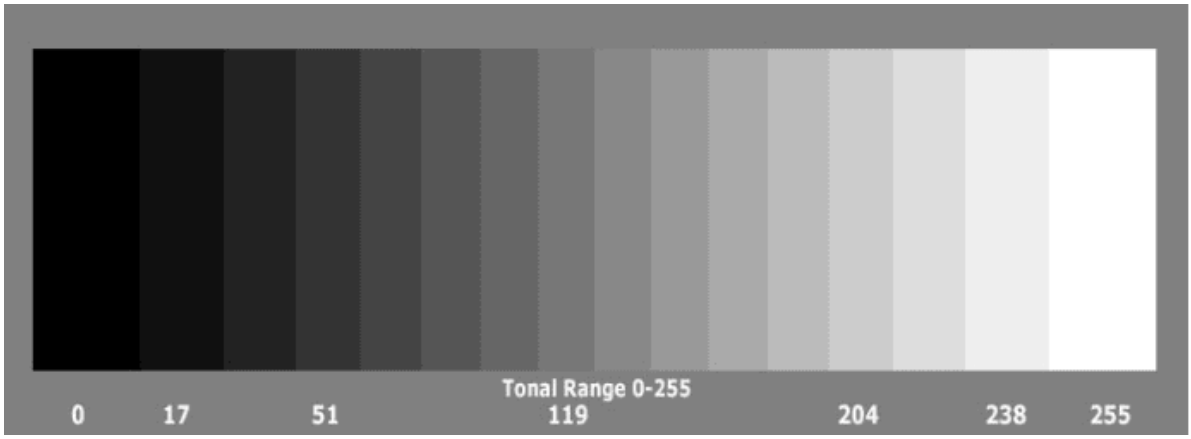
2.10.3 Převod do černobílého spektra

Většina snímků, které jsou zpracovány strojovým viděním či neuronovými sítěmi bývají běžně převáděny do černobílého spektra, a to z několika důvodů. První důvod je snížení celkového objemu dat, neboť jak již bylo řečeno, neuronové sítě potřebují zpracovat velké množství dat pro dosažení dobrých výsledků. Proto je velmi logické snížit celkový datový objem vstupních dat převodem do černobílého spektra, jeden pixel v barevném spektru má velikost tři bajty, jeden bajt pro každý barevný kanál RGB. Převodem do černobílého spektra se tento objem dat sníží na velikost jeden bajt na pixel (0-255, 0 pro černou a 255 pro bílou). V reálu se celková velikost nesníží na 1/3 kvůli faktorům jako je komprese dat nebo různé typy černobílého kódování, ale zmenšení je pořád dost významné.

Dalším důvodem je zjednodušení a standardizace. Převod na černobílou barvu odstraňuje zbytečné informace o barvě, které pro daný úkol strojového vidění nebo neuronové sítě nemusí být relevantní. To umožňuje síti zaměřit se na to, co je pro úkol důležité, a zjednodušit tak model. Navíc barevné obrázky se mohou lišit v závislosti na osvětlení, fotoaparátu a dalších faktorech. Převod na černobílou barvu pomáhá standardizovat vzhled obrázků a usnadňuje tak síti jejich pochopení.

Převodem na černobílé spektrum se také zvýší invariance modelu neuronové sítě. Neuronové sítě by měly být invariantní k malým změnám v datech, aby byly robustní a spolehlivé a převod na černobílou barvu může pomoci zvýšit tuto invarianci snížením vlivu drobných variací v barvě.[14]

Samozřejmě budou existovat i aplikace, kde nebude možné vstupní snímky převést do černobílého spektra, i když to bude znamenat složitější model a delší dobu učení. To je obvykle případ úkolů, kde jsou informace o barvě důležité, jako je klasifikace objektů v barevných obrázcích.[14]



Obrázek 13. Černobílá paleta[14]

II. PRAKTICKÁ ČÁST

3 DOSTUPNÉ NEURONOVÉ SÍTĚ POUŽITÉ PRO DETEKCI VAD

Tato kapitola se primárně zaměřuje na popis neuronových sítí, které budou později testovány a porovnávány dle jejich schopnosti vizuálně detekovat vady a dalších parametrů. V této kapitole se dále nachází stručný přehledem dalších, v průmyslu používaných neuronových sítí a jejich softwaru.

3.1 COGNEX VISIONPRO DEEP LEARNINIG

VisionPro Deep Learning představuje softwarový nástroj založený na umělé inteligenci, určený pro analýzu obrazu v náročných průmyslových aplikacích. Jeho primárním cílem je automatizace úkolů, které se obtížně programují pomocí tradičních algoritmů strojového vidění, a to v důsledku jejich komplexnosti a časové náročnosti. VisionPro nabízí konzistentní výsledky a rychlé zpracování dat, čímž překonává limity manuální kontroly obrazových dat. Díky schopnosti pracovat s přirozenou variabilitou dat a rozlišovat mezi akceptovatelnými a neakceptovatelnými anomáliemi se VisionPro Deep Learning stává klíčovým nástrojem pro vývoj aplikací v oblastech s vysokou mírou variability.[15]

3.1.1 Cognex Toolsets

Prostředí společnosti Cognex poskytuje širokou škálu nástrojů fungujících na základě principů hlubokého učení. Všechny dostupné nástroje jsou následující:

- Nástroj Locate je schopen identifikovat součásti navzdory variacím v perspektivě, orientaci, jasu, odlesku a barvě, a to díky učení se ze vzorků poskytnutých uživatelem. Tento nástroj je schopen rozpoznat rysy i v rušných pozadích, ve špatně osvětlených prostředích, na součástech s nízkým kontrastem, a dokonce i na součástech, které se ohýbají nebo mění tvar. Locate představuje spolehlivé řešení pro automatizaci ověřování montáže. Lze ho naučit lokalizovat různé komponenty, i když se zdají odlišné nebo mají různé velikosti.
- Nástroj Analyze je schopen identifikovat jemné defekty na široké škále pozadí a povrchových textur součástí. Může být trénován tak, aby toleroval běžné variace osvětlení a pozicování součástí, přičemž detekuje vady, kontaminaci a další nedostatky. V situacích, kde není praktické sbírat obrázky vad nebo kde jsou vady velmi nekonzistentní, lze použít nekontrolovaný režim. Ten je trénován pouze z dobrých obrázků a identifikuje případy odchylující se od normálního vzhledu

součásti. Nástroj Analyze lze také použít k segmentaci konkrétních variabilních oblastí na obraze, což usnadňuje další kontroly.

- Nástroj Classify je robustní klasifikátor, který lze použít k řešení náročných úkolů klasifikace a rozlišení mezi různými objekty a defekty. Identifikuje a třídí produkty do tříd na základě jejich společných charakteristik, jako jsou barva, textura, materiály, balení a typ defektu. Tento nástroj toleruje přirozenou odchylku uvnitř stejné třídy a spolehlivě rozlišuje přijatelnou variaci od různých tříd.
- Nástroj Read rozpoznává špatně deformované, zkreslené a špatně vytištěné kódy pomocí optického rozpoznávání znaků (OCR). Využívaje hluboké učení založeného na předtrénované knihovně písma, může být tento nástroj natrénován na čtení kódů specifických pro danou aplikaci, které tradiční nástroje OCR nedokážou dekódovat. Kromě toho funkce vizuálního ladění identifikuje chybně přečtené kódy, které lze snadno opravit.

3.2 Aurora Vision Studio

Software Aurora Vision Studio Deep Learning využívá sadu pokročilých nástrojů založených na hlubokém učení, které pomáhají zlepšit kvalitu a provozní efektivitu stávajících řešení strojového vidění. Díky případům použití napříč různými průmyslovými aplikacemi tento software umožňuje řešení složitých problémů strojového vidění, které dříve nebylo možné řešit pomocí tradičních algoritmů a přístupů.[16]

3.2.1 Aurora Deep Learning Toolbox

Stejně jako prostředí Cognex, tak i prostředí Aurora Deep Learning nabízí různé nástroje určené ke konkrétnímu typu aplikace. Tyto nástroje se často svým účelem i názvem liší od nástrojů v softwaru Cognex, proto zde budou stručně popsány:

- Anomaly Detection – pro detekci neočekávaných změn objektů, trénuje se na vzorových snímcích označených jednoduše jako dobré nebo špatné.
- Feature Detection – pro detekci oblastí s vadami (např. škrábance na povrchu) nebo znaky (např. cévy na lékařských snímcích). Trénováno na vzorových snímcích doplněných přesně označenými pravdivými oblastmi.

- Object Classification – pro identifikaci třídy nejvýznamnějšího objektu na vstupním snímku. Trénováno na vzorových snímcích doplněných očekávanými značkami tříd.
- Instance Segmentation – pro současnou lokalizaci, segmentaci a klasifikaci více objektů na scéně. Trénováno na vzorových snímcích doplněných přesně označenými oblastmi každého jednotlivého objektu.
- Point Location – pro lokalizaci a klasifikaci více klíčových bodů. Trénováno na vzorových obrazech doplněných o označené body očekávaných tříd.
- Read Characters – pro lokalizaci a klasifikaci více znaků. Tento nástroj používá předtrénovaný model a nelze jej trénovat, proto není v tomto článku popsán.
- Object Location – pro umístění a klasifikaci více objektů. Trénováno pomocí vzorových obrázků doplněných o vyznačené ohraničující obdélníky očekávaných tříd.

3.3 Další komerčně dostupné nástroje pro detekci vad

3.3.1 Omron Deep Learning

Platforma pro detekci vad s funkcemi pro analýzu obrazu a klasifikaci. Nabízí širokou škálu funkcí pro detekci vad, včetně Segmentace objektů, což je automatické rozdělení obrazu na jednotlivé objekty pro analýzu. Klasifikace čili identifikace a kategorizace objektů v obrazu. Detekce vad, identifikace abnormalit a vad v objektech. Lokalizace vad, určení přesné polohy vad v obraze. Platforma je flexibilní a umožňuje uživatelům trénovat vlastní modely na základě jejich specifických dat a požadavků. Nabízí snadno použitelné rozhraní a podporuje integraci s existujícími systémy. [17]

3.3.2 TensorFlow

Vyvinuto Googlem, TensorFlow je open-source knihovna pro strojové učení a neuronové sítě. Je široce používána v průmyslu pro různé aplikace, včetně detekce vad. TensorFlow nabízí bohatou sadu nástrojů pro trénování a implementaci modelů hlubokého učení.[18]

3.3.3 Keras

Keras je high-level API běžící na vrcholu TensorFlow (ale i dalších backendů). Je známá pro svou snadnost použití a rychlý vývoj prototypů. V průmyslu je často využívána pro rychlé sestavování a testování modelů neuronových sítí. [19]

3.3.4 NVIDIA DeepStream

NVIDIA DeepStream SDK je určený pro analýzu videa a zpracování streamů, využívající GPU akceleraci. Tento software je často používán v průmyslových aplikacích pro detekci vad v reálném čase. [20]

4 PRÁCE SE SOFTWAREM NEURONOVÉ SÍTĚ

Práce v obou popsaných softwarových prostředích se liší, ale principiálně jsou stejné. Do nástroje se nejdříve vloží snímky, na které se bude neuronová síť aplikovat. Ty se poté manuálně označí na dobré (OK) a špatné (NOK), (V Aurora Vision je též nutné vybrat validační snímky). Též jde využít zabudované funkce připnutí stringu k názvu obrázku, což významně ulehčuje orientaci mezi dobrými a špatnými snímky.

Poté se zvolí mód, ve kterém bude neuronová síť pracovat např. Cognex umožňuje Locate, Analyze, Classify nebo Read. Aurora Vison nabízí Anomaly Detection, Feature Detection, Object Classification, Instance Segmentation, Point Location, Read Characters, Object Location. Tyto mody byly důkladněji popsány v kapitole 36

Následně se nastaví parametry neuronové sítě, ty budou popsány dále v kapitole 4.1.1.

Dále se provede nastavení Feature Size, což je nastavení přibližné velikosti prvku, který bude neuronová síť hledat. Velikost se udává v pixelech. Pro nástroj Analyze (v režimu bez učitele nebo režim s učitelem) by se velikost prvku měla blížit velikosti typické vady. U podlouhlých defektů, jako jsou pruhy, šmouhy, škrábance a trhliny, by se velikost prvku měla blížit šířce defektu. Pro režim Locate se doporučuje, aby velikost prvku odpovídala velikosti identifikovaného objektu. Pro nástroj Read by měla být velikost prvku přibližně stejná jako velikost ohraničujícího rámečku typického znaku "A" nebo "E". U nástroje Classify je velikost prvku subjektivní. Velikost prvku by měla odpovídat velikosti prvků obrazu, které lidský pozorovatel použije ke klasifikaci snímků.

Poté se zvolí konkrétní oblast, ve které bude probíhat detekce ROI (Region of interest).

Výstup se liší podle konkrétního nástroje. Např. výstupem nástroje Analyze je bodové ohodnocení každého snímku. Rozdíl mezi špatnými a dobrými snímky by měl být ideálně co největší, to by znamenalo, že neuronová síť bezpečně rozezná vadný kus od dobrého. V realitě budou nějaké snímky blízko u středu, což znamená, že neuronová síť si není jistá zařazením daného snímku. V některých nástrojích se v nastavení pak dá hýbat s prahem rozeznání (threshold) neuronové sítě, to nastavuje, od jaké hranice má neuronová síť snímek považovat za OK nebo NOK.

4.1.1 Nastavení parametrů učení

V části parametry nástroje je možné doladit výkon nástroje před tréninkem a způsob, jakým bude nástroj zpracovávat snímky za běhu. Oba testované nástroje mají některé parametry stejné, ale s jiným rozsahem a nastavitelnými hodnotami a některé parametry se liší. Zde budou popsány nastavitelné parametry nástroje firmy Cognex.

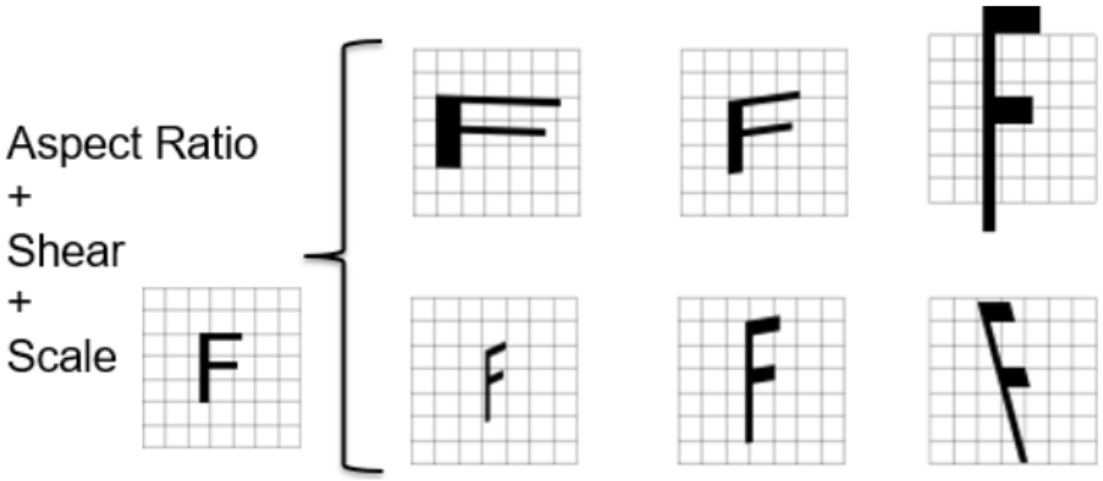
- Mode – zde je možné nastavit dva základní režimy učení, a to Supervised a Unsupervised (S učitelem, bez učitele), rozdíly mezi těmito dvěma režimy byly popsány teoretické části bakalářské práce v kapitole 1.2.
- Network Model – Tento parametr umožňuje zmenšit velikost natrénované sítě nástrojů, čímž se zkrátí doba potřebná ke zpracování. Výchozí hodnota (Large) zachovává standardní chování a výkon nástroje. Další dvě hodnoty, Medium a Small, zajišťují postupně rychlejší provádění za běhu. Výběr jiné hodnoty než Large může vést k odlišným výsledkům nástroje, ale tyto rozdíly jsou obvykle malé.
- Feature size – Udává předpokládanou velikost hledaného prvku. Tento parametr je hlouběji popsán v kapitole 0.
- Color – Tento parametr udává počet barevných kanálů, jeden typicky pro černobílé snímky a tři typicky pro barevné RGB snímky.
- Epoch count – Tento parametr určuje, kolikrát se provede zpřesnění neuronové sítě. Během trénování vstupní vzorky opakovaně procházejí sítí a jejich výstupy se porovnávají s označením zadaným uživatelem. Na základě těchto porovnání se pak upravují váhy sítě s cílem minimalizovat chybu. Vzhledem k velkému počtu uzlů a vah v síti lze tento proces opakovat prakticky donekonečna, přičemž každá iterace vede k mírnému zlepšení chyby. Zvýšením počtu epoch se zvýší počet provedených tréninkových iterací. To povede ke snížení chyby sítě na trénovacích snímcích, ale za cenu delší doby tréninku. Je důležité si uvědomit, že cílem trénování je naučit model obecně rozpoznávat detekované vady, a ne pouze ty specifické, které byly použity k trénování. Tento problém se nazývá přetrénování a je podrobněji popsán v kapitole 2.9. S rostoucím počtem epoch bude mít síť tendenci k přetrénování, kdy se chyba na netrénovaných snímcích zvyšuje, zatímco chyba na trénovaných snímcích se snižuje. Z tohoto důvodu je nutné s tímto parametrem zacházet opatrně a vždy

sledovat jeho vliv na celkovou sadu snímků. Optimální počet epoch se liší v závislosti na konkrétní úloze a použité datové sadě. Obecně se doporučuje začít s nízkým počtem epoch a postupně ho zvyšovat, dokud se nedosáhne požadované přesnosti.

Je důležité sledovat křivky chyby na trénovacích a validačních sadách snímků. Pokud se validační chyba začne zvyšovat, i když se trénovací chyba snižuje, je to známka přetrénování a počet epoch by měl být snížen.

- Low Precision – tento režim převede učený model na model s nízkou učenou přesností. Výhodou tohoto režimu je, že je o 20% - 50% rychlejší.

Petrturbation parametrs: Tyto parametry uměle generují z naučených snímků další snímky s pozměněnými charakteristikami, na kterých se síť dále trénuje, což zlepšuje výsledky u aplikací s velkým rozptylem. Tyto parametry jsou společné pro všechny nástroje. Parametry Perturbation lze také kombinovat. To umožňuje generovat složitější obrazy pomocí pozměněných charakteristik, samostatně i v kombinaci.



Obrázek 14. Petrturbation parametrs [21]

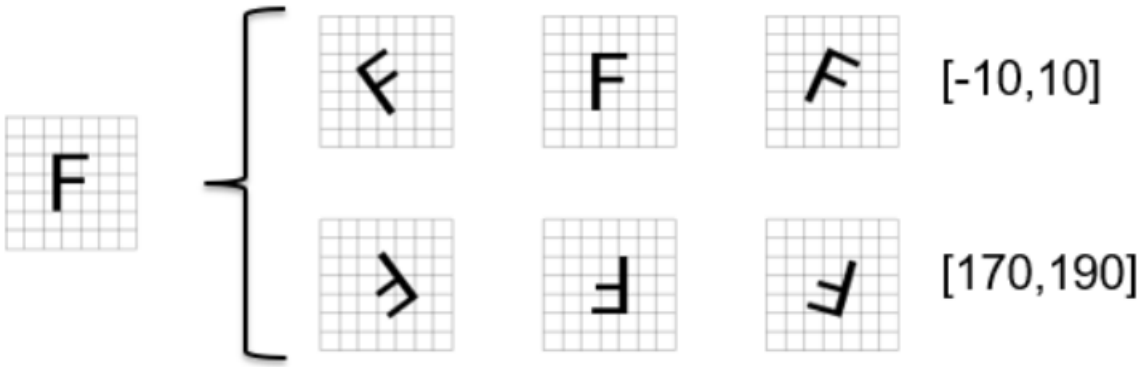
- Rotation – Uživatel má možnost definovat možné orientace dílu. Tyto orientace jsou specifikovány pomocí rovnoměrného rozdělení v zadaném rozsahu úhlů. Z tohoto rozdělení je následně vybrán náhodný úhel natočení dílu pro simulaci rušení. Pokud je rozsah nastaven na [0°, 0°], žádné rušivé natočení se neaplikuje.

Příklad:

Nastavení [-10°, 10°]: Díl může být orientován v rozsahu od -10° do 10° s náhodným úhlem natočení v tomto intervalu.

Nastavení [170°, 190°]: Díl může být orientován v rozsahu od 170° do 190° s náhodným úhlem natočení v tomto intervalu.

Tento typ nastavení je užitečný pro díly, které se obvykle vyskytují v pravidelné orientaci s malou odchylkou úhlu, ale nikdy ne mezi nimi. Například víčko lahve se může vyskytovat vzhůru nohama s odchylkou 5°, ale nikdy ne v úhlu 90°. Rozdělení úhlů se předpokládá jako rovnoměrné. To znamená, že každý úhel v zadaném rozsahu má stejnou pravděpodobnost výskytu.

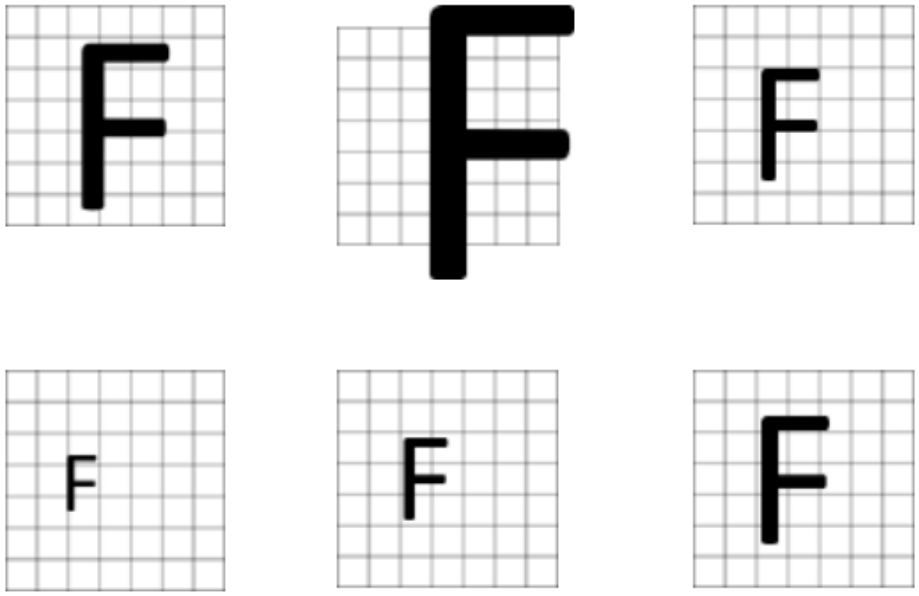


Obrázek 15. Parametr rotation [21]

- Scale – Uživatel má možnost definovat normální rozdělení měřítkového faktoru s průměrem 1. Z tohoto rozdělení je následně vybrán náhodný měřítkový faktor pro simulaci rušení. Pokud je průměr nastaven na 0 %, žádné rušení měřítkem se neaplikuje. Rozdělení měřítkového faktoru se předpokládá jako normální. To znamená, že většina měřítkových faktorů bude blízko průměru (1) a s rostoucí vzdáleností od průměru bude jejich pravděpodobnost klesat. Tento typ nastavení je užitečný pro simulaci variability velikosti dílu. Například jablko se může vyskytovat v různých velikostech, i když jeho tvar je relativně konzistentní.

Příklad:

Nastavení 100 %: měřítkový faktor se bude pohybovat v rozmezí 0 % až 200 % s průměrem 100 %. To znamená, že obrázky mohou být zmenšeny na polovinu (0 %) nebo zvětšeny na dvojnásobek (200 %) původní velikosti.



Obrázek 16. Parametr scale [21]

- Aspect-Ratio – Uživatel má možnost definovat normální rozdělení poměru stran s průměrem 1. Z tohoto rozdělení je následně vybrán náhodný faktor poměru stran pro simulaci rušení. Pokud je průměr nastaven na 0 %, žádné rušení poměru stran se neaplikuje.

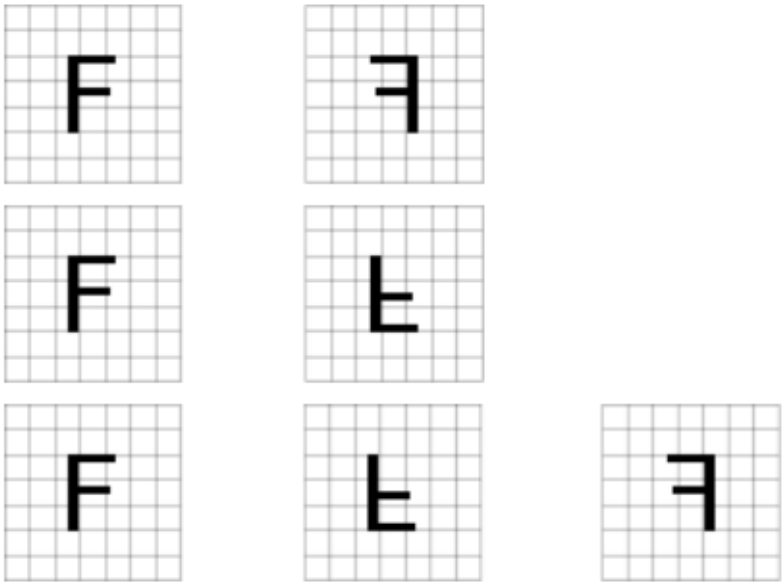
  Příklad:

  Nastavení 100 %: poměr stran se bude pohybovat v rozmezí 50 % až 200 % s průměrem 100 %. To znamená, že obrázky mohou být nataženy na polovinu (50 %) nebo stlačeny na dvojnásobek (200 %) původního poměru stran.

  Tento typ nastavení je užitečný pro simulaci variability tvaru dílu. Například plechovka s nápojem může mít mírně odlišný poměr stran v závislosti na výrobním procesu. Rozdělení poměru stran se předpokládá jako normální. To znamená, že

většina poměrů stran bude blízko průměru (1) a s rostoucí vzdáleností od průměru bude jejich pravděpodobnost klesat.

- Flip – Určuje, zda se má obraz při vzorkování náhodně převracet v horizontálním, vertikálním nebo obou směrech. To je užitečné, pokud objekt nebo povrch vykazuje odpovídající symetrie. Toto nastavení může pomoci výrazně zvýšit množství trénovacích dat.



Obrázek 17. Parametr flip [21]

- Luminance – Určuje normální rozdělení kolem hodnoty 1, ze kterého se čerpá rušivý faktor jasu (pokud je nastaveno na 0 %, nebude použit).
- Invert Contrast – Určuje, zda se má náhodně invertovat kontrast, aby se simulovalo obrácení kontrastu v trénovacích vzorcích.

- Sample Density – Parametr Hustota vzorkování určuje, jak hustě je obraz vzorkován. Čím menší číslo, tím méně vzorků bude z obrazu odebráno. Tato hodnota má významný vliv na dobu učení. Čas doby učení se bude měnit kvadraticky v závislosti na hodnotě vzorkování.

  Doba učení v závislosti na hustotě vzorkování = t

  Hodnota nastavení hustoty vzorkování = x

$$t = x^2$$

5 TRÉNINKOVÁ A TESTOVACÍ DATA

Správná volba a příprava trénovacích dat má zásadní vliv na úspěšnost trénování a výkonnost neuronových sítí. Je důležité, aby tréninková data správně reprezentovala celou sadu dat, to je klíčové ke správné generalizaci neuronové sítě.

5.1 Předzpracování snímků

Všechny snímky byly normovány, takže hledaný objekt se vždy nachází v jasně vymezeném prostoru. Všechny snímky byly normalizovány na stejnou velikost a byly převedeny do černobílého spektra což je běžná úprava pro většinu aplikací strojového vidění. Smysl této úpravy je popsán v kapitole 2.10.3.

5.2 Detekce vad na povrchu munice

První sada dat byla sestavena ze snímků povrchu střeliva ráže 5.56. Snímková sada byla poskytnuta firmou AMV z průmyslové aplikace. Všechny snímky byly předzpracovány pro lepší výsledky. Snímky byly získány 360° kamerou, která pomocí rotace nasnímala celý povrch náboje a uložila ho v podobě jednoho snímku.

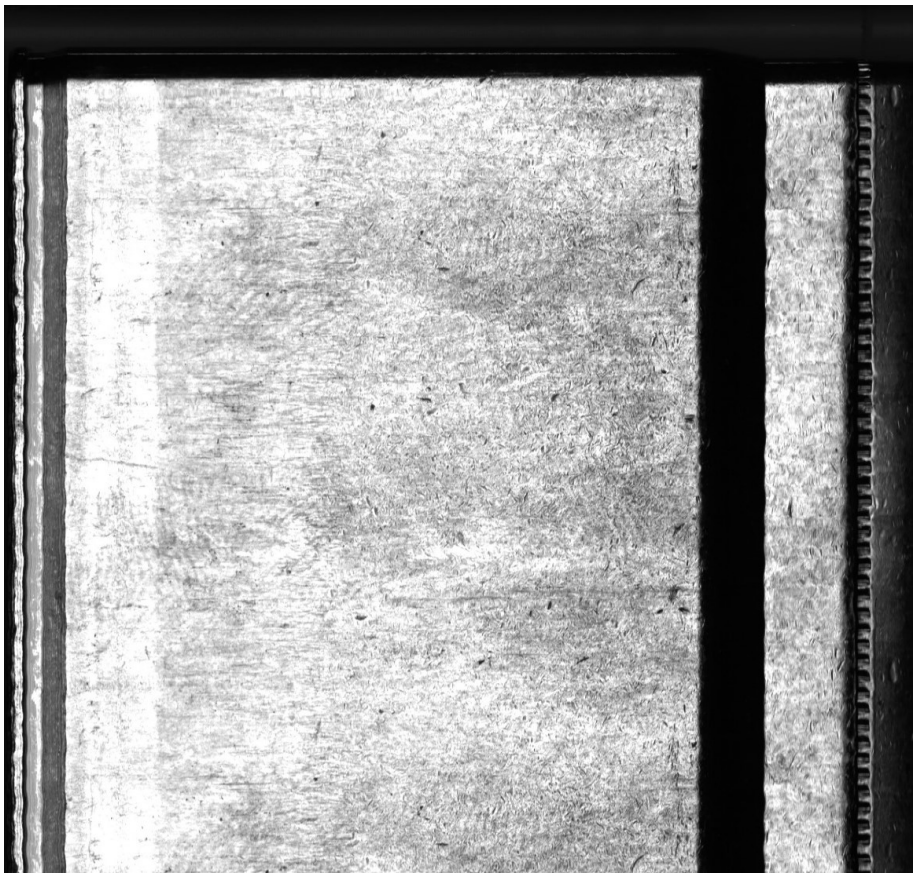
Problematika, která musela být u této sady řešena, spočívala v tom, že hledaná vada nebyla nijak striktně definována. V podstatě na každém snímku se v určité míře vyskytovaly nějaké povrchové vady. Na některých snímcích byly vady velmi výrazné, zatímco na jiných zanedbatelné. U některých se nedalo jednoznačně určit, zda je tento povrch v pořádku či nikoliv. V reálu by záleželo na konkrétních potřebách aplikace, od jaké míry by byl povrch považovaný za NOK a kdy za OK. Proto byly snímky roztříděny na OK a NOK, čistě podle toho, jakou míru povrchových vad jsem považoval za OK a jakou za NOK, tudíž se jednalo spíše o nějaké spektrum vad, na kterém bylo třeba rozhodnout, kdy je snímek spíše OK a kdy spíše NOK, to by se poté dalo řešit pomocí konkrétního nastavení prahu. Do snímkové sady byly schválně zařazeny některé snímky, u kterých nebylo zcela jasné, do které kategorie by měly být zařazeny, aby mohla být zhodnocena reakce modelu neuronové sítě na tento typ snímků.

Tato sada snímků byla ideální pro vyzkoušení vlastností neuronové sítě, protože spektrum vad bylo velmi široké a řešení pomocí technik bez využití neuronové sítě by byla extrémně obtížná až nemožná.

5.2.1 Sada pro nástroj ANALYZE, povrchové nerovnosti

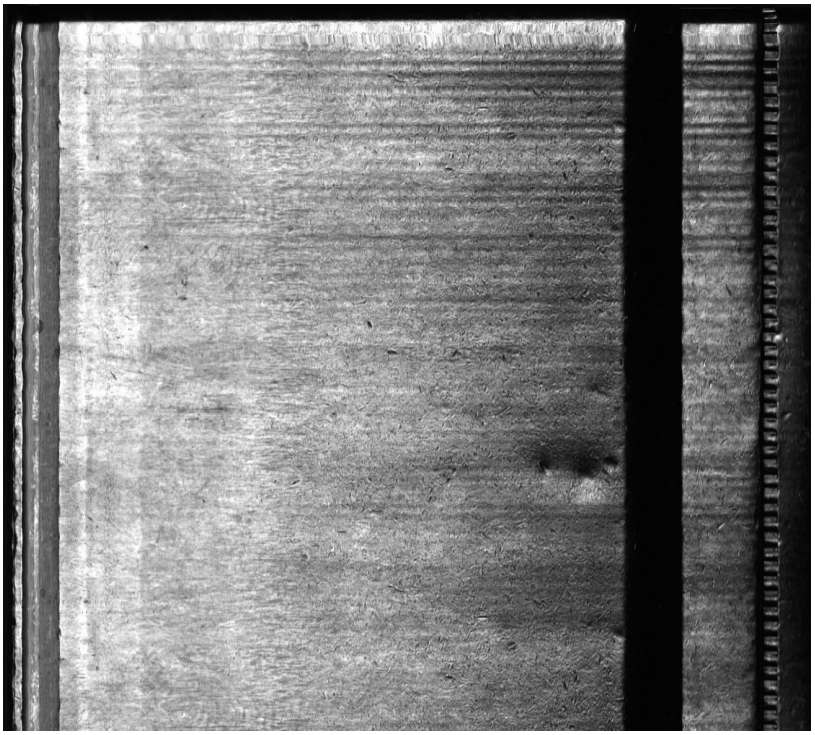
Na této sadě bylo cílem vycvičit model tak, aby se naučil rozeznávat nevýznamné narušení povrchu v podobě černých teček od významného poškození povrchu v podobě povrchové nerovnosti. Do sady bylo též zařazeno pár snímků, u kterých není zcela jasné, zda se jedná o významnou či nevýznamnou vadu, aby mohla být zkoumána reakce modelu na tento typ problému.

Na typickém OK snímku se nacházely nějaké malé povrchové vady v podobě černých teček, ale nebyly nijak výrazné, či hluboké.



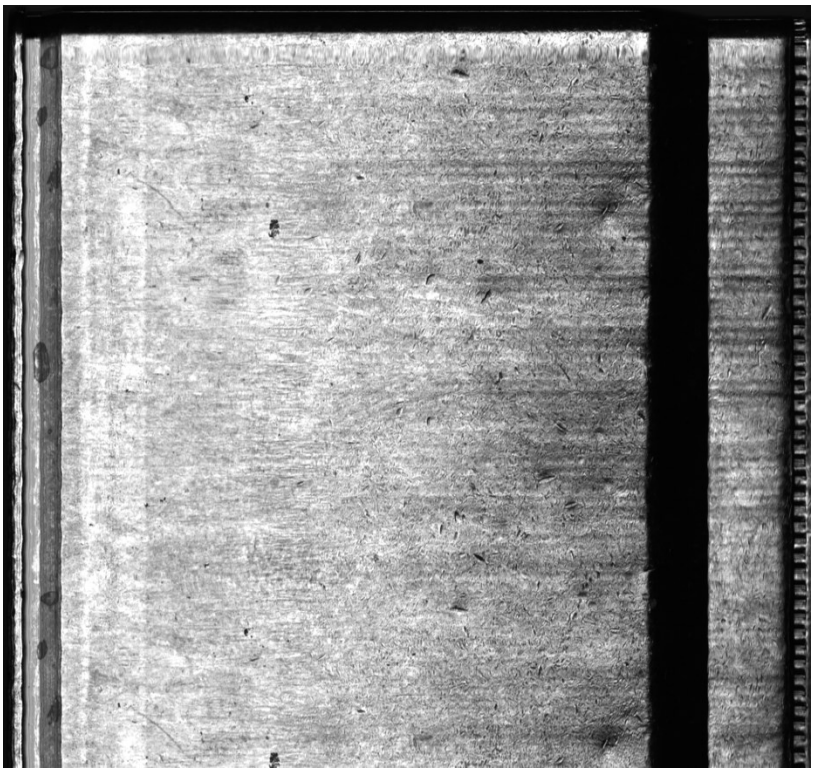
Obrázek 18. Příklad OK povrchu

Na NOK povrchu se nacházely různé větší povrchové nerovnosti



Obrázek 19. Příklad NOK povrchu

Bylo též zařazeno pár neurčitých snímků, kde není zcela jasné, zda se již jedná o vadu či nikoliv. Tyto snímky byly zařazeny čistě dle mé osobní preference podle toho kam by spíše náležely.



Obrázek 20. Příklad neurčitého povrchu

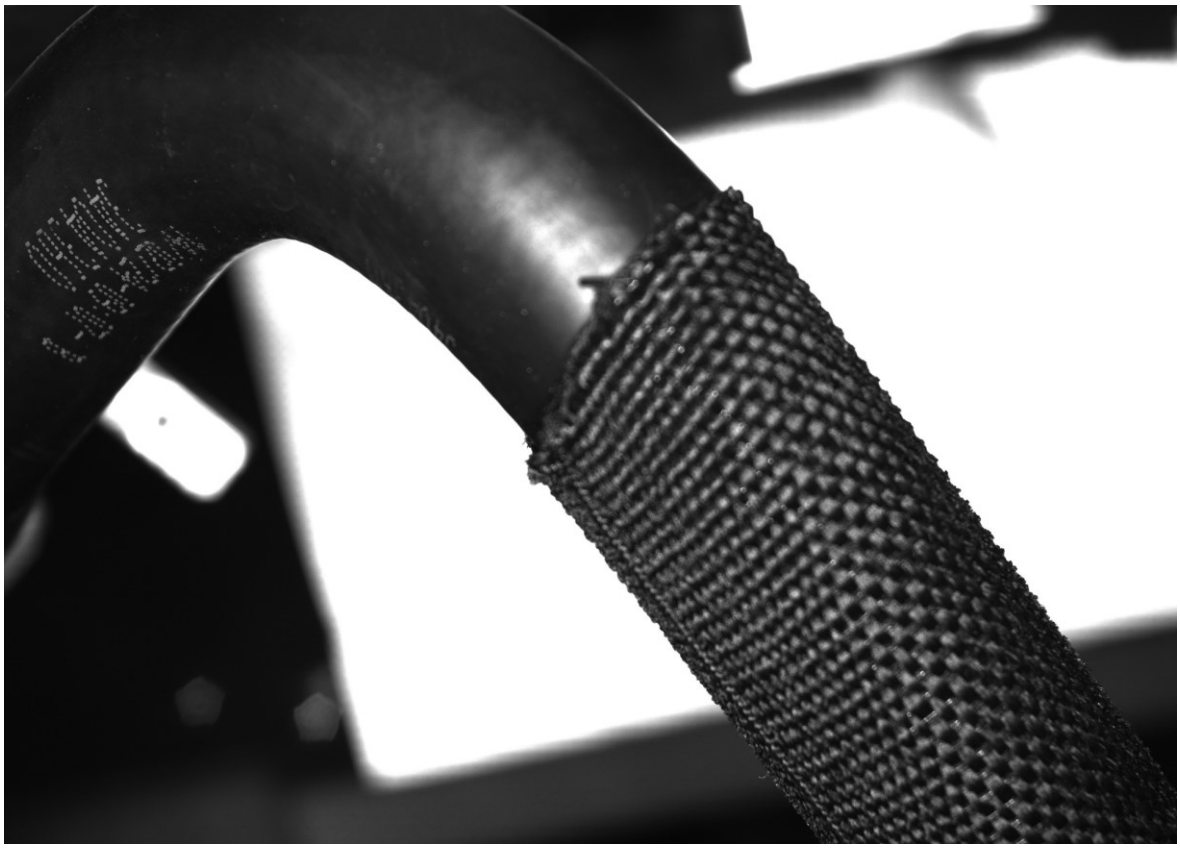
5.3 Detekce pozice rukávu na autodílu

Tato sada byla sestavena ze snímků, na kterých se snímá pozice ochranného rukávu na autodílu. Tyto snímky též pochází z reálné aplikace a též byly poskytnuty firmou AMV.

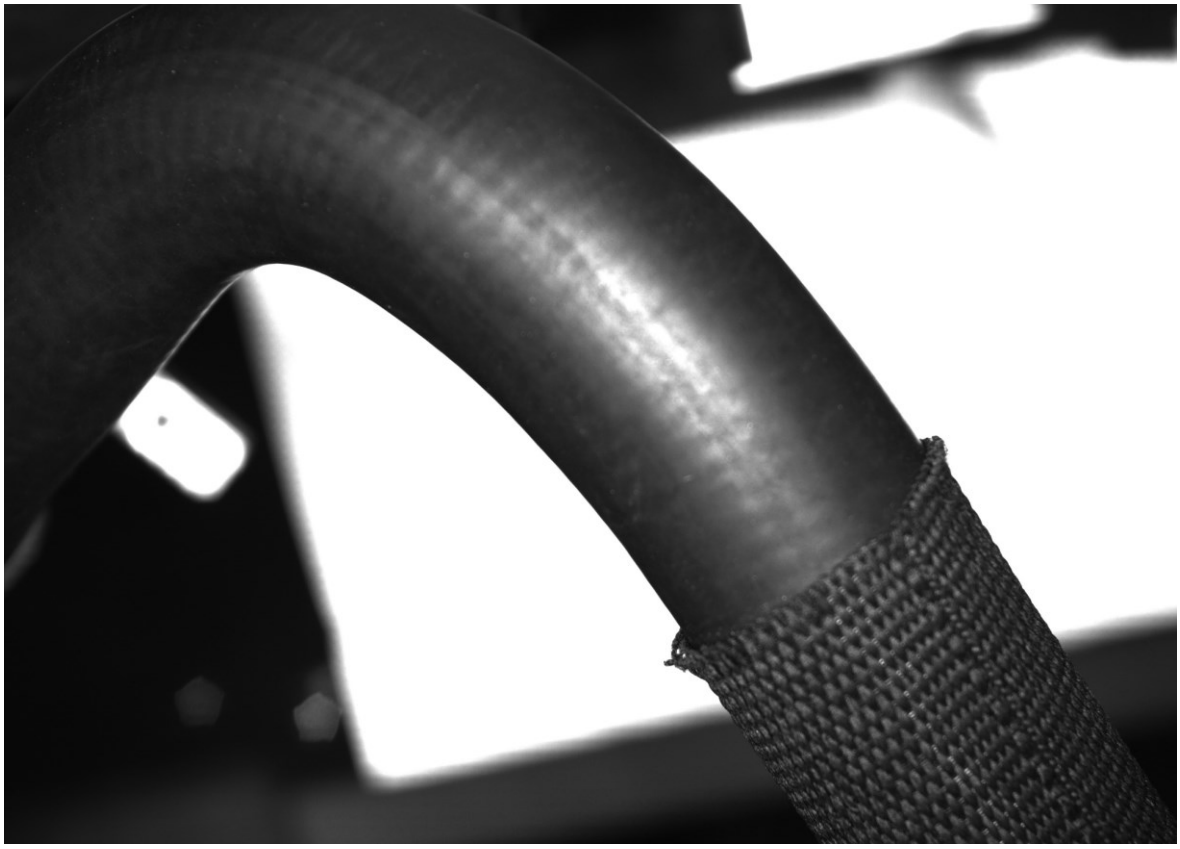
Snímková sada je ideálním případem pro nasazení neuronové sítě s nástrojem Locate, jelikož se tvar i pozice rukávu dost mění, tudíž je použití jiných metod dost komplikované až nemožné.

V úloze typu Locate se snímky nezařazují do kategorií OK a NOK, protože cílem je najít na každém snímku přesně jeden rukáv. V reálné aplikaci by se nadefinovalo správné umístění nastavením ROI (Region Of Interest) a snímky kde by byl rukáv nalezen by byly OK a kde by nebyl nalezen by byly NOK.

V snímkové sadě jsou různé variace nasazení rukávu, na některých variacích je rukáv tak vysoko, že hledaná hrana rukávu se na obrázku nenachází, tudíž nemůže být detekována. Na některých obrázcích rukáv úplně chybí, ale většinou je různě naklopen a hrana se dost liší, protože na některých snímcích je roztřepená a někdy velmi ucelená.



Obrázek 21. Příklad pozice rukávu naklopený okraj



Obrázek 22. Příklad pozice rukávu ucelený okraj

6 POUŽITÝ HARDWARE

Pro konzistentnost výsledků, byl na všechny testy použit stejný hardware. Jak již bylo popsáno v teoretické části, technologie neuronových sítí je extrémně náročná na procesorový výkon.

Většina moderních neuronových sítí je optimalizována na grafické procesory (GPU). Takže je ani není možné na klasickém CPU zpustit. Toto se týká i obou testovaných neuronových sítí.

Trénování neuronových sítí, zejména hlubokých neuronových sítí (Deep Neural Networks, DNN), je extrémně výpočetně náročné. Tento proces zahrnuje dopředné šíření (Forward Propagation) což je výpočet výstupu sítě pro dané vstupní hodnoty. A také zpětné šíření chyby (Backpropagation), což je aktualizace vah pomocí gradientních sestupových algoritmů (popsáno v teoretické části, kapitola 1.3.2), což může být velmi náročné na procesorový výkon.

GPU procesory byly zvoleny kvůli jejich schopnosti paralelně zpracovávat velké množství dat, tím že mají více jader, může být práce rozdělena mezi ně. Což je ideální pro operace jako je maticové násobení což je nejběžnější výpočet u neuronové sítě, ale využívá se i paralelního sčítání, násobení, čímž je dramaticky zkrácena doba učení. Další důležitou vlastností je velká propustnost paměti. GPU procesory dokáží zapisovat a číst velké množství dat, což je klíčové u zpracování učebního modelu hlubokých neuronových sítí.

Proto byla použita výkonná PC sestava:

- CPU- Intel(R) Core(TM) i9-14900KF 3.20 GHz
- GPU – Nvidia GeForce RTX 3070 TI
- 48 GB RAM

Obrázek 23.Grafická karta RTX 3070 TI [23]

7 TESTOVÁNÍ NASAZENÍ NEURONOVÉ SÍTĚ

V této kapitole proběhne testování průmyslových nástrojů Cognex Vision a Aurora Vision na různých typech úloh. Cílem je důkladně zhodnotit jejich schopnosti v oblasti strojového vidění, změřit dobu učení a analyzovat přesnost dosažených výsledků.

Vzhledem k značným rozdílům v architektuře a funkcích obou nástrojů budou testovány samostatně. Výsledky testování budou následně zhodnoceny a provede se komparace silných a slabých stránek daných nástrojů.

Testování nástroje Cognex Vision se zaměří na zhodnocení jeho kapacity zvládat různé typy úloh, analýzu doby učení a posouzení přesnosti dosažených výsledků. Nástroj Cognex Vision umožňuje uživateli nastavit počet učících epoch, nicméně předčasné ukončení učení není implementováno.

V případě Aurora Vision bude testování probíhat s cílem dosažení co nejpodobnějších výsledků jako u Cognex Vision, a to s přihlédnutím k identickým parametrům a srovnatelné délce učení. Aurora Vision umožňuje průběžné sledování procesu učení a predikční přesnosti, čímž dává uživateli možnost učení v případě dosažení požadované přesnosti ukončit. U softwaru Aurora Vision nebude testován nástroj Locate object (sada s rukávem), protože byl z aktuální verze nástroje dočasně odstraněn, viz patch notes: „Removed: Two tools are temporarily not available - DL_LocateObjects and DL_SegmentInstances. We are still working on porting them to the new training service. DL_LocateObjects will be coming first.“[22]

7.1 COGNEX VISIONPRO DEEP LEARNING

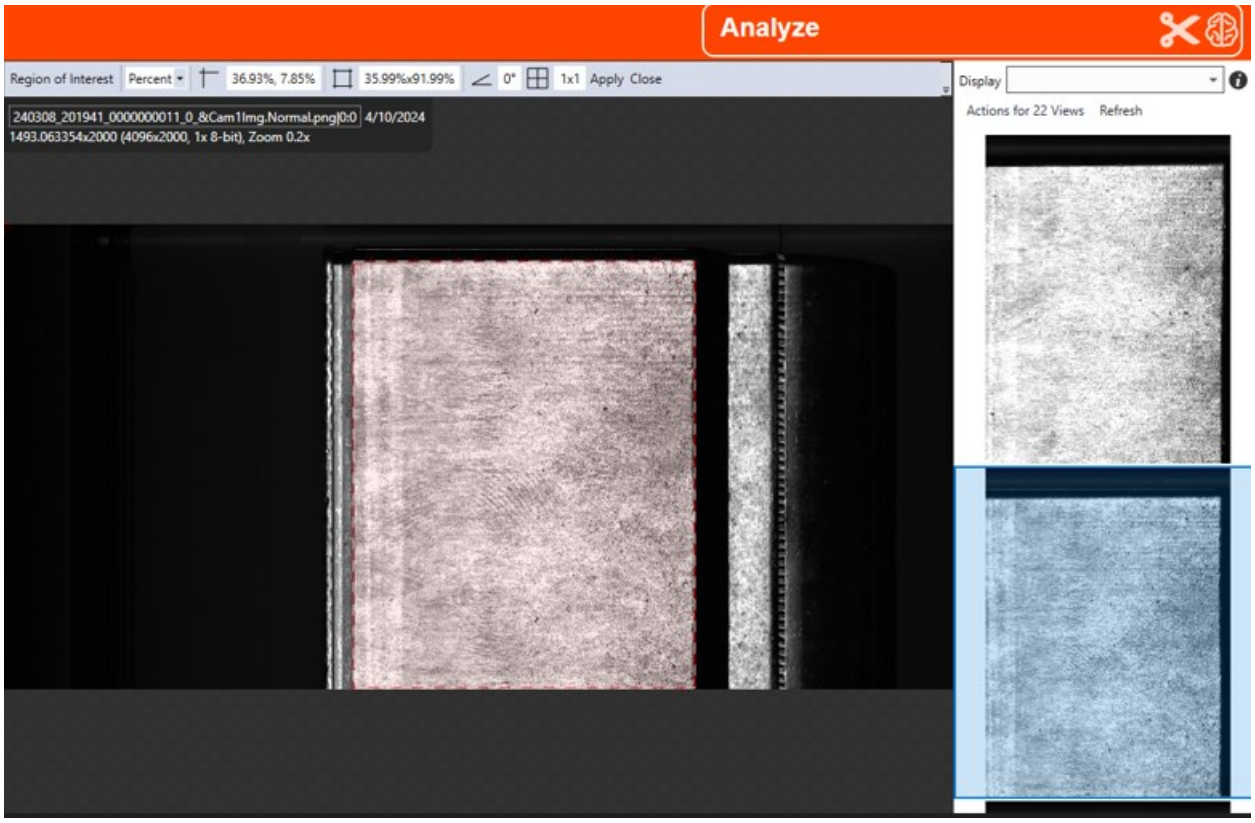
7.1.1 Analyze, vyhledávání povrchových vad

Cílem tohoto testu je experimentovat s minimálním počtem naučených dat a sledovat závislost počtu dat na přesnosti výstupních výsledků.

Objektem sledování jsou povrchové vady popsané v kapitole 5.

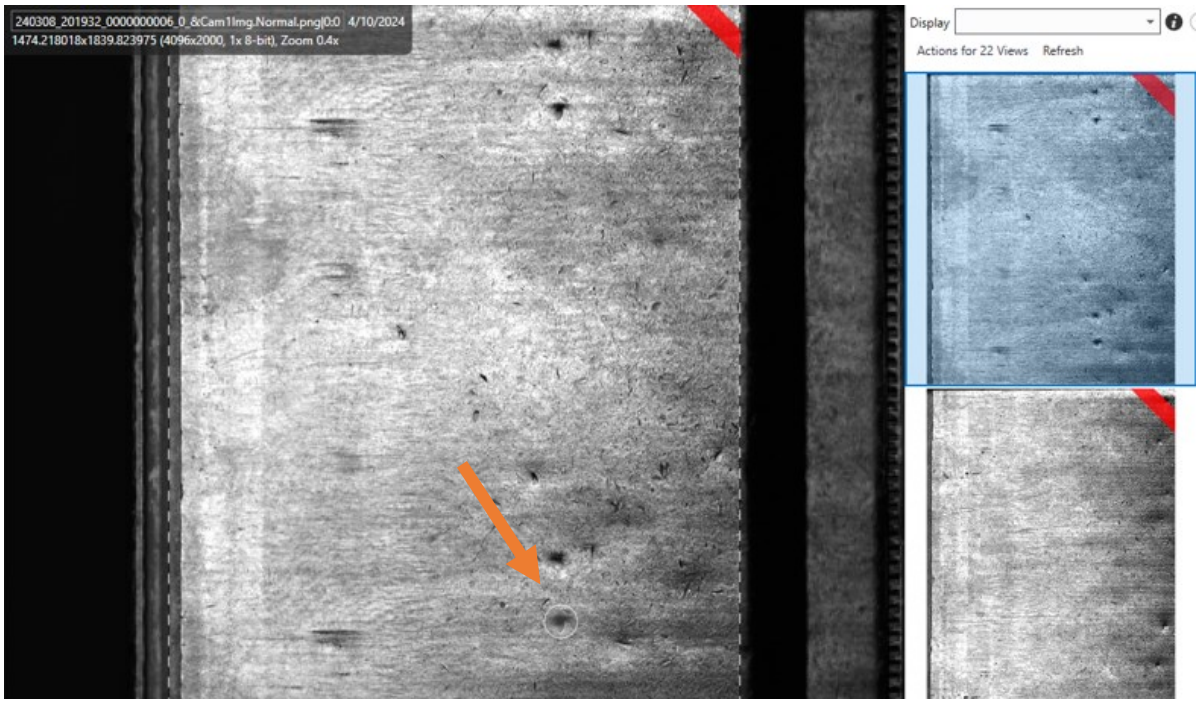
Při prvotním testování tohoto nástroje bylo zjištěno, že ke korektnímu nalezení vad mu stačí i velmi malý počet vstupních dat. Proto byla zvolena malá snímková sada, skládající se z 22 snímků (12 OK snímků a 10 NOK snímků). Byly zařazeny i snímky které se nedají jasně zařadit do jedné či druhé skupiny.

Pro konzistentní výsledky byl u všech testů tohoto typu použit ROI se stejnou velikostí. ROI se udává v % z celkového obrazu.



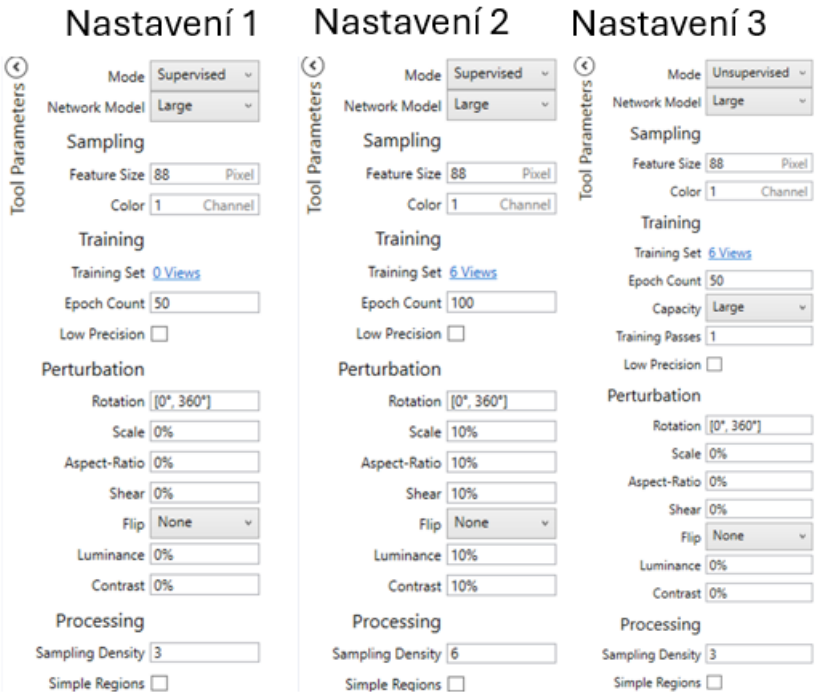
Obrázek 24. Nastavení ROI

Feature size bylo nastaveno na velikost odpovídající průměrné velikosti hledané vady, což odpovídá 88 pixelům.



Obrázek 25. Nastavení Feature size

Test bude proveden pro 3 různé nastavení učení modelu.



Obrázek 26. Nastavení 1-3 pro nástroj Analyze povrchové vady

Nastavení 1:

V tomto testu byly ponechány defaultní hodnoty, jenom parametr rotace byl nastaven na možný výskyt v 360° rotaci. Cílem je zjistit, jak bude neuronová síť měnit svou predikci rozřazení snímků na základě počtu naučených snímků s tím, že počet snímků bude poměrně malý.

Výsledky budou prezentovány pomocí matice zmatení (confusion matrix). Z té se dá zjisti jak neuronová síť predikovala výsledky oproti realitě a také jaké snímky vyhodnotila jako falešně pozitivní či falešně negativní. Počet nejistých snímků (Inter) závisí na nastavení trash holdu. Naučené snímky značí, na kolika snímcích byl konkrétní model učen.

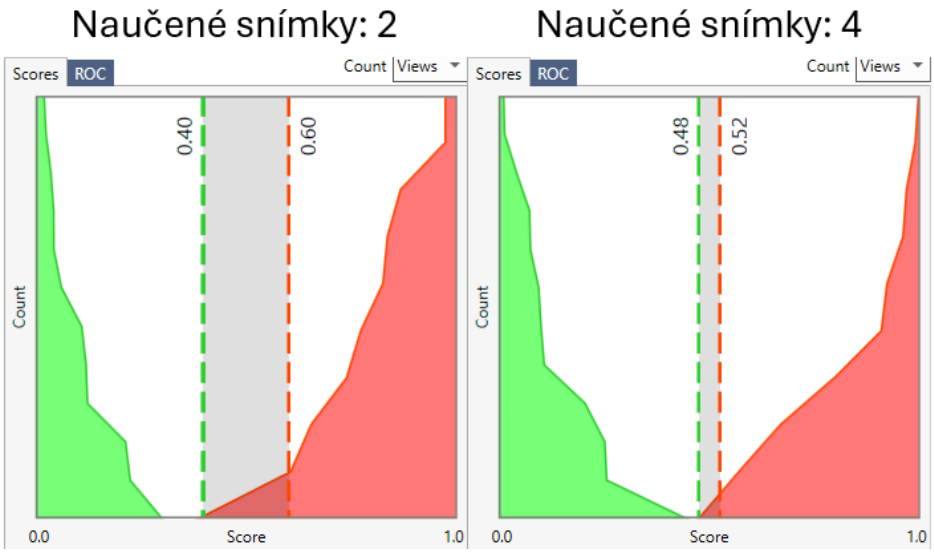
Naučené snímky: 2

		Predicted			
		Good	Inter	Bad	Total
Actual	Good	12	0	0	12
	Bad	1	0	9	10

Naučené snímky: 4

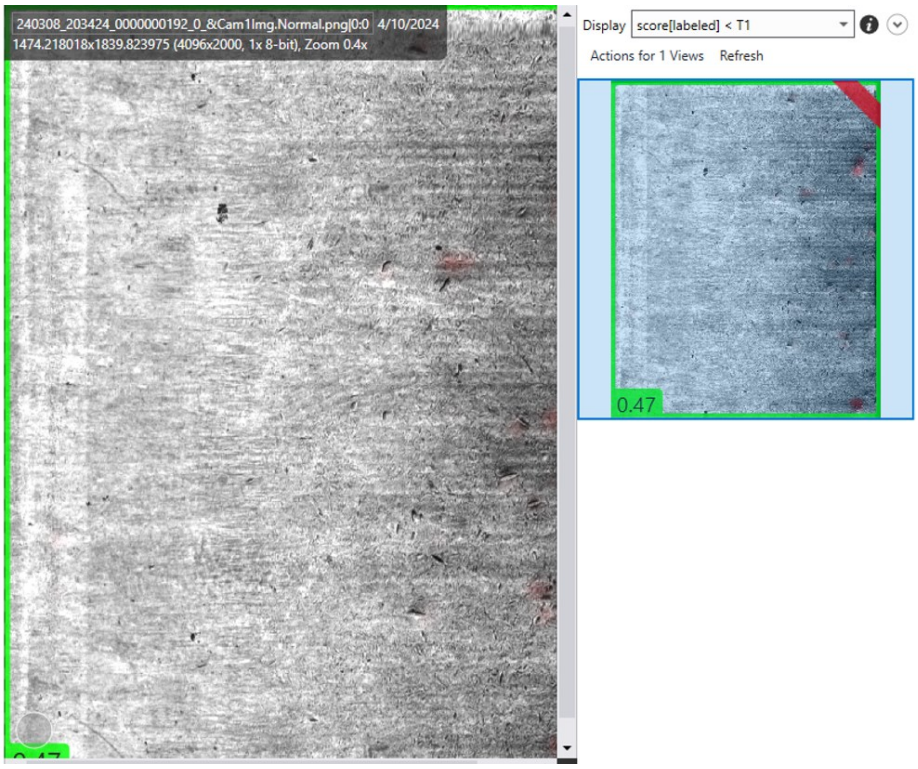
		Predicted			
		Good	Inter	Bad	Total
Actual	Good	12	0	0	12
	Bad	1	0	9	10

Obrázek 27. Výsledné matice zmatení pro Nastavení 1



Obrázek 28. Výsledné grafy predikcí pro Nastavení 1

Z výsledné matice zmatení vyplývá, že neuronová síť byla schopná předpovědět výsledky správně, až na jedinou chybu, pouze na základě dvou naučených snímků. Avšak vzhledem k mezeře mezi kategoriemi OK a NOK na grafu, je možné tuto chybu odstranit změnou prahu. Z grafu je rovněž patrné, že neuronová síť úspěšně identifikovala OK snímky s vysokou přesností. U snímků NOK nebyla přesnost tak jasná. Stojí za zmínku, že snímek, který neuronová síť nesprávně klasifikovala jako správný, je velmi sporný a jeho vada není jednoznačná. Neuronová síť mu přiřadila výstup 0,39.



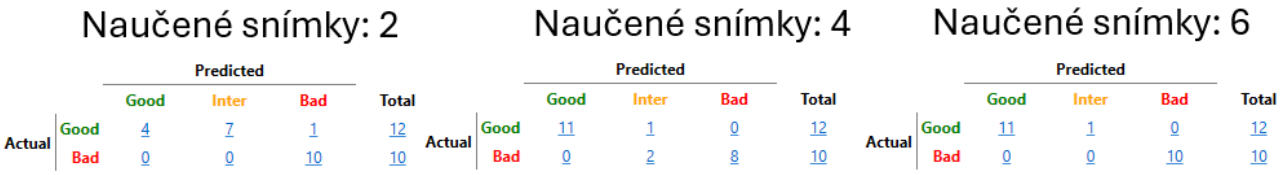
Obrázek 29. Špatně zařazený snímek

Zde je vidět, že na snímku se nacházejí dvě mírné lokální povrchové nerovnosti, které neuronová síť dokonce označila, ale nepovažovala je za dostatečné poškození na to, aby tento snímek zařadila jako NOK.

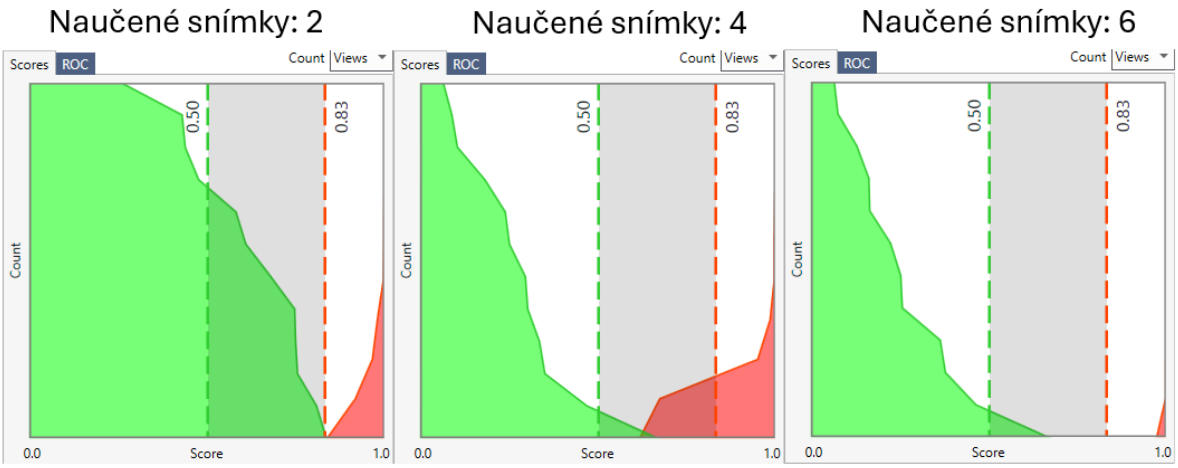
Po naučení dalších dvou snímků vznikl zajímavý jev. Neuronová síť začala dávat jistější bodové hodocení NOK snímkům, ale začala dávat méně jednoznačné výsledky pro OK snímky, takže některé parametry neuronové sítě se zlepšily ale jiné se zase zhoršily. Hodnocení nejistého snímku se změnilo na 0,47 což už velmi dobře odpovídá spornému hodnocení, takže se dá říci, že celkové parametry neuronové sítě se zlepšily, protože dokáže neurčité snímky ohodnotit odpovídajícím hodnocením.

Nastavení 2:

Ve druhém nastavení byl Epoch time zdvojnásoben, což znamená, že model provede dvojnásobný počet iterací při korekci chyb. Toto prodloužení času učení by mohlo mít pozitivní dopad na přesnost modelu. Všechny parametry Perturbation byly nastaveny na 10%, protože vady na NOK snímcích mají různý charakter. Toto nastavení umožňuje umělé zvýšení rozmanitosti učebních snímků, což by mohlo vést k lepším výsledkům. Sampling density byla zvýšena na hodnotu 6, což sice zpomalí proces učení modelu, ale může mít pozitivní vliv na jeho schopnost přesně predikovat výsledky.



Obrázek 30 Výsledné matice zmatení pro Nastavení 2



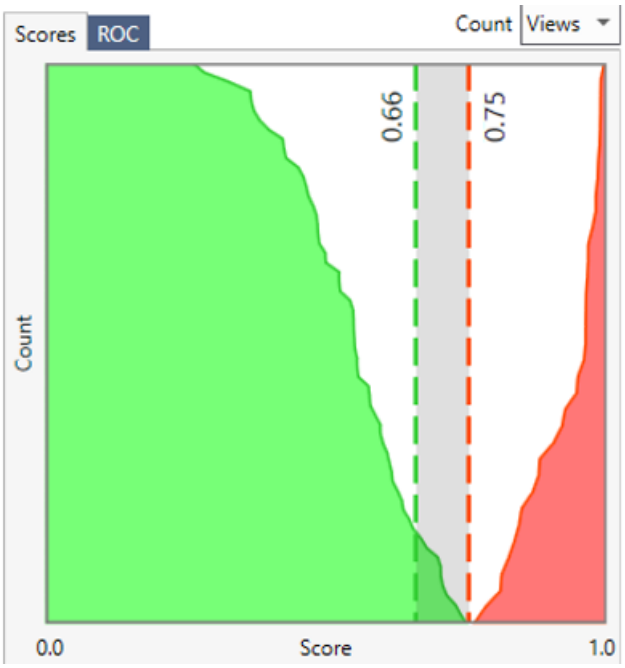
Obrázek 31. Výsledné grafy predikcí pro Nastavení 2

Z výsledků je zřejmé, že při tréninku s pouhými 2 snímky dokáže model spolehlivě identifikovat většinu NOK snímků. Avšak kvůli tomu, že 7 OK snímků bylo zařazeno do neurčité kategorie, je zřejmé, že model není schopen spolehlivě rozlišovat OK snímky. Po dodatečném tréninku s dalšími dvěma snímky se výrazně zlepšila přesnost modelu. při rozpoznávání OK snímků. Nicméně opět se objevil stejný jev jako v předchozích případech,

kdy model začal méně spolehlivě rozpoznávat NOK snímky a nebyl si tak jistý jejich zařazením.

Po naučení celkem 6 snímků, je model schopen s velkou jistotou rozlišit všechny NOK snímky. Jistota zařazení OK snímků zůstává stejná, jako při 4 snímcích. Mezera mezi snímky OK a NOK je nyní poměrně velká, takže model je schopen solidně rozlišit OK snímky od NOK. Pouze snímek s neurčitou chybou byl zařazen do neurčité kategorie, což je opět správné chování.

Poslední model byl nasazen na větší sadu snímků, celkem 95 různých, z nichž 60 bylo označeno jako OK a 35 jako NOK. Mezi snímky se nachází několik takových, u kterých není jednoznačné, zda obsahují vadu či nikoliv.

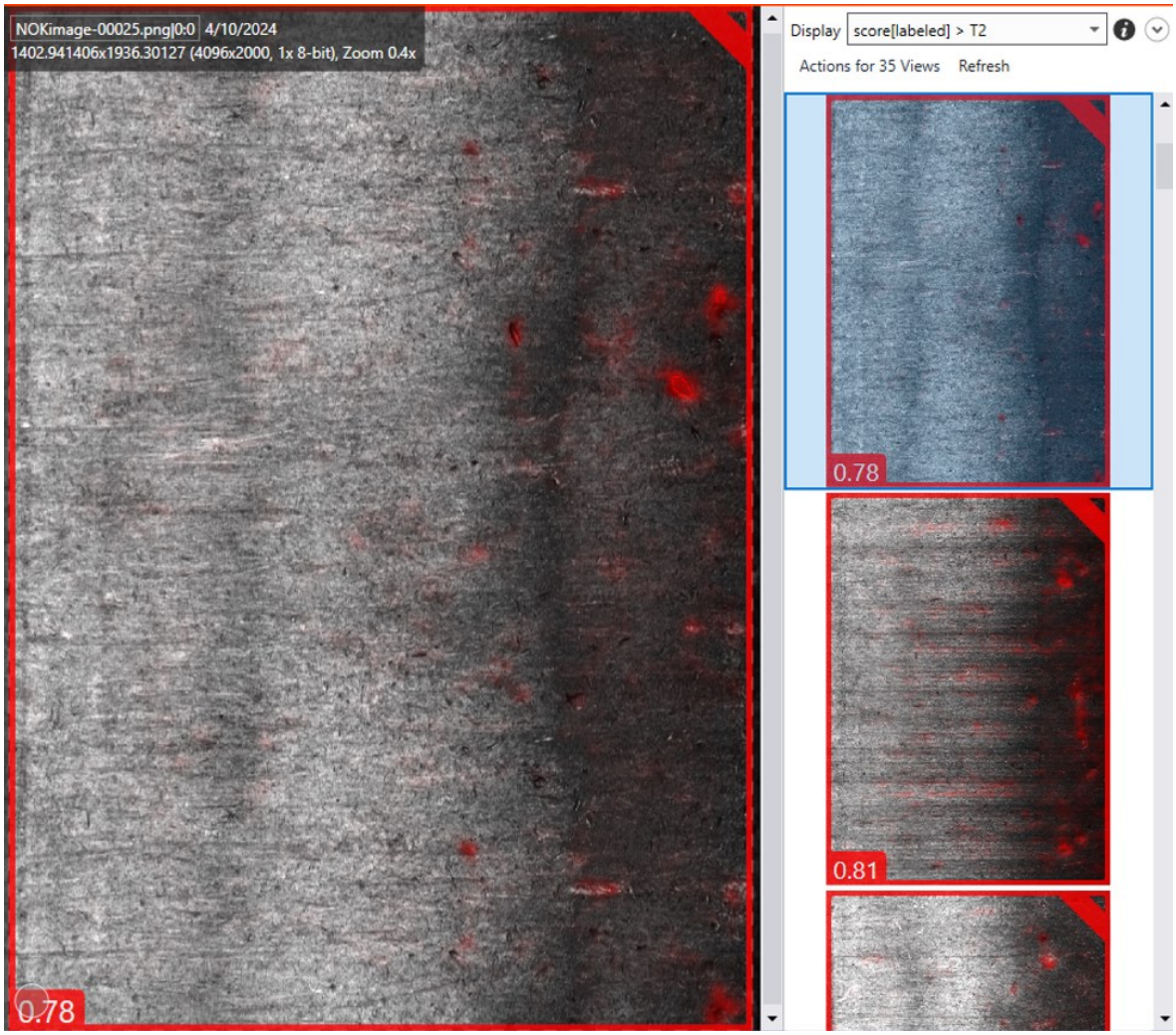


Obrázek 32. Výsledný graf predikce na velké sadě

		Predicted			
		Good	Inter	Bad	Total
Actual	Good	50	10	0	60
	Bad	0	0	35	35

Obrázek 33. Výsledná matice zmatení u velké sady

Z výsledných dat je patrné, že model naučený na 12 snímcích byl schopen celkem spolehlivě rozeznat většinu snímků jako OK či NOK. Pouze u 10 z nich nebyl schopen jednoznačně určit, zda se jedná o OK nebo NOK snímky, a z těchto 10 bylo 6 snímků nejednoznačných, což znamená, že nebylo jasné, zda vada na snímku je OK nebo NOK. Avšak s výjimkou těchto případů byl model schopen bezpečně rozeznat většinu snímků.



Obrázek 34. Ukázka rozlišovacích schopností nástroje

Nastavení 3:

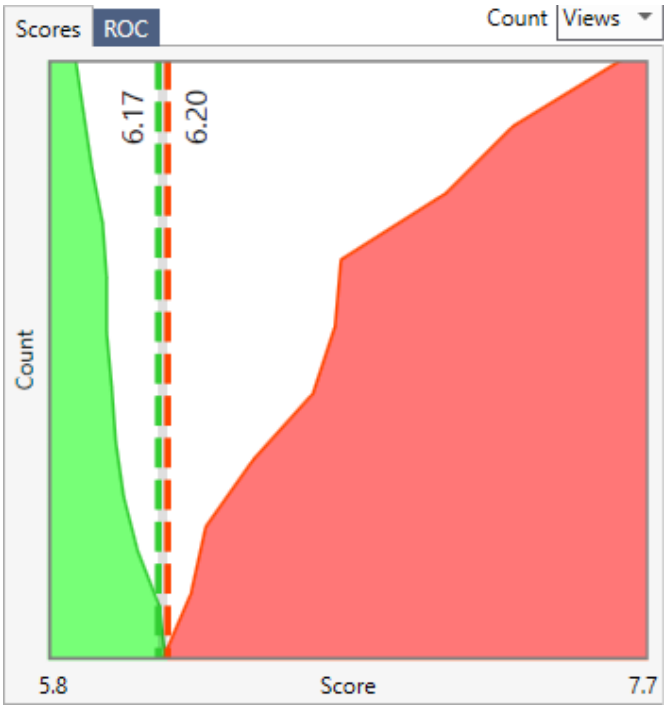
V nastavení 3 byly všechny parametry ponechány stejné jako v nastavení 1, ale typ učení neuronové sítě byl změněn na učení bez učitele, což znamená že neuronová síť se bude snažit najít ve vstupních datech opakující se vzorce a oddělit od nich výkyvy. Cílem tohoto měření je porovnat rozdíly ve výstupu neuronové sítě na základě změny typu učení modelu.

Naučené snímky: 6

Confusion Matrix

		Predicted			
		Good	Inter	Bad	Total
Actual	Good	11	1	0	12
	Bad	0	1	9	10

Obrázek 35. Výsledná matice zmatení pro zmatení 3



Obrázek 36. Výsledný graf predikcí pro Nastavení 3

Při testování modelu učení bez učitele bylo zjištěno, že naučený model poskytoval mnohem horší výsledky pro menší počet naučených snímků. Do 6 snímků poskytoval v podstatě nesmyslné výsledky a shlukoval data, která nebyla pro aplikaci vůbec relevantní. Teprve po naučení 6 snímků začal model správně rozřazovat data podle označení OK a NOK.

Zajímavý je fakt, že model byl schopen mnohem přesněji rozlišovat OK snímky od NOK, což je opačné chování než u modelu v rámci učení s učitelem. Myslím si, že to může být způsobeno tím, že při učení bez učitele model spoléhá na vnitřní strukturu dat a přirozené shlukování, což může vést k odlišným výsledkům v porovnání s tréninkem s učitelem. Bez učitele se model učí pouze na OK snímkcích, hledá v ních vzorce dat a pak označí vše, co se tomu vymyká a u učení s učitelem se model učí primárně z toho jak vypadá konkrétní vada.

Srovnání časové náročnosti všech nastavení:

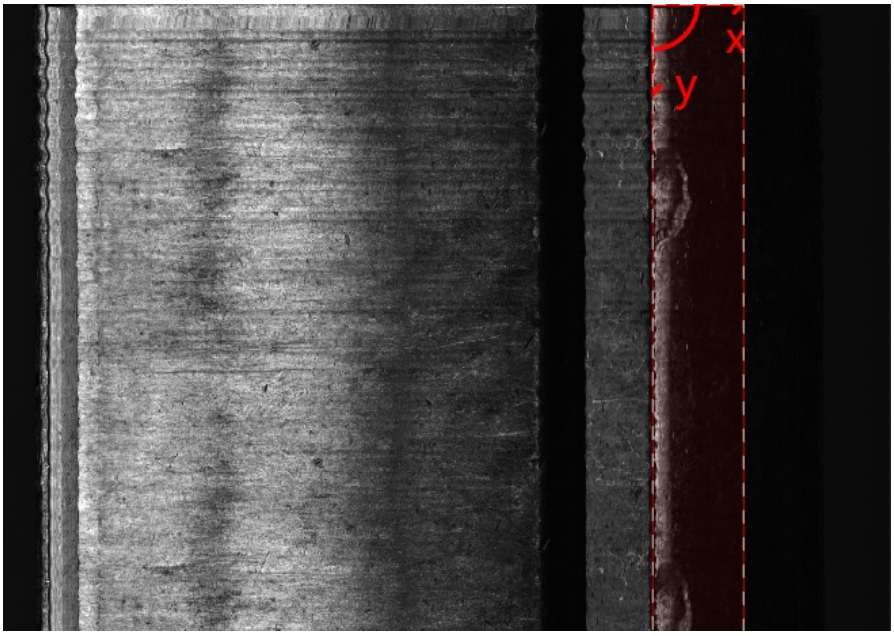
	Nastavení	Počet naučených vad	Doba učení modelu (s)	Průměrný čas zpracování (ms)	Výstup modelu	Ve skutečnosti OK.NOK
Analyze-Povrchové vady	1	2	57	6,4±0,4	OK: 13, NC:0, NOK: 9	12.10
		4	58	6,5±0,5	OK: 13, NC:0, NOK: 9	
	2	2	137	20,4±0,8	OK: 4, NC:7, NOK: 11	
		4	119	20,2±0,9	OK: 11, NC:3, NOK: 8	
		6	121	20,7±1,5	OK: 11, NC:1, NOK: 10	
	3	6	60	X	OK: 11, NC:2, NOK: 9	

Tabulka 1. Srovnání časové náročnosti všech nastavení

V tabulce jsou uvedeny doby učení jednotlivých modelů. Z dat vyplývá, že čas učení modelu s učitelem a bez učitele je velmi podobný. Velký rozdíl nastává až se změnou počtu epoch přizpůsobení a se změnou hustoty vzorkování. Toto nastavení mělo také pozitivní vliv na výstupy naučeného modelu neuronové sítě.

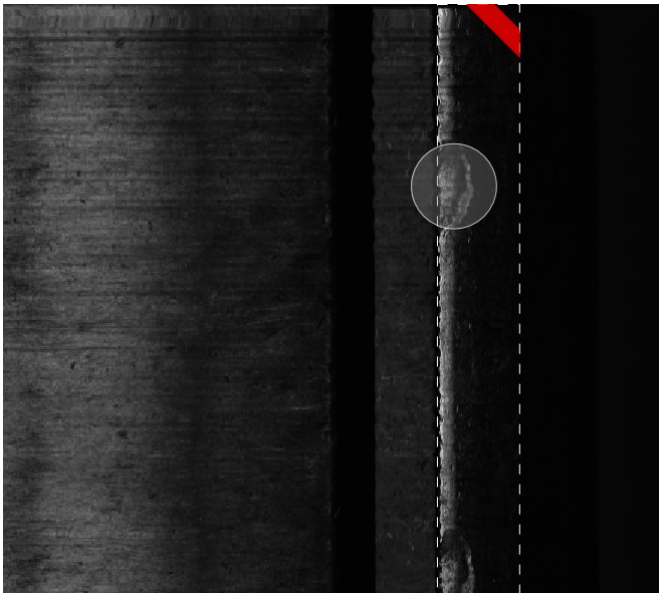
7.1.2 Analyze detekce vad na uložení projektilu

ROI bylo nastaveno jen na oblast hrany uložení projektilu, protože zkoumané vady se nachází jen v této oblasti.



Obrázek 37. Nastavení ROI

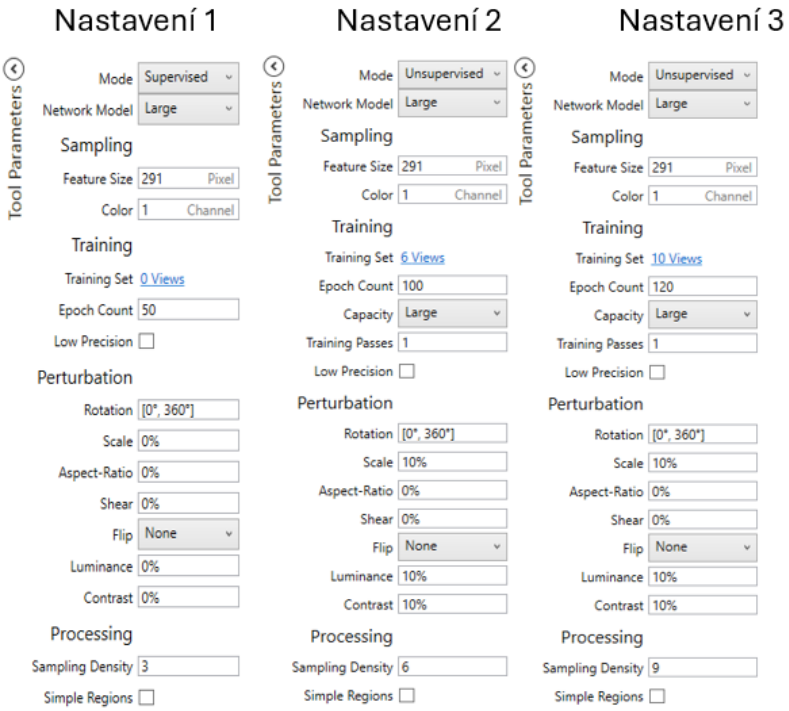
Feature size bylo nastaveno na 291 pixelů. To přibližně odpovídá průměrné velikosti detekované vady na okraji. Vady v této sadě byly opravdu velmi různorodé, a dokonce i různého typu od uštípnutých okrajů až přes zploštěné hrany.



Obrázek 38. Nastavení Feature size

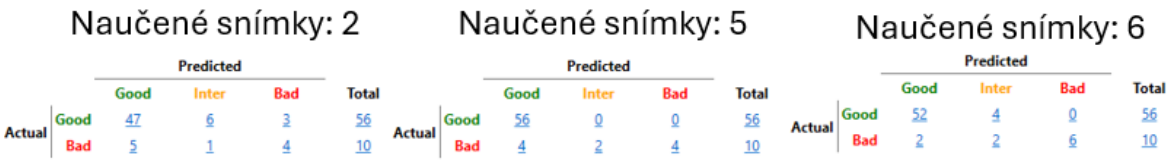
Pro 3 různá nastavení:

V prvním nastavení probíhá učení s učitelem, zatímco v dalších nastaveních se typ učení změní na učení bez učitele. Zároveň se zvyšuje epoch count a parametry Perturbation. Ve třetím nastavení se epoch count ještě více zvýší. Sample density se také navýší.

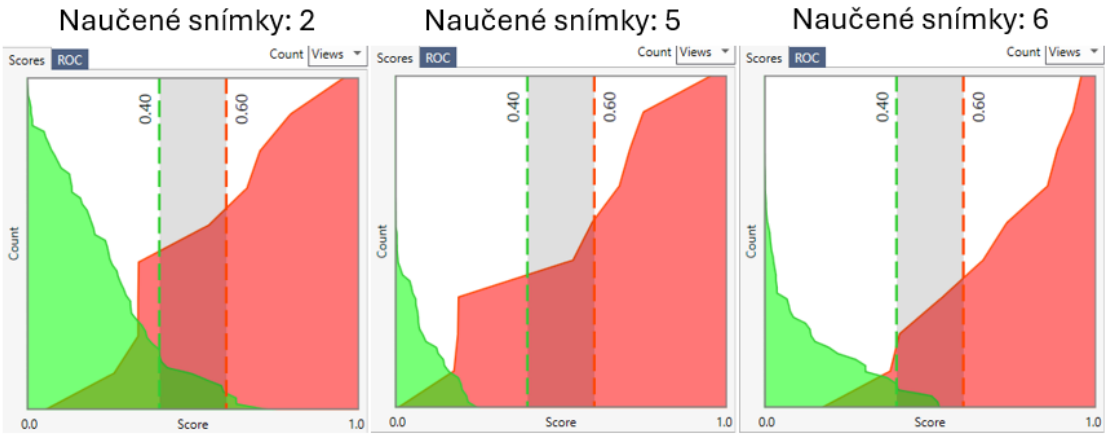


Obrázek 39. Nastavení

Nastavení 1:



Obrázek 40. Výsledné matice zmatení pro nastavení 1

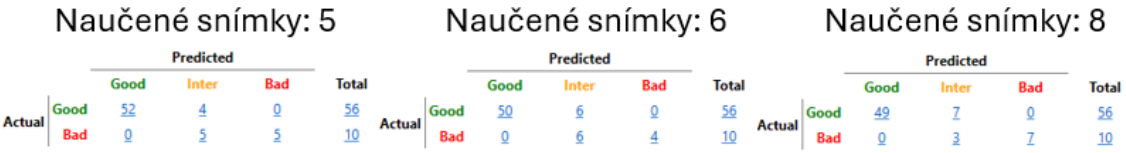


Obrázek 41.Výsledné grafy predikce pro nastavení 1

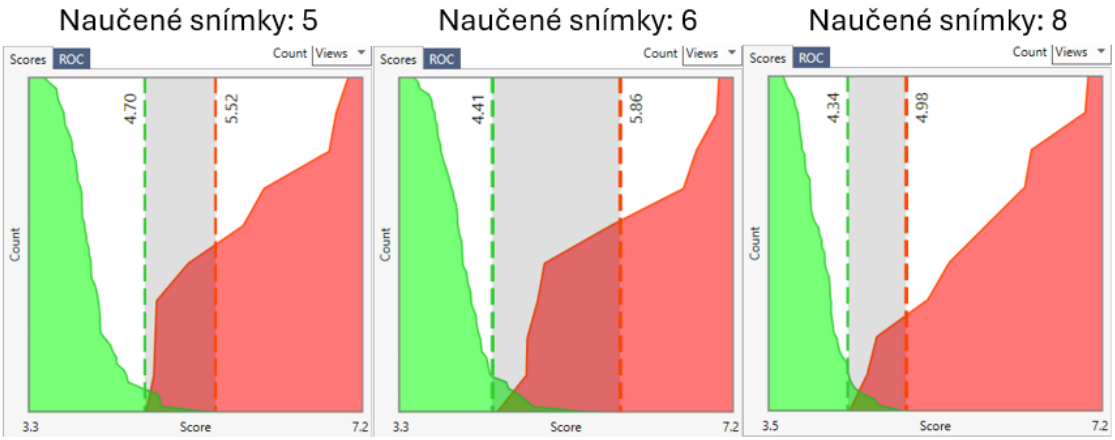
Výsledná data naznačují, že metoda učení s učitelem není pro tento typ úloh příliš vhodná. Pravděpodobně je to způsobeno velkou různorodostí vad a tím, že naučené snímky neobsahují všechny typy vad. Pro dva naučené snímky jsou výsledky dost nekonzistentní, a dokonce jeden NOK snímek byl modelem označen skóre 0, což znamená, že model s 100% jistotou považoval tento snímek za OK, což naznačuje, že model nechápe, která data jsou na snímku podstatná, a obecně naznačuje špatné naučení modelu.

I po naučení dalších tří snímků s různými vadami se výsledek příliš nezlepšil. Model se sice naučil lépe rozlišovat OK snímky s poměrně dobrou jistotou, ale stále téměř vůbec nerozlišuje NOK snímky.

Nastavení 2:



Obrázek 42. Výsledné matice zmatení pro nastavení 2



Obrázek 43. Výsledné grafy predikce pro nastavení 2

Metoda učení bez učitele se zdá pro tento typ úlohy s různorodými vadami vhodnější. Jelikož se model učí pouze jak má vypadat OK snímek a pak jen shlukuje podobná data k sobě. Ty, které se příliš nepodobají označí jako NOK. Z předchozích experimentů vyšlo najevo, že učení bez učitele vyžaduje větší množství naučených dat, proto byl první model naučen na 5 snímcích.

Na výsledném modelu se projevuje efekt přeučení, kdy při 8 naučených snímcích a velkém počtu iterací se již chyba predikce zhoršuje.

Nastavení 3:

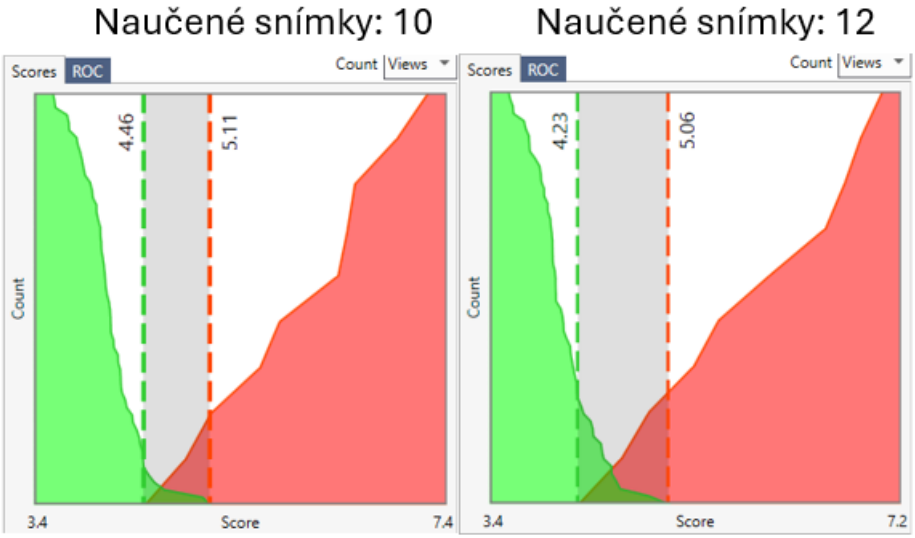
Naučené snímky: 10

		Predicted			
		Good	Inter	Bad	Total
Actual	Good	51	5	0	56
	Bad	0	2	8	10

Naučené snímky: 12

		Predicted			
		Good	Inter	Bad	Total
Actual	Good	41	15	0	56
	Bad	0	3	7	10

Obrázek 44. Výsledné matice zmatení pro nastavení 2



Obrázek 45. Výsledné grafy predikce pro nastavení 3

Na těchto výsledcích měl být prezentován efekt přeučení neuronové sítě. Jelikož v tomto případě byl epoch count nastaven na 120, tak výsledky pro 5,6,8 naučených snímků byly velmi podobné a ani pro 10 snímků není výstup modelu o moc lepší. Po 12 naučených snímcích se výsledky dokonce ještě více zhoršují. To bude pravděpodobně způsobeno přílišným přizpůsobením modelu popsaném v kapitole 2.9.

Srovnání časové náročnosti všech nastavení:

	Nastavení	Počet naučených vad	Doba učení modelu (s)	Průměrný čas zpracování (ms)	Výstup modelu	Ve skutečnosti OK.NOK
Analyze-Hrana uložení projektilu	1	2	34	1,7±0,2	OK: 52, NC:7, NOK: 7	56.10
		5	32	1,7±0,3	OK: 60, NC:2, NOK: 4	
		6	35	1,7±0,4	OK: 52, NC:6, NOK: 6	
	2	5	129	1,7±0,5	OK: 52, NC:9, NOK: 5	
		6	119	1,7±0,6	OK: 50, NC:12, NOK: 4	
		8	116	1,7±0,7	OK: 49, NC:10, NOK: 7	
	3	10	141	1,7±0,8	OK: 51, NC:7, NOK: 8	
		12	136	1,7±0,9	OK: 41, NC:18, NOK: 7	

Tabulka 2. Srovnání časové náročnosti všech nastavení

Z výsledků pro učení s učitelem je patrné, že nastavení ROI má velký vliv na dobu učení modelu, což je pravděpodobně způsobeno celkovým zmenšením objemu vstupních dat. I prohledávaná oblast (oblast zpracování) se oproti předchozí úloze výrazně zmenšila.

Z dat také vyplývá, že zvýšení počtu přizpůsobovacích epoch má velký vliv na dobu učení modelu a také na přesnost a výskyt jevu přeučení modelu.

7.1.3 Locate ochranný rukáv

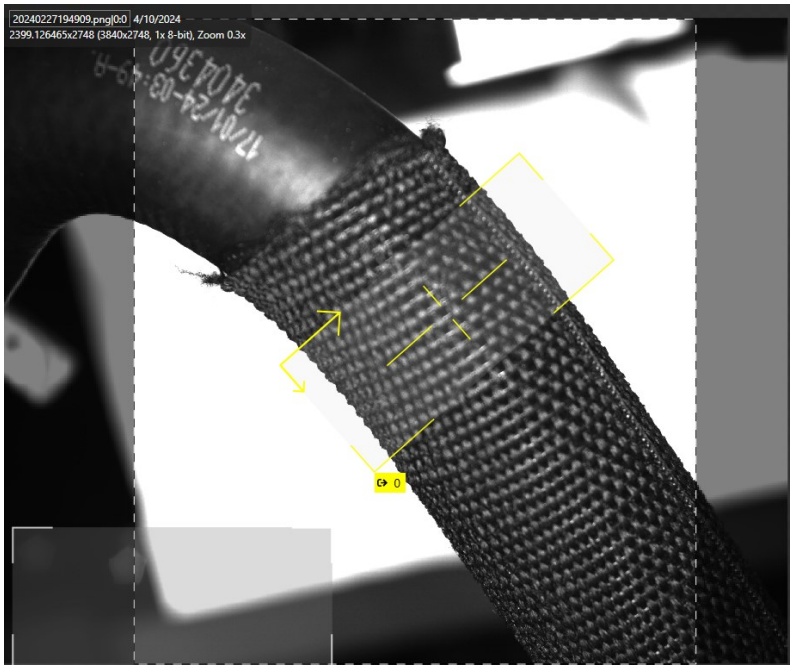
Obrázek 46. Nastavení modelu pro Locate

Nastavení 1:

Počet naučených instancí	Doba učení modelu (s)	Nenalezeno	Správně nalezeno	Špatně nalezeno	Falešně pozitivně nalezeno	Celkem snímků
1	75	4	52	6	1	63
3	71	4	58	0	1	

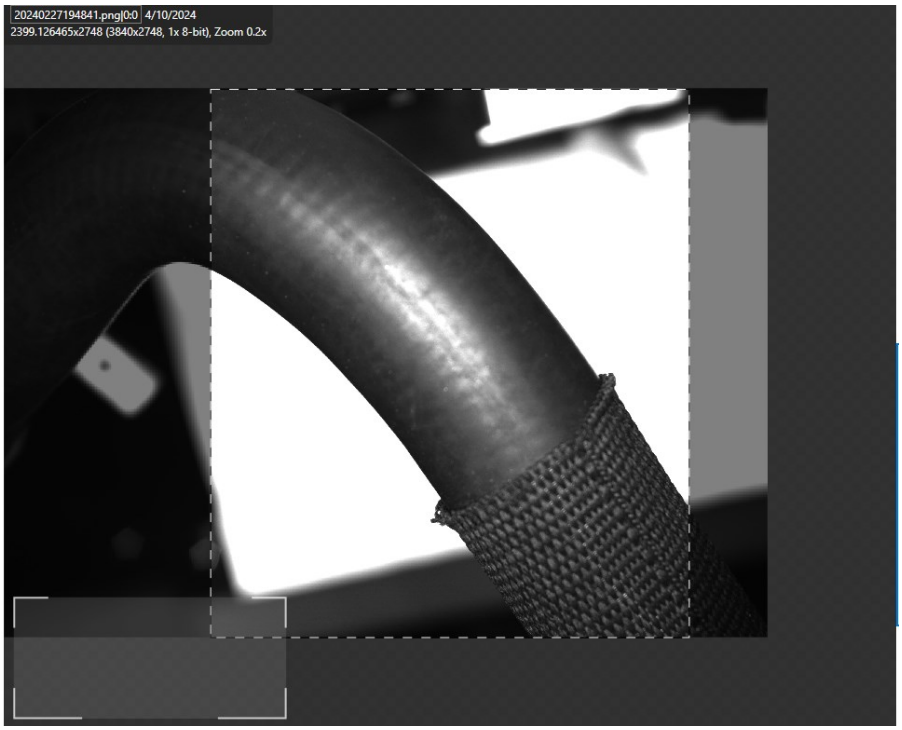
Tabulka 3. Výsledky Locate nastavení 1

Výsledky pro první nastavení ukazují, že již 1 naučený snímek stačil pro správné nalezení 52 z 63 rukávů na snímcích, na většině špatně vyhodnocených snímcích byl rukáv nalezen špatně, což je pro příklad ukázáno na obrázku 46. Na 4 snímcích nebyl rukáv nalezen vůbec, to bylo způsobeno velkou variací ve vzhledu rukávu.



Obrázek 47. Příklad špatně nalezeného rukávu.

Po naučení dalších 2 snímků už dokázal správně lokalizovat rukáv na snímcích, kde jej předtím nalezl špatně viz obrázek 46. Ovšem počet rukávů, které nenašel vůbec se nezměnil. To bylo způsobeno tím, že na snímcích se pozice a tvar rukávu dost měnily, jak ukazuje obrázek 47. Toto by se dalo vyřešit snížením trasholdu, nebo doučením těchto snímků.



Obrázek 48. Příklad nenalezeného rukávu

Nastavení 2:

Počet naučených instancí	Doba učení modelu (s)	Nenalezeno	Správně nalezeno	Špatně nalezeno	Falešně pozitivně nalezeno	Celkem snímků
1	74	1	44	2	16	63

Tabulka 4.Výsledky Locate nastavení 2

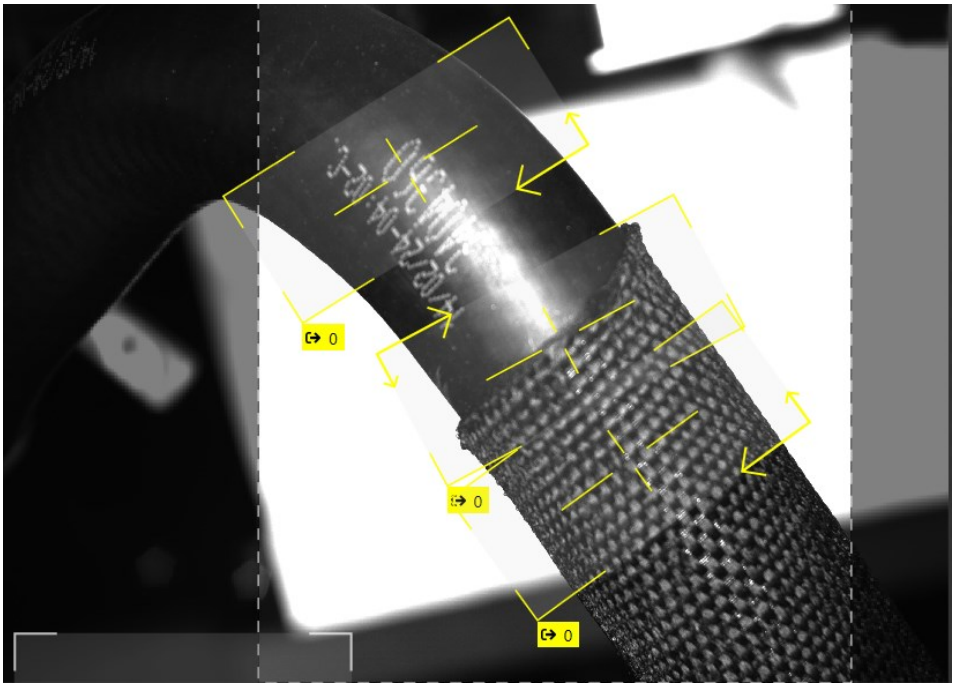
Po zvýšení preciznosti na 10 % se výstup modelu výrazně zhoršil. Model začal hledat naučený vzor i v místech, kde se nenacházel. Proto se počet falešně pozitivně nalezených výstupů zvýšil na 16.

Nastavení 3:

Počet naučených instancí	Doba učení modelu (s)	Nenalezeno	Správně nalezeno	Špatně nalezeno	Falešně pozitivně nalezeno	Celkem snímků
3	77	0	19	0	44	63

Tabulka 5. Výsledky Locate nastavení 3

Zvýšená přesnost s perturbation 10 % měla na výstup modelu velmi špatný vliv. Celkový počet falešně pozitivně nalezených výstupů se zvýšil na 44 snímků. Téměř na každém výstupu bylo několik falešně nalezených rukávů a celkově výstupy nedávaly smysl. Mohlo to být způsobeno kombinací vysoké přesnosti a parametrů perturbation, kvůli čemu model hledal rukáv i tam kde nebyl. Příkladem výstupu tohoto modelu je obrázek 48. který ukazuje že naučený model hledal kvůli nastaveným parametrům rukáv úplně všude.



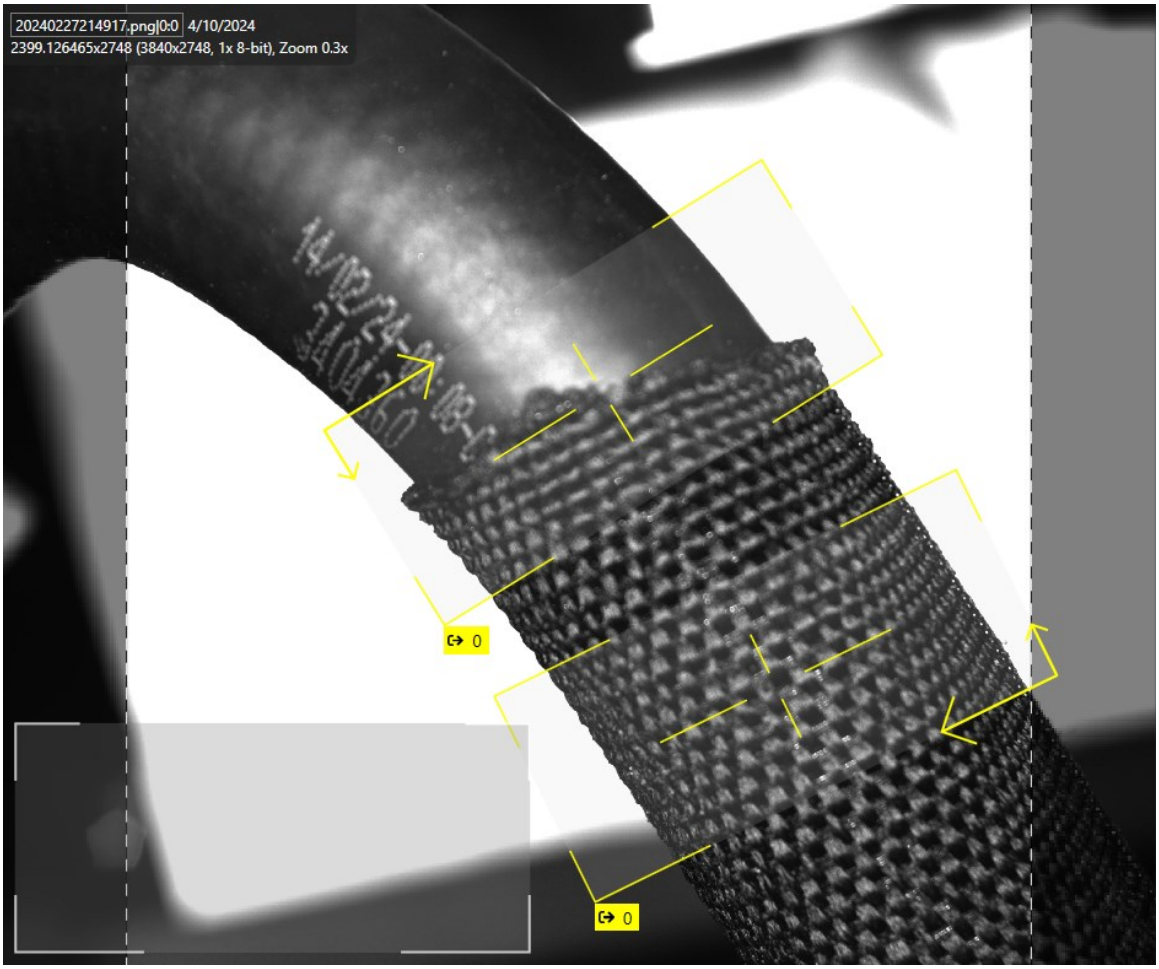
Obrázek 49. Příklad špatného výstupu, způsobeného nevhodným nastavením

Nastavení 4:

Počet naučených instancí	Doba učení modelu (s)	Nenalezeno	Správně nalezeno	Špatně nalezeno	Falešně pozitivně nalezeno	Celkem snímků
5	79	0	38	0	25	63
9	81	0	57	0	6	

Tabulka 6. Výsledky Locate nastavení 4

Výstup nastavení 4 byl poměrně zajímavý, přesto že nastavení 4 je skoro stejné jako nastavení 1. Pouze sample density bylo zvýšeno na 6, tak model poskytl o něco horší výsledky než nastavení 1. Není úplně jasné, čím byla tato chyba modelu způsobena. Po doučení dalších 4 snímků poskytl již poměrně dobré výsledky. Jediné chybné výstupy byly ty, kde na výstupu bylo nalezeno více rukávů než ve skutečnosti. Jak ukazuje obrázek 49. tak na všech špatně vyhodnocených snímcích byl vždy jeden správně nalezený rukáv, včetně orientace a jeden falešně nalezený, typicky pod správně nalezeným. Dá se předpokládat, že tento nedostatek by se dal vyřešit správným nastavením trasholdu, což bude vyzkoušeno v následujícím testu.



Obrázek 50. Příklad falešně pozitivně nalezeného rukávu

Nastavení 5:

Počet naučených instancí	Doba učení modelu (s)	Nenalezeno	Správně nalezeno	Špatně nalezeno	Falešně pozitivně nalezeno	Celkem snímků
9	72	0	62	0	1	63
10	68	0	63	0	0	
10	68	0	430	0	9	439

Tabulka 7.Výsledky Locate nastavení 5

V dalším experimentu bylo cílem doladit celou detekci tak, aby poskytovala konzistentní výsledky. Jako výchozí bod bylo použito nastavení 4. Sice poskytovalo výsledky srovnatelné s nastavením 1. až s naučením více snímků, ale důležité je že všechny chyby v nastavení 4. byly pozitivně negativní. Proto jsem předpokládal, že zvýšení trasholdu by mohlo výsledky zlepšit. Z výsledků vyplývá, že model je již dobře naučený pro řešení tohoto problému. S 9 naučenými snímky dokázal model správně najít rukáv na všech snímcích až na jeden, kde nalezl falešně pozitivně dva rukávy. Po naučení tohoto snímku již model našel všechny rukávy správně, včetně jejich orientace. Poté byl tento model vyzkoušen na celé sadě všech 439 snímků, kde úspěšně nalezl všechny rukávy s výjimkou 9 snímků, kde nalezl více rukávů, než ve skutečnosti na snímku bylo, což by se dalo pravděpodobně vyřešit dalším upravováním trasholdu.

Finální srovnání:

	Nastavení	Počet naučených instancí	Doba učení modelu (s)	Nenalezeno	Správně nalezeno	Špatně nalezeno	Falešně pozitivně nalezeno	Celkem snímků
Locate-Ochranný rukáv	1	1	75	4	52	6	1	63
		3	71	4	58	0	1	
	2	1	77	1	16	2	44	
		3	74	0	47	0	16	
	3	5	79	0	38	0	25	
		9	81	0	57	0	6	
	4	9	72	0	62	0	1	
		10	68	0	63	0	0	
		10	68	0	430	0	9	439

Tabulka 8. Finální srovnání Locate

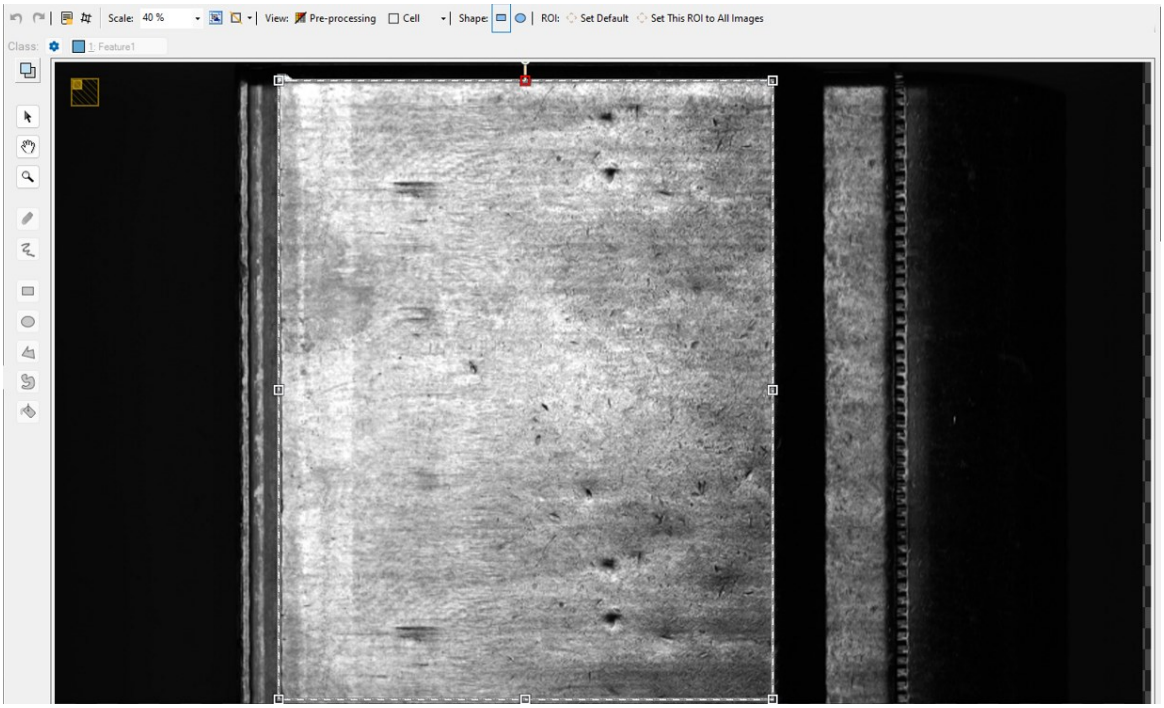
7.2 Aurora Vision Studio

7.2.1 Detect Features

Cíle tohoto experimentu jsou stejné jako u nástroje od firmy Cognex, z kapitoly 7.1.1.

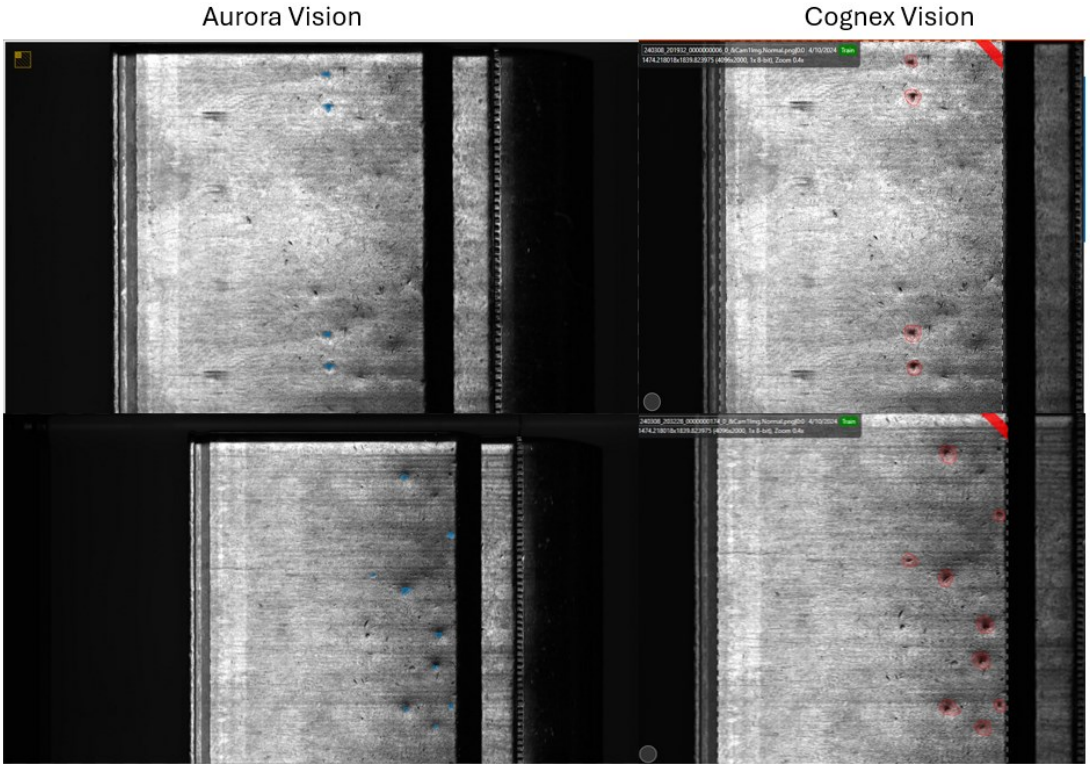
Jelikož tento nástroj jako výstup nevrací bodové ohodnocení snímků jako první nástroj, ale vrací masku ke každému snímku, na které jsou vyznačeny detekované anomálie, tak není možné exaktně určit, který snímek byl vyhodnocen jako OK nebo NOK. Proto budou výstupy nástroje pouze diskutovány a nebudou uvedeny numericky.

Pro konzistentnost výsledků byly ROI a Feature size nastaveny stejně jako u prvního experimentu s nástrojem firmy Cognex.



Obrázek 51.Nastavení ROI v prostředí Aurora Vision

Přesto, že se tyto nástroje poměrně dost liší, tak byl kladen důraz na označení stejných defektů. U nástroje Cognex Vision se hledané objekty označují štětcem velikosti feature size. U nástroje Aurora Vison je doporučeno co nejpřesněji označit hledanou vadu, a to pomocí štětců různých velikostí až do velikosti několika pixelů. Z tohoto důvodu se označení mírně liší, ale na obou snímcích byly označeny stejné vady.



Obrázek 52.Porovnání označení vad v obou softwarech

Test bude proveden pro dvě nastavení učení modelu. První bude defaultní a druhé se zapnutými parametry Augmentations, ty by měly zlepšit variabilitu modelu a obecné pochopení dat na snímcích.

Nastavení 1

Name	Value
Network Depth	3
Feature Size	80
Device	NVIDIA GeForce ...
Augmentations	
Rotation [°]	0.000
Min. Scale [%]	100.000
Max. Scale [%]	100.000
Shear Vertical Angle [°]	0.000
Shear Horizontal Angle [°]	0.000
Flip Up-Down	False
Flip Left-Right	False
Noise	0.000
Blur	0
Luminance [%]	0.000
Contrast [%]	0.000
Stopping Conditions	
Iterations	50
Training Time	Nil
Validation Score	Nil
Iterations Without Improve...	Nil
Pre-processing	
Downsample	0

Nastavení 2

Name	Value
Network Depth	3
Feature Size	80
Device	NVIDIA GeForce ...
Augmentations	
Rotation [°]	180.000
Min. Scale [%]	75.000
Max. Scale [%]	150.000
Shear Vertical Angle [°]	0.000
Shear Horizontal Angle [°]	0.000
Flip Up-Down	True
Flip Left-Right	True
Noise	25.000
Blur	25
Luminance [%]	25.000
Contrast [%]	25.000
Stopping Conditions	
Iterations	50
Training Time	Nil
Validation Score	Nil
Iterations Without Improve...	Nil
Pre-processing	
Downsample	0

Obrázek 53. Nastavení učení modelu Aurora Vision

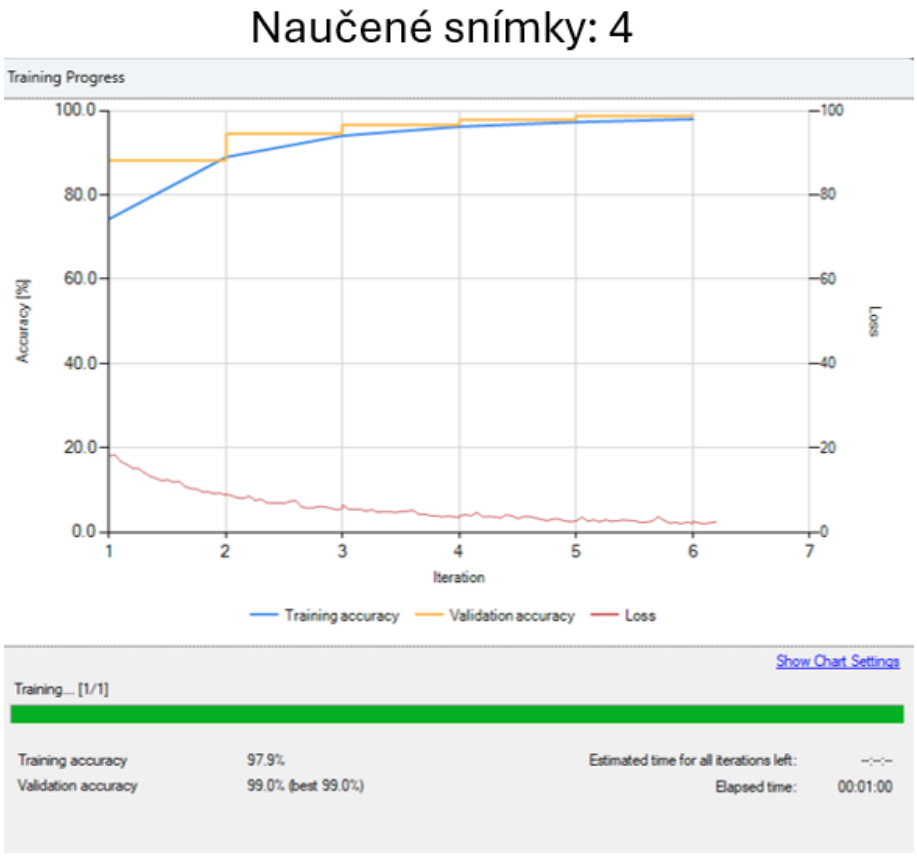
Nastavení 1:

Nastavení	Počet naučených vad	Doba učení modelu (s)	Počet Iterací	Ve skutečnosti OK.NOK
1	2	73	7	12.10
	4	74	7	

Tabulka 9.Výsledky nastavení 1 Aurora Vision

Výsledky přesnosti budou diskutovány později. Doba učení je velmi konzistentní a potvrzuje trend předchozích měření, totiž že počet snímků, na kterých se model učí, má mírný vliv na dobu učení, ale při větším počtu snímků by byl vliv významný. Zajímavý je počet iterací, protože neuronová síť od firmy Cognex dokázala za přibližně stejnou dobu provést řádově několikanásobně více iterací. Jak je však patrné z křivky učení na obrázku 48, přizpůsobení

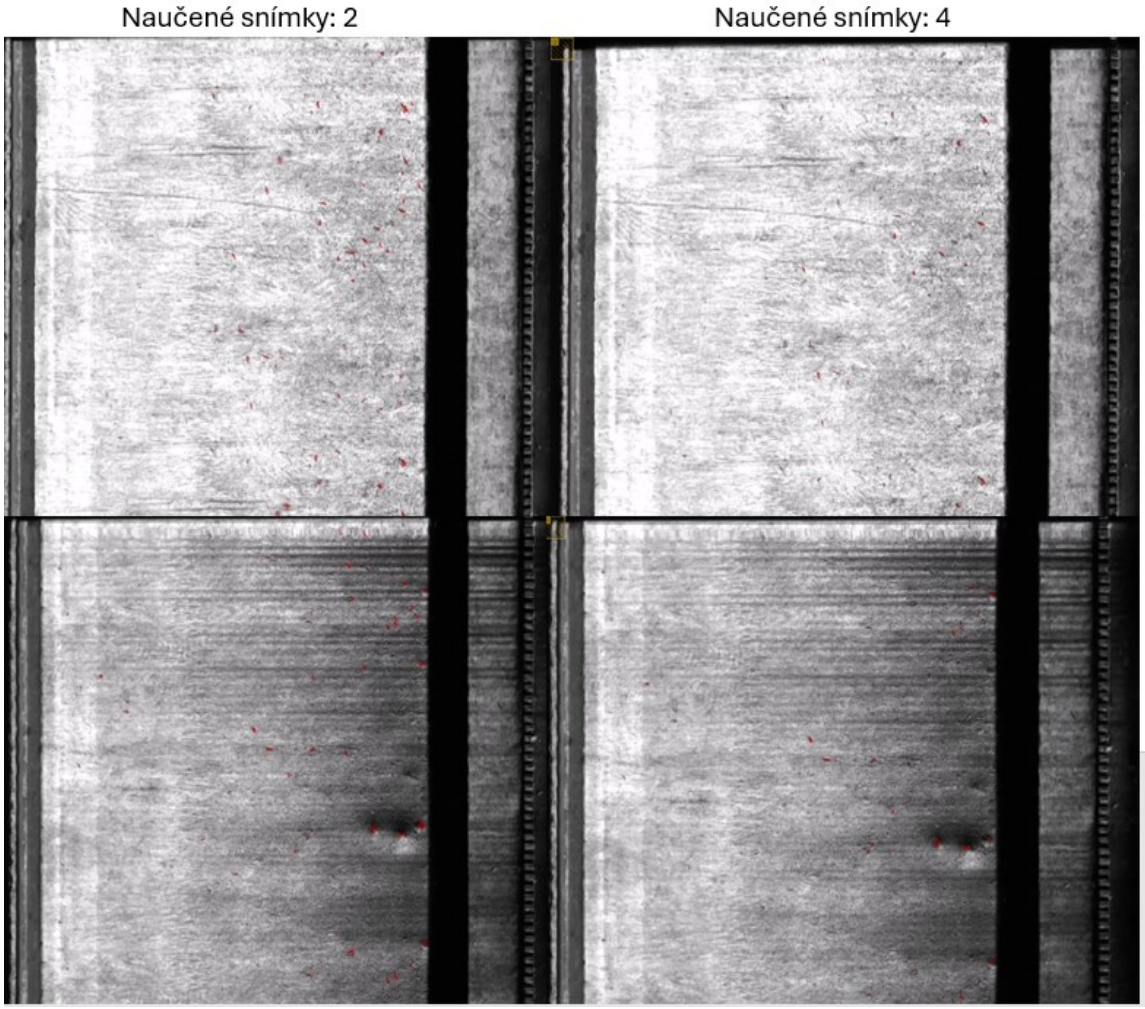
neuronové sítě Aurora Vision bylo již velmi blízko maximu a ani větší počet iterací by ho již příliš nezlepšil. Přestože se oba nástroje liší v přístupu k učení, výsledek, a tedy i výkon, bude podobný.



Obrázek 54. Křivka učení

Učení bylo vypnuto přibližně po stejné době, kterou se učil model v testu nástroje Cognex. Jelikož musí doběhnout poslední iterace, tak jsou časy učení jen přibližně podobné. Na grafu učení je vidět postupné zmenšování predikční chyby a přesnosti a také zmenšování chybové funkce, takže je možné, že kdyby se učení nechalo běžet déle, tak by se chyba ještě více zmenšila, ale ne nijak významně.

Výsledky jsou poměrně dobré. Již při dvou naučených snímcích byla neuronová síť schopna označit většinu vad na většině snímků. Při doučení dalších dvou snímků přestala označovat menší nevýznamné povrchové vady a začala se více zaměřovat na výraznější poškození povrchu, což je chování, kterého mělo být dosaženo. Výsledky označování vad jsou podobné jako u nástroje od firmy Cognex, akorát jsou zde označovány přesněji konkrétní vady, což by dávalo smysl vzhledem k požadavku na přesnější označování konkrétních vad při učení modelu neuronové sítě.

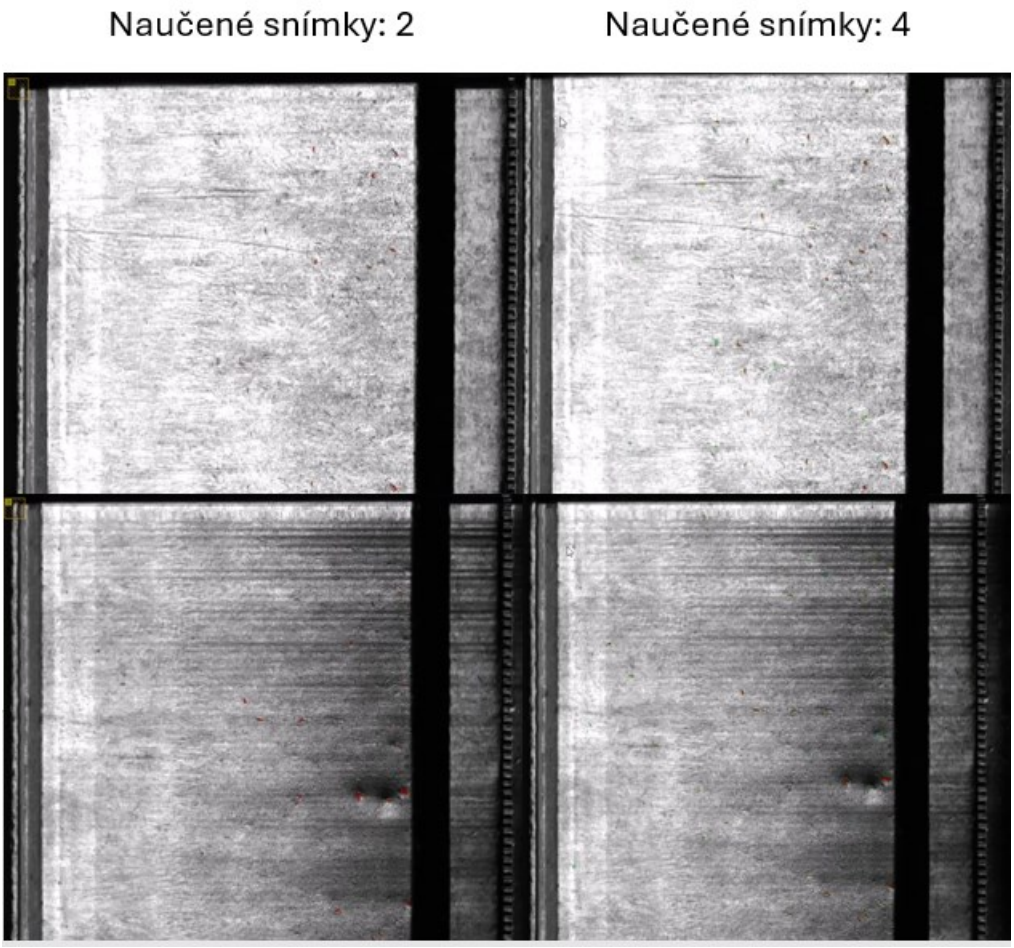


Obrázek 55. Ukázka výsledků pro 2 a 4 naučené snímky, nastavení 1

Nastavení 2:

Nastavení	Počet naučených vad	Doba učení modelu (s)	Počet Iterací	Ve skutečnosti OK.NOK
2	2	75	7	12.10
	4	75	7	

Tabulka 10.Výsledky nastavení 2



Obrázek 56. Ukázka výsledků pro 2 a 4 naučené snímky, nastavení 2

Pro nastavení 2 jsou výsledky velmi podobné jako u nastavení 1. Díky parametrům augmentations, které uměle zvýšily variace učených snímků, dokázala neuronová síť identifikovat již ze 2 snímků, že se nemá zaměřovat na nevýznamné povrchové vady. Takže u nastavení 2 bylo označeno méně nevýznamných vad než u nastavení 1, při stejném počtu naučených snímků.

7.2.2 Detect Anomalies

Nástroj Detect Anomalies funguje na principu učení bez učitele (unsupervised learning) popsaného v kapitole 1.2. Na rozdíl od nástroje od firmy Cognex je možné v tomto nástroji učit i na NOK snímcích. Avšak po několika testech bylo zjištěno že výsledky jsou mnohem konzistentnější s učení pouze OK snímků. Neuronová síť poté shlukuje OK snímky a ty co se liší (NOK) vyřadí.

Nastavení 1

Training Parameters

Name	Value
Network Type	SimilarityBased
Complexity	5
Device	NVIDIA GeForce RTX...
Pre-Processing	
Convert to Grayscale	True
Downsample	0
Augmentations	
Rotation Angle [°]	0.000
Flip Up-Down	False
Flip Left-Right	False
Minimum Scale [%]	100.000
Maximum Scale [%]	100.000
Shear Vertical Angle [°]	0.000
Shear Horizontal Angle [°]	0.000
Noise [%]	2.000
Luminance [%]	4.000
Gaussian Blur Kernel Size [...	0
Contrast [%]	0.000

Nastavení 2

Training Parameters

Name	Value
Network Type	SimilarityBased
Complexity	2
Device	NVIDIA GeForce RTX...
Pre-Processing	
Convert to Grayscale	True
Downsample	0
Augmentations	
Rotation Angle [°]	180.000
Flip Up-Down	True
Flip Left-Right	True
Minimum Scale [%]	75.000
Maximum Scale [%]	150.000
Shear Vertical Angle [°]	0.000
Shear Horizontal Angle [°]	0.000
Noise [%]	25.000
Luminance [%]	25.000
Gaussian Blur Kernel Size [...	0
Contrast [%]	0.000

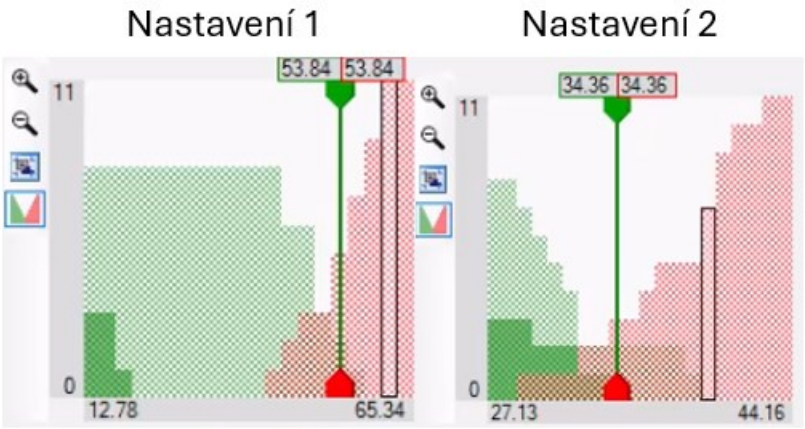
Obrázek 57. Nastavení

Nastavení se liší v komplexnosti neuronové sítě. Obecně se doporučuje větší komplexita pro složitější problémy a menší komplexita pro jednodušší problémy. V druhém nastavení také byly přidány parametry Augmentations.

Nastavení	Počet naučených snímků	Doba učení modelu (s)	Výstup modelu	Ve skutečnosti OK.NOK
1	3	123	OK: 14, NC:0, NOK: 8	11.11
2		85	OK: 11, NC:0, NOK: 11	

Tabulka 11. Výsledky Detect Anomalies

Nastavení s menší komplexitou má výrazně nižší čas učení a též poskytuje lepší výsledky. Vyšší komplexita se doporučuje na snímky, které by bylo složité např. rozlišovat pomocí lidského operátora. Použité snímky se dají rozeznat pomocí lidského operátora, takže komplexita odpovídá spíše menší úrovni, proto i výsledky jsou lepší s tímto nastavením.



Obrázek 58. Grafy predikce pro nastavení 1 a 2

Nastavení 1

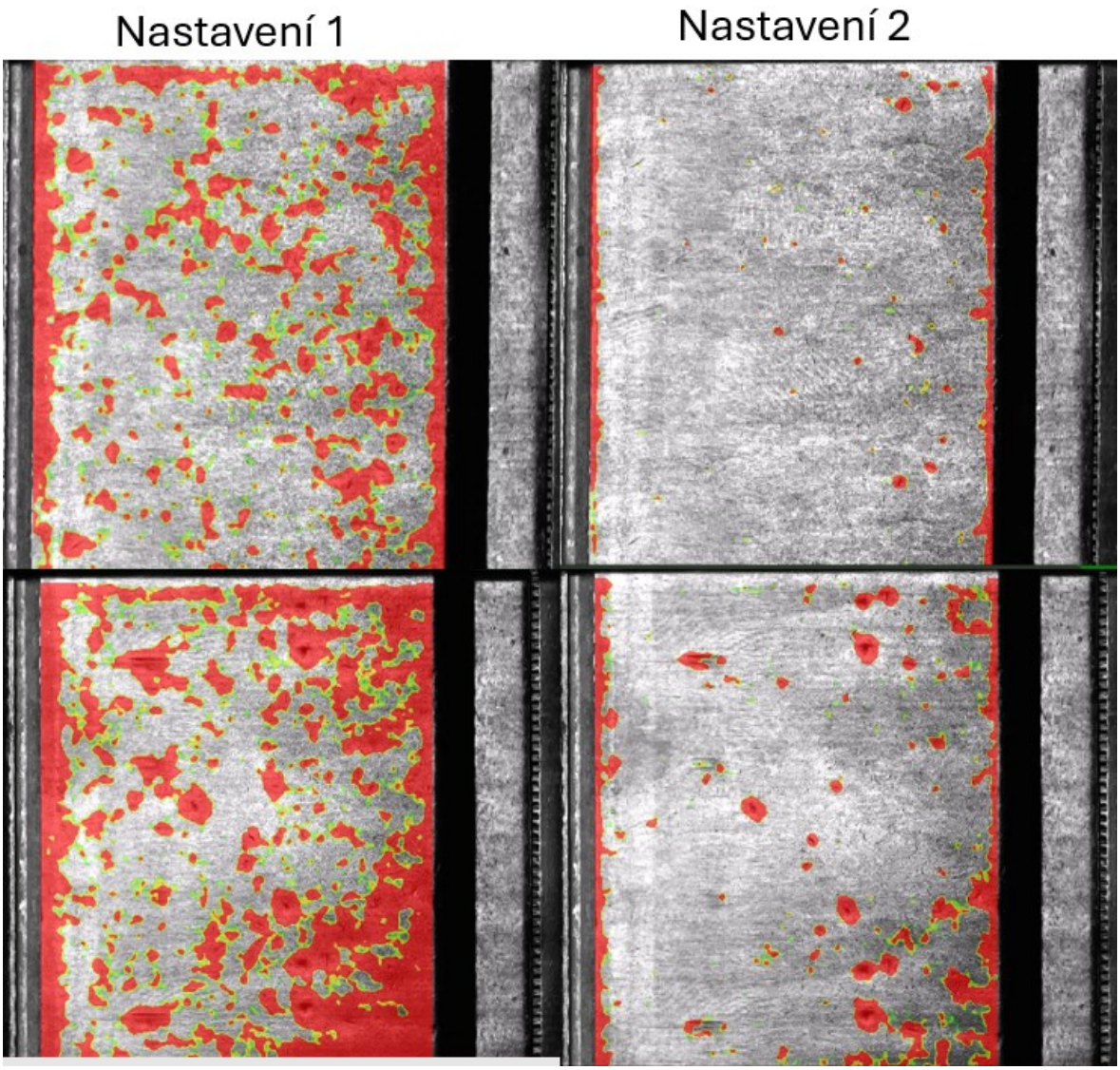
Predicted \ Actual	Good	Weak	Bad
Good	10	0	4
Weak	0	0	0
Bad	1	0	7

Nastavení 2

Predicted \ Actual	Good	Weak	Bad
Good	9	0	2
Weak	0	0	0
Bad	2	0	9

Obrázek 59. Matice zmatení pro nastavení 1 a 2

Jak je vidět z výsledků, tak nastavení 1 (vysoká komplexnost modelu) nebyla vhodná pro tuto aplikaci. Toto nastavení způsobilo, že neuronová síť na snímcích hledala různé skryté korelace, které v datech pravděpodobně nebyly, nebo byly velmi skryté. Přes to pomocí této logiky rozřadila většinu snímků správně. Nastavení 2 bylo úspěšnější a skutečně se zabývalo podstatnými daty a snímky roztřídila lépe, včetně sporných snímků



.

Obrázek 60. Příklad vyhodnocení OK a NOK snímku pro nastavení 1 a 2

Z výstupu neuronové sítě vyplývá že při nastavení velké komplexity sítě se model naučil třídit na základě korelací, které s detekcí vad pravděpodobně nesouvisely a shlukoval data na základě nějakých jiných pravidel. Jelikož se u této technologie téměř vždy jedná o black box, tak není možné určit co konkrétně v datech hledal. Přesto na základě těchto dat dokázal roztřídit většinu snímků správně. Nastavení 2, kde byla komplexita sítě natavena na odpovídající hodnotu této aplikaci, je již patrné, že model shlukuje správná data, tedy povrchové vady, z čehož se dá usoudit, že pochopil, o co primárně v této detekci jde. Přesto si z nějakého důvodu asocioval korelaci mezi OK snímkem a hranou náboje. Tento problém by se dal vyřešit například úpravou ROI. Přesto, že bral v potaz hranu, která s detekcí povrchové vady nesouvisela, tak tento model roztřídil valnou většinu snímků správně, včetně toho, že neurčité snímky ohodnotil hodnotami blížícím se středu.

7.3 Shrnutí výsledků měření

Všechny výsledky testů byly popsány v předchozích kapitolách. Tato kapitola se bude zabývat shrnutím výsledků a poznatků a jejich implikacemi.

7.3.1 Zlepšení s větším množstvím dat:

- Byla potvrzena premisa, že modely zlepšují své výsledky při poskytnutí většího množství učících dat.
- I minimální počet naučených snímků může vést k solidní přesnosti, ale zvýšení počtu snímků zvyšuje robustnost modelu.

7.3.2 Supervizované vs. nesupervizované učení:

- Metoda supervizovaného učení se ukázala jako vhodná pro úlohy, kde je hledaný defekt jasně definován. Pokud tomu tak není, je vhodnější použít metodu nesupervizovaného učení.
- Nesupervizované učení dokáže dosáhnout stejných výsledků jako supervizované, ale až s výrazně větším počtem naučených dat.
- U Aurora vision má největší vliv na výsledky i dobu učení nastavení komplexity modelu. Je důležité chápat, že větší hloubka neznamená nutně lepší výsledky. Proto by tento parametr měl být nastavován vždy podle složitosti konkrétní aplikace
- Supervizované a nesupervizované učení má přibližně stejnou dobu učení při zachování všech ostatních parametrů stejných.

7.3.3 Důležitost správných dat a parametrů:

- U obou nástrojů je extrémně důležité poskytovat učícímu modelu správná data. Například jedna špatně označená chyba u supervizovaného učení nebo jeden sporný snímek u nesupervizovaného učení může vést k výrazně horším výsledkům.
- Pro dobré výsledky je důležité správně nastavit Feature size a ROI.
- Nastavení sampling size má velký vliv na dobu učení.

- Zvýšení počtu epoch může mít pozitivní vliv na výsledky, ale je vždy lepší začít s menším počtem a postupně přidávat, aby se předešlo efektu přeučení.

7.3.4 Efekt přeučení:

- Efekt přeučení se obvykle projevuje tím, že model začne snímky roztřiďovat s menší jistotou, až nakonec bude většinu hodnotit velmi neutrálně.
- Při vysokém počtu epoch nebo příliš velkém množství dat může dojít k přeučení, což zhorší výsledky.

7.3.5 Čas učení:

- Na čas učení má velký vliv nastavení ROI; při větším ROI bude objem vstupních dat též větší, což prodlužuje dobu učení modelu.
- Pro optimální výsledky je nutné ROI a Feature size vždy upravit v závislosti na konkrétní aplikaci.
- Parametry perturbation/augmentation můžou mít poměrně velký vliv na dobu učení

7.3.6 Porovnání nástrojů Cognex a Aurora Vision:

- Cognex i Aurora Vision dokázaly správně rozřadit většinu snímků s pouhými 2 naučenými snímky, které reprezentovaly typickou vadu.
- Cognex i Aurora Vision poskytly za přibližně stejný učící čas srovnatelné výsledky s tím, že Aurora Vision označovala defekty o něco precizněji.
- Jednotlivé iterace v nástroji Aurora Vision trvaly déle než u nástroje Cognex Vision, ale změna přizpůsobení mezi jednotlivými iteracemi byla též větší, takže výsledky byly přibližně srovnatelné. Aurora Vision umožňovala navíc real time sledovat učící křivku a v případě potřeby učení ukončit dříve.
- Prostředí Cognex bylo uživatelsky jednodušší a snadněji pochopitelné. Aurora Vision naopak nabízí více možností, pokud jde o jednotlivé nástroje, specifická nastavení modelů učení a možnosti práce s výstupními daty.

7.3.7 Specifická nastavení pro nástroj Locate:

- U nástroje Locate vedlo zvýšení hustoty vzorkování k lepším výsledkům za cenu delší doby učení.
- Se změnou nastavení přesnosti u nástroje Locate musí být zacházeno opatrně, neboť vyšší nastavení této hodnoty vedlo k mnoha falešně pozitivním výsledkům. To samé platí s parametry perturbation; pokud variabilita hledaného objektu je menší než nastavené parametry perturbation, model hledá objekt i tam, kde ve skutečnosti není.
- Spousta chyb se dá vyřešit správným nastavením thresholdu.
- Je důležité vyladit hodnotu thresholdu pro konkrétní aplikaci tohoto nástroje.

ZÁVĚR

Tato bakalářská práce se komplexně zabývala principy fungování neuronových sítí a jejich praktickou aplikací v oblasti detekce vad.

V první části práce byly detailně rozebrány teoretické základy neuronových sítí a umělé inteligence obecně. Byla řešena témata jako nesupervizované a supervizované učení, optimalizace výstupní chyby pomocí regrese a gradientního sestupu a principy hlubokého učení. Práce se také věnovala neuronovým sítím z historického hlediska. Dále se věnovala popisu funkce neuronových sítí, principu umělého neuronu, aktivačním funkcím, adaptačním pravidlům a možným problémům, které se mohou vyskytnout při učení neuronových sítí. Tím byl splněn bod zadání číslo 1.

Druhá část prezentuje výsledky experimentů zaměřených na otestování výkonu a schopností vybraných nástrojů využívajících technologii neuronových sítí a hlubokého učení.

Nejdříve byly popsány dostupné softwary, se kterými byly později prováděny jmenované testy, ale byly popsány i další dostupné nástroje se stejným zaměřením. Tím byl splněn bod zadání 2.

V další kapitole byly popsány množiny snímků, na kterých byly testovány dostupné neuronové sítě. Kapitola se věnuje odůvodnění výběru a popisem sad. Tím byl splněn bod zadání 3.

V další části byla provedena samotné srovnání. Ta byla provedena na základě různých metrik a testovacích sad dat, čímž umožňují komplexní hodnocení efektivity a účinnosti jednotlivých nástrojů při řešení specifických úkolů spjatých s vizuální kontrolou snímků.

Na konci byly výsledky shrnuty a zhodnoceny, včetně různých postřehů a zjištění. Tím byly splněny body zadání 4 a 5.

Hlavními přínosy bakalářské práce jsou:

- Ucelený přehled základních principů neuronových sítích a jejich fungování
- Prezentace výsledků experimentů s neuronovými sítěmi pro detekci vad
- Cenný vhled do praktického využití neuronových sítí pro různé aplikace v průmyslu

SEZNAM POUŽITÉ LITERATURY

[1] What is a neural network? *IBM* [online]. 2021 [cit. 2024-02-19]. Dostupné z: https://www.ibm.com/topics/neural-networks

[2] Training Deep Neural Networks. *Towardsdatascience.com* [online]. 2018, 11.9.2018 [cit. 2024-02-19]. Dostupné z: https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964

[3] *Deep Learning A PRACTITIONER'S APPROACH* [PDF]. OREILLY [cit. 2024-02-21].

[4] PATTERSON, Josh a Adam GIBSON. *Deep Learning*. Printed in the United States of America: O'Reilly Media, Inc., August 2017:. ISBN 9781491914250.

[5] *Umělá inteligence Rozpoznávání vzorů v dynamických datech*. BEN, 2014. ISBN 9788073004972.

[6] Co je posilovací učení? *Unite.AI* [online]. 2021 [cit. 2024-05-02]. Dostupné z: https://www.unite.ai/cs/co-je-posilovac%C3%AD-u%C4%8Den%C3%AD/

[7] Overview of a Neural Network's Learning Process. *Medium.com* [online]. 2022 [cit. 2024-03-12]. Dostupné z: https://medium.com/data-science-365/overview-of-a-neural-networks-learning-process-61690a502fa

[8] *Linear Regression — Detailed View* [online]. 2018 [cit. 2024-04-29]. Dostupné z: https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86

[9] CHOLLET, François. *Deep learning v jazyku Python: knihovny Keras, Tensorflow*. 2., přepracované a rozšířené vydání. Praha: Grada Publishing, 2019. Knihovna programátora (Grada). ISBN 978-80-271-5133-2.

[10] The Essential Guide to Neural Network Architectures. *V7labs* [online]. 2021 [cit. 2024-05-22]. Dostupné z: https://www.v7labs.com/blog/neural-network-architectures-guide

[11] History: The 1940's to the 1970's. *Cs.stanford.edu* [online]. cca 2010 [cit. 2024-02-19]. Dostupné z: https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html

[12] History: The 1980's to the present. *Cs.stanford.edu* [online]. cca 2010 [cit. 2024-02-19]. Dostupné z: https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history2.html

[13] A Brief History of Neural Nets and Deep Learning. *Skynettoday* [online]. 2020 [cit. 2024-03-25]. Dostupné z: https://www.skynettoday.com/overviews/neural-net-history

[14] How to Convert an RGB Image to a Grayscale. In: *Baeldung* [online]. 2024 [cit. 2024-04-17]. Dostupné z: https://www.baeldung.com/cs/convert-rgb-to-grayscale

[15] *VISIONPRO DEEP LEARNING DataSheet* [online]. Cognex, 2023 [cit. 2024-04-08]. Dostupné z: https://www.cognex.com/downloads/visionpro-deep-learning-datasheet

[16] *Zebra* [online]. 2024 [cit. 2024-05-14]. Dostupné z: https://www.zebra.com/us/en/products/oem/software/aurora-deep-learning.html

[17] *Omron AI Visual Inspection System* [online]. 2020 [cit. 2024-05-22]. Dostupné z: https://www.omron.com/global/en/technology/omrontechnics/vol51/003.html

[18] *TensorFlow* [online]. 2020 [cit. 2024-05-22]. Dostupné z: https://www.tensorflow.org/

[19] *Keras* [online]. 2020 [cit. 2024-05-22]. Dostupné z: https://keras.io/

[20] *NVIDIA DeepStream* [online]. 2020 [cit. 2024-05-22]. Dostupné z: https://developer.nvidia.com/deepstream-sdk

[21] *Cognex Vidi* [online]. 2020 [cit. 2024-05-20]. Dostupné z: https://support.cognex.com/docs/vidi_410/web/EN/vidisuite/Content/ViDi-Topics/get-started/get-started.htm

[22] *Release Notes: Zebra Aurora Vision Studio™ 5.4* [online]. 2024.02.12 [cit. 2024-05-15]. Dostupné z: https://www.adaptive-vision.com/download/ReleaseNotes.html

[23] MSI GeForce RTX 3070 Ti GAMING X TRIO 8G, LHR, 8GB GDDR6X. *CzC* [online]. 2020 [cit. 2024-05-22]. Dostupné z: https://www.czc.cz/msi-geforce-rtx-3070-ti-gaming-x-trio-8g-lhr-8gb-gddr6x/321075/produkt

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AI – Artificial Intelligence (Umělá inteligence)

ANN – Artificial Neural Network (Umělá neuronová síť)

BERT – Bidirectional Encoder Representations from Transformers

CNN – Convolutional Neural Network (Konvoluční neuronová síť)

CPU – Central Processing Unit

DNN – Deep Neural Network (Hluboká neuronová síť)

DL – Deep Learning (Hluboké učení)

FFNN – Feedforward neural networks

GPU Graphics Processing Unit

GRU – Gated Recurrent Unit (Rekurentní jednotka s bránou)

GPT – Generative Pre-trained Transformer

ML – Machine Learning (Strojové učení)

NC-Not classified

NLP – Natural Language Processing

OCR – Optical Character Recognition

PC-Personal computer

RAM – Random Access Memory

ReLU – Rectified Linear Unit

ROI – Region of interest)

RNN – Recurrent Neural Network (Rekurentní neuronová síť

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: OBSAH CD

/fulltext – text práce

/testovaciSnimky – Sady snímků použitých v práci