



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**AUTOMATICKÁ ANALÝZA OBRAZU PRO KONTROLU
KVALITY VÝROBY TEXTILIÍ**

AUTOMATIC IMAGE ANALYSIS FOR PRODUCTION QUALITY CONTROL OF TEXTILE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TEREZA SÝKOROVÁ

VEDOUcí PRÁCE

SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2022

Zadání diplomové práce



Studentka: **Sýkorová Tereza, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Počítačové vidění
Název: **Automatická analýza obrazu pro kontrolu kvality výroby textilií**
Automatic Image Analysis for Production Quality Control of Textile
Kategorie: Zpracování obrazu
Zadání:

1. Prostudujte existující postupy a literaturu na téma automatická kontrola kvality výroby prostřednictvím automatické analýzy obrazu. Zaměřte se zejména na výskyt vad v textiliích.
2. Vyberte vhodnou metodu pro klasifikaci typu vad v textiliích (otvory, vady textilie/vláken textilie, objekty zachycené v textiliích, například hmyz apod.)
3. Navrhněte způsob implementace vybrané metody a diskutujte dosažitelné výsledky a diskutujte její vlastnosti .
4. Implementujte vybranou klasifikační metodu a demonstруйте funkčnost na vhodném příkladu.
5. Zhodnořte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 18. května 2022

Datum schválení: 3. listopadu 2021

Abstrakt

Tato práce se zabývá problematikou klasifikace defektů, které vznikají při výrobě netkaných textilií. Úloha klasifikace vad je součástí systému pro automatickou kontrolu kvality výroby. Cílem je implementovat metodu, která bude co nejlépe klasifikovat problematické třídy defektů. Toho bylo dosaženo s využitím konvolučních neuronových sítí (CNN). Nejlepší výsledky měla síť EfficientNet, která dosáhla přesnosti 81% při vyhodnocení metodou křížové validace na dostupném datasetu. V rámci práce je provedena řada experimentů, které jsou zaměřeny na úpravu vstupních dat. Zkoumán je vliv tvaru a složení vstupních snímků na výslednou klasifikaci. Implementován byl také model CNN, který kromě obrázku využívá pro klasifikaci také další informace.

Abstract

This work deals with the classification of defects that occur in the production of nonwovens. The defect classification task is part of a system for automatic production quality control. The goal is to implement a method that will classify problematic defect classes with sufficient accuracy. That was achieved using convolutional neural networks (CNN). The best results were achieved by the EfficientNet network, which had an accuracy of 81 % when evaluated by cross-validation on an available dataset. Within the work, a number of experiments are performed, which are focused on the modification of input data. The influence of the shape and composition of the input images on the final classification is examined. A CNN model was also implemented, which uses additional information for classification in addition to the image.

Klíčová slova

klasifikace vad, netkané textilie, kontrola kvality, konvoluční neuronové sítě, zpracování obrazu

Keywords

defect classification, nonwoven fabric, quality control, convolutional neural network, image processing

Citace

SÝKOROVÁ, Tereza. *Automatická analýza obrazu pro kontrolu kvality výroby textilií*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Dr. Ing. Pavel Zemčík

Automatická analýza obrazu pro kontrolu kvality výroby textilií

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením prof. Dr. Ing. Pavla Zemčíka. Další informace mi poskytl Ing. Peter Honec, Ph.D. z firmy CAMEA, spol. s r. o. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....

Tereza Sýkorová

17. května 2022

Obsah

1	Úvod	2
2	Kontrola kvality výroby	3
2.1	Vady netkaných textilií	3
2.2	Kontrolní systém	4
2.3	Popis referenčního řešení	5
2.4	Metody automatické analýzy obrazu	6
2.5	Existující přístupy	6
3	Umělé neuronové sítě	10
3.1	Neuron	10
3.2	Dopředné neuronové sítě	10
3.3	Učení neuronové sítě	11
3.4	Konvoluční neuronové sítě	15
3.5	Architektury CNN	17
3.6	Knihovny pro počítačové vidění	20
4	Návrh řešení	22
4.1	Analýza současného stavu	22
4.2	Popis dat	23
4.3	Návrh řešení	25
4.4	Způsob vyhodnocení	26
5	Implementace řešení a vyhodnocení	28
5.1	Předzpracování dat	29
5.2	Rozměry obrazu	29
5.3	Složení trénovacích dat	31
5.4	Augmentace	31
5.5	Modely CNN	32
5.6	Vyhodnocení experimentů	33
6	Závěr	39
	Literatura	40

Kapitola 1

Úvod

Kontrola kvality produktů je důležitou součástí každého výrobního procesu. I přes to, že jsou dnes dostupné pokročilé technologie výroby, nelze zcela zabránit poruchám výrobního procesu. Výjimkou není ani textilní průmysl. Netkané textilie mají široké uplatnění v mnoha oblastech každodenního života, ať už se jedná o dětské pleny, hygienické potřeby, nábytkové potahy, nebo izolace. Kontrola materiálu je tak důležitá především proto, aby se vadný produkt nedostal k zákazníkovi. Cílem této práce je vytvořit klasifikátor, který bude spolehlivě rozlišovat defekty, které při výrobě netkaných textilií vznikají.

Dříve byla kontrola výroby prováděna především manuálně, trénovanými zaměstnanci, a i dnes vyžaduje do jisté míry lidský dohled. Tento způsob je však časově náročný a výsledky se mohou lišit podle toho, jak daný pracovník vadu subjektivně posoudí. Kvůli stoupající únavě pak může docházet k přehlédnutí některých defektů. Díky vývoji techniky i algoritmů pro zpracování obrazu je dnes možné kontrolu výroby provádět automatizovaně v reálném čase. Ačkoli došlo v této oblasti ke značnému pokroku, stále je prostor pro vývoj přesnějších a spolehlivějších metod.

Mezi hlavní úkoly při kontrole kvality patří jak detekce a lokalizace defektu, tak jeho klasifikace. Určení správné kategorie nalezené vady je zásadní především proto, aby bylo možné rozhodnout, co s vadným kusem materiálu provést. Lehké poškození může být přehlédnuto, závažné defekty však nemohou v produktu zůstat a vadný kus je zapotřebí odstranit. Na základě statistických informací o tom, kolik konkrétních vad vzniká, firma může provést kroky k omezení jejich vzniku a minimalizovat tak množství odpadního materiálu. Dojde ke zlepšení kvality produktů i snížení výrobních nákladů.

V následující kapitole je popsán způsob výroby a snímání vad. Zmíněny jsou také některé druhy defektů a jak vznikají. Dále jsou zde popsány metody zpracování obrazu pro detekci a klasifikaci. Uveden je přehled dosud existujících aplikací některých metod. Zvláštní kapitola 3 je věnována neuronovým sítím, které jsou v posledních letech stále více využívány a mají velmi dobré výsledky při klasifikaci. Zvláště jsou popsány konvoluční sítě. Dále je v kapitole 4 popsána datová sada spolu s návrhem řešení. V poslední kapitole je popis jednotlivých experimentů a jejich vyhodnocení.

Kapitola 2

Kontrola kvality výroby

Detekce vad vznikajících při výrobě různých materiálů je důležitou součástí kontroly kvality výrobků. Dříve byl tento úkol prováděn vyškolenými pracovníky především manuálně. Tento způsob má však značné nevýhody a zpomaluje celkovou výrobu. Posouzení kvality je subjektivní, protože závisí na rozhodnutí konkrétního pracovníka, a během času klesá pozornost a stoupá celková únava. Následkem toho může docházet k přehlédnutí některých vad.

Automatizace toho procesu je klíčová pro zlepšení kvality a snížení celkové ceny výrobků. Díky rychlému rozvoji v oblasti počítačového vidění je možné používat kvalitní kamery, výkonné počítače a spolehlivé algoritmy pro zpracování a vyhodnocení obrazu, které rychle a přesně detekují a klasifikují vady na výrobcích. První pokusy byly založeny například na prahování, morfologických operacích nebo spektrální analýze. Dnes se stále více rozšiřuje použití umělé inteligence, především konvolučních neuronových sítí.

V této kapitole je popsán úvod do problematiky výroby netkaných textilií a poškození, ke kterým dochází. Kapitola popisuje také způsob kontroly kvality pomocí kamer a referenční řešení, které je v současnosti používáno firmou CAMEA s r. o. Na závěr jsou uvedeny metody automatické analýzy obrazu bezprostředně související s touto prací a několik konkrétních řešení detekce a klasifikace defektů. Nejedná se o encyklopedický přehled, protože rozsah práce je omezen.

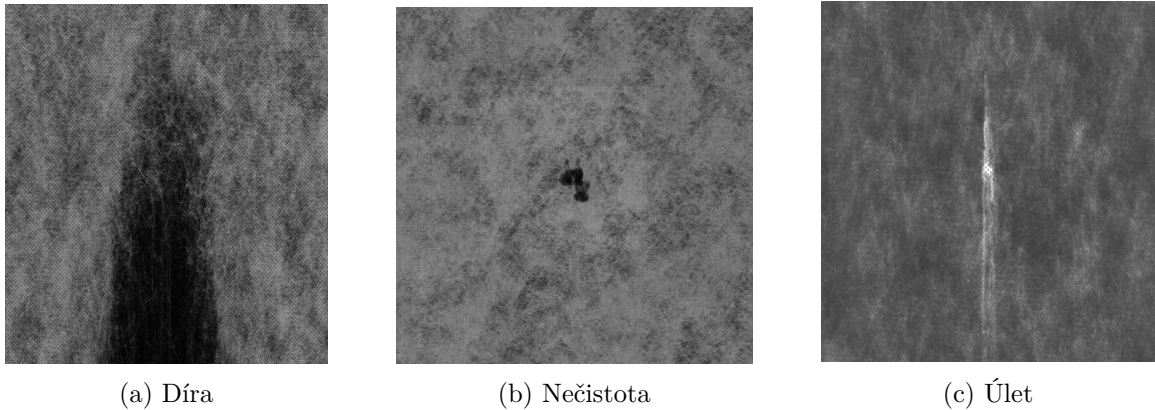
2.1 Vady netkaných textilií

Netkané textilie jsou vyráběny v podobě širokých pásů dlouhých stovky metrů a během toho jsou kontrolovány různými kamerami. Ačkoli jsou dnes dostupné kvalitní technologie výroby, nelze s jistotou zabránit vzniku různých defektů. Jedná se o určitým způsobem poškozené oblasti, které narušují plynulost a rovnoměrnost textury. Tyto vady mají za následek zhoršení kvality výsledných produktů, mezi které patří například hygienické prostředky, utěrky, ochranné oděvy, chirurgické pláště a masky, nábytkové potahy, geotextilie, nebo izolace [4].

Pro efektivní výrobu je důležitá jak detekce vad, tak jejich klasifikace. Typy defektů lze rozdělit do dvou kategorií na závažné a méně závažné [48]. Při výrobě může dojít například k utržení vlákna, skladu, vznikají zesílená nebo zeslabená místa, díry, nebo se do výroby může připlést hmyz. Ukázka některých vad je na obrázku 2.1. Pokud se v látce vyskytne některý z vážných defektů, je zapotřebí vadu lokalizovat a odstranit část vadné role textilie. Dochází tak ke ztrátám materiálu i času. Klasifikaci lze také využít pro analýzu a vylepšení

systému. Díky určení, jaké vady při výrobě nejčastěji vznikají, můžeme podniknout kroky k jejich minimalizaci.

Zvláštní pozornost si zaslouží dvě vizuálně podobné kategorie vad, a to tzv. úkap a úlet. Úlet vzniká nahromaděním prachových částic na nějakém povrchu, které se uvolní a přichytí na textilií. Tato vada je sice viditelná, ale ve výsledném produktu nemusí způsobit velkou škodu, protože má podobnou strukturu jako samotná textilie. Na druhou stranu úkap představuje mnohem závažnější problém. Ten vzniká během výroby nahromaděním výrobního materiálu na jednom místě. Nejen, že je úkap na povrchu viditelný, ale zároveň tvoří v textilií hrubá, tvrdá místa. Pokud se taková vada dostane do výsledného produktu (např. dětské pleny), výrazně snižuje kvalitu výrobku. Proto je velmi důležité tyto vady detekovat a rozlišit při výrobě, aby se vadné výrobky nedostaly k zákazníkovi.



Obrázek 2.1: Ukázky vad vznikajících při výrobě netkaných textilií.

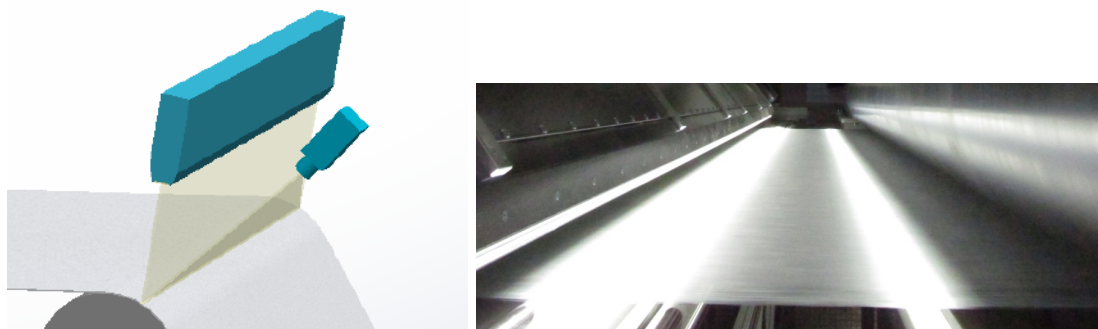
2.2 Kontrolní systém

Informace pro tuto sekci jsou čerpány z [44]. Systém pro detekci vad se skládá z několika vysokorychlostních kamer, osvětlení a řídicího systému, který provádí zpracování obrazových dat. Pásky široké až 5 metrů běží rychlostí až 2000 m/min přes válce a během toho jsou snímány různými kamerami, které jsou schopny zachytit vady už od velikosti 0.1 mm². Ukázka části systému je na obrázku 2.2.

Při snímání je důležité kvalitní osvětlení. Využíváno je více druhů LED světél, které jsou nastaveny do různých úhlů v kombinaci s kamerami. Pro zvýraznění různých typů vad je možné použít světla odlišných vlnových délek nebo sklonu.

Aby bylo možné provádět detailní kontrolu vysokou rychlostí, systém kombinuje nejmodernější řádkové kamery se snímači s vysokým rozlišením. U kamery je možné dálkově nastavovat clonu objektivu a zaostření, nebo upravovat jejich vzdálenost od povrchu. Různé kombinace kamer a osvětlení umožňují detekovat různé typy vad.

Data z kamer jsou přenášena do řídicí jednotky, která je zpracovává a vyhodnocuje. Vady jsou detekovány algoritmem, který kompenzuje nerovnoměrné osvětlení. Snímky jsou zpracovávány v reálném čase a vizualizovány, aby poskytovaly informaci o stavu kvality pásu. Dohlížející operátor tak může okamžitě zareagovat na vznikající vady.



Obrázek 2.2: Ukázka části systému pro kontrolu výroby netkaných textilií [44].

2.3 Popis referenčního řešení

Pro klasifikaci defektů byly ve firmě CAMEA s r. o. testovány dva přístupy [49]. První je metoda podpurných vektorů (SVM) pojmenovaná jako metoda CL1. Jako druhý klasifikátor byl vyzkoušen rozhodovací strom. Jeho pravidla byla vytvořena pomocí software Rapid-Miner (metoda CL2). U obou metod je ke klasifikaci využíván příznakový vektor, který je určen jednorůchodovým algoritmem v průběhu rychlé detekce vad, pak již nedochází k další analýze obrazu. Mezi využívané hodnoty patří například jasové hodnoty uvnitř a vně vady, percentilové jasové hodnoty, síla hran. Bylo však zjištěno, že tyto příznaky nejsou vhodné k dostatečně spolehlivé klasifikaci defektů.

Lze konstatovat, že průměrné úspěšnosti obou klasifikátorů jsou velmi podobné a to 68,1% pro CL1 a 65,6% pro CL2. V současnosti je využíván klasifikátor CL1. Úspěšnost byla počítána na testovacím souboru o velikosti 1069 snímků pro skupinu 12 jako podíl správně klasifikovaných vad a celkového počtu snímků. Výsledky klasifikátoru CL1 pro jednotlivé vady jsou uvedeny v tabulce 2.1.

Kategorie vady	úspěšnost v %	celkový počet snímků
úkap	57,3	75
úkap s dírou	17,9	28
úlet	51,3	80
utržené vlákno	43,5	138
vlaštovka	51,6	124
zesílené místo	63,2	174
sklad	69,2	13
prášení	97,0	397
dlouhý tenký úlet	27,5	40

Tabulka 2.1: Úspěšnosti (recall) klasifikátoru CL1 pro jednotlivé vady skupiny 12 (nadsvit).

Jeden z problémů při klasifikaci vzniká u třídy *utržené vlákno*, která je velmi často zařazena jako *úkap*, *prášení*, *dlouhý tenký úlet*. Při zkoumání oněch snímků bylo zjištěno, že nejsou dostatečně dlouhé a proto skončily v jiné kategorii vyhrazené pro krátké defekty (*prášení*). Stejně je tomu i v případě třídy *prášení*, i ta je dost často zaměňována za *zesílené místo* či *utržené vlákno*. Třída *úkap* je svými hodnotami také nevyhovující a dost často zaměňována za téměř všechny ostatní třídy. Problémem byla také přeučenost klasifikátoru a neschopnost tak správně reagovat na neznámá testovací data.

2.4 Metody automatické analýzy obrazu

Automatickou kontrolu kvality výroby lze rozdělit na tři části podle toho, co požadujeme jako výsledek [5]. Typicky prvním krokem při vyhodnocování kvality výrobku je určení, zda se na něm nachází nějaký defekt nebo ne, a zaznamenání poškozené části. Následně je možné provést segmentaci obrazu tak, abychom určili přesnou pozici a velikost plochy, kterou poškození pokrývá. Tyto informace lze také použít pro klasifikaci vady. Rozhodnutí, o jakou vadu se jedná, a její zařazení do některé z definovaných kategorií může přinést užitečné informace o výrobním procesu.

Existují různé přístupy a metody jak pro detekci, tak pro klasifikaci vad na povrchu výrobků. Tyto metody můžeme rozdělit do několika kategorií. Nejčastější dělení je na metody statistické, spektrální a metody založené na modelech [22, 32]. Někteří zvláště rozlišují ještě kategorie strukturální, metody využívající učení, nebo hybridní metody, které kombinují více přístupů [5, 12, 27].

Statistické metody [33] patří mezi základní způsoby jak popsat texturu obrazu. Statistiky prvního řádu, mezi které řadíme průměrnou hodnotu, rozptyl, směrodatnou odchylku nebo histogram obrazu, umožňují získat informace o rozložení intenzit pixelů v dané textuře. Statistiky druhého řádu pomáhají určit vztah vybraného pixelu k pixelům v jeho určitém okolí. Do této skupiny lze zařadit metody:

- lokální binární vzor (Local Binary Pattern - LBP)
- koincidenční matice stupňů šedi (GrayLevel Co-occurrence Matrices - GLCM)
- autokorelační funkce
- matematická morfologie

Mnoho prací zabývajících se detekcí vad v textiliích využívá metody pro spektrální analýzu obrazu. Obraz je transformován do frekvenční domény, která umožňuje rozlišit defekty v periodické struktuře tkaniny. Tyto metody proto nejsou vhodné pro materiály, které mají náhodnou texturu. Nejčastěji využívané jsou Gaborova, Fourierova a vlnková transformace [5, 12, 27].

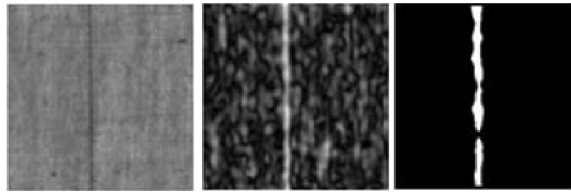
Mezi metody, které jsou založené na vytvoření modelu textury, můžeme zařadit autogresní modely nebo Markovská náhodná pole (MRF). Metoda GMRF byla využita například v práci Cohen a spol. [7] pro vytvoření modelu textury tkaniny bez poškození. Hajimowlana a spol. [11] aplikovali 1D autoregresivní model pro detekci defektů v systému kontroly výroby.

V posledních letech se pro kontrolu kvality výroby čím dál více rozšiřuje využití umělých neuronových sítí. Ty mohou být použity jak pro detekci vad, tak pro segmentaci a jejich klasifikaci. Existují různé varianty neuronových sítí a často jsou používány v kombinaci s dříve uvedenými metodami pro extrakci příznaků. Mezi další algoritmy učení patří také metoda podpůrných vektorů (support vector machines - SVM) nebo Bayesův klasifikátor [12].

2.5 Existující přístupy

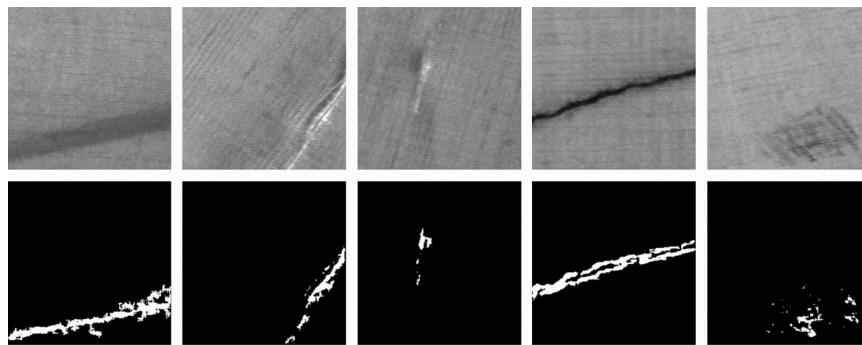
Mezi často používané metody pro detekci vad v textiliích patří algoritmy využívající spektrální analýzu. Zhang a spol. [46] ve své práci použili pro detekci defektů Gaborovy filtry. Šedotónový výstup filtrů byl převeden na binární obraz a díky tomu získali masku, kde bílé pixely odpovídají pozici vady, viz 2.3. Z této oblasti byly následně vypočteny různé

příznaky (průměr, odchylka), které sloužily k trénování GMM (Gaussian mixture models) klasifikátoru. Klasifikace 9 různých tříd dosahovala v průměru úspěšnosti 85%.



Obrázek 2.3: Zleva ukázka defektu, výsledek po použití Gaborových filtrů a binární maska [46].

Tong a spol. [43] dosahuje segmentace vad pomocí Gaborových filtrů, jejichž parametry jsou optimalizovány kompozitní diferencíální evolucí (composite differential evolution - CoDE). Tato metoda má lepší výsledky detekce než například upravený Gaborův filtr od Mak a spol. [20], nebo práce Hu a spol. [14], která optimalizuje parametry filtru pomocí simulovaného žhání. Jing a spol. [18] také používá Gaborovy filtry a prahování pro segmentaci vad. Na rozdíl od předchozích prací provádí také klasifikaci defektů do 5 tříd viz 2.4. Data, která byla použita pro trénování a vyhodnocení, obsahují 45 snímků pro každou třídu. Příznakový vektor pro učení neuronové sítě je složen z hodnot získaných metodou LBP [29] a z 6 příznaků pro textury definovaných Tamura metodou [40].



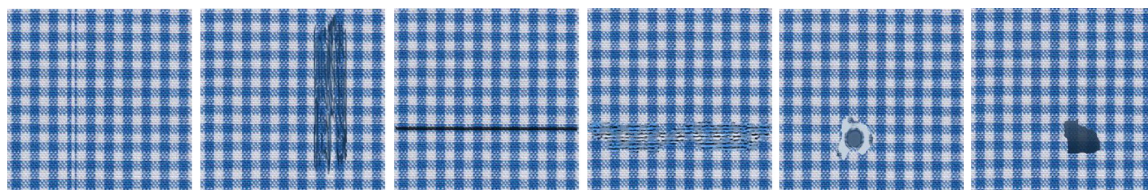
Obrázek 2.4: Ukázka 5 různých kategorií defektů ke klasifikaci z práce [18].

Mezi metody spektrální analýzy patří také rychlá Fourierova transformace (FFT), kterou spolu se statistickými metodami použil Shady a spol. [35] pro extrakci příznaků. Klasifikováno bylo 6 kategorií vad pletených látek s četností 30 obrázků. Neuronová síť, která měla jako vstup příznaky získané FFT, měla o něco lepší výsledky (přesnost 91%), než síť učená na základě statistických příznaků (89%). Çelik a spol. [48] dosáhl detekce vad pomocí vlnkové transformace, prahování a použitím morfologických operací. Příznaky získané vlnkovými transformacemi a metodou GLCM jsou použity jako vstup dopředné neuronové sítě. Čtyři typy vad byly klasifikovány s přesností přes 96%.

Sakhare a spol. [34] ve své práci porovnávali detekci defektů pomocí transformace do frekvenční domény se statistickými a morfologickými metodami. Databáze obsahovala 4 druhy vad. Z použitých postupů měla rychlá Fourierova transformace nejvyšší úspěšnost v detekci oblasti defektu (87,9%). Na druhém místě vyšla diskretní kosinová transformace (85,1%) následována Gaborovými filtry (81,5%) a vlnkovou transformací (76,5%). Nejhorší výsledky měl přístup využívající morfologické operace (52,44%).

Güler a spol. [9] porovnává tři různé přístupy pro extrakci příznaků založené na spektrální analýze obrazu. Jsou to rychlá Furierova transformace, vlnková transformace a Gaborovy filtry, které jsou aplikované na 25 snímků textilií. Příznaky z každé transformace byly použity pro učení neuronové sítě pro klasifikaci 3 různých defektů. Nejlepšího výsledku dosahovala vlnková transformace (96%) následovaná Gaborovými filtry (84%).

Jedním z používaných algoritmů umělé inteligence pro klasifikaci je metoda SVM. Využil ji například Li a spol. [24], který pro trénování SVM použil geometrické (výška, šířka, plocha) a texturové (kontrast) vlastnosti defektů. Klasifikace defektů barvených tkanin do 6 tříd (viz obrázek 2.5) dosahovala v průměru úspěšnosti 87,8%. Metoda byla porovnána s klasifikací pomocí neuronové sítě, která měla úspěšnost 85,9%.



Obrázek 2.5: Ukázky z jednotlivých tříd klasifikovaných defektů z [24].

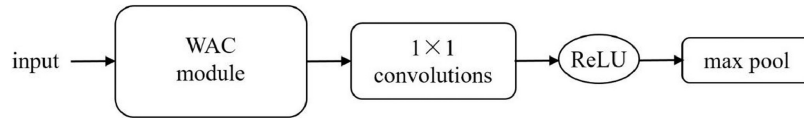
V posledních letech se rozšiřuje použití konvolučních neuronových sítí (CNN) pro detekci a klasifikaci objektů v obraze. Jejich výhodou je, že jsou schopné samy se naučit rozeznat vhodné příznaky bez toho, aby byly předtím ručně extrahovány jiným algoritmem. Nevýhodou může být větší časová a paměťová náročnost. Z přístupů, které jsou založeny na CNN, lze zmínit například práci Wei a spol. [45], který využívá komprimované snímání (compressive sensing) pro kompresi a augmentaci dat. Takto upravená data jsou použita jako vstup do konvoluční neuronové sítě, která klasifikuje vady do 10 tříd. Síť je složena ze dvou konvolučních a dvou pooling vrstev, na které navazují tři plně propojené vrstvy. Metoda dosahuje úspěšnosti klasifikace 97,9%.

Calderisi a spol. [3] porovnávali jednoduchý model CNN se složitějšími architekturami jako jsou Inception v3, DenseNet 201, MobileNet a ResNet 50. Složitější modely byly úspěšnější, ale měly také vyšší výpočetní nároky. Jejich cílem tedy bylo vylepšit jednoduchý model využitím metadat, která měli k dispozici o každém snímku. Výslednou síť, která klasifikovala vzorky do 2 kategorií vad, složili ze dvou částí. První je jednoduchá konvoluční síť, která má na vstupu snímky vad, a druhá je plně propojená síť, která zpracovává metadata. Oba výstupy jsou spojeny a použity jako vstup do výsledné plně propojené sítě. Tento kombinovaný model má podobnou úspěšnost (okolo 88%) jako složitější architektury, ale je méně náročný na paměť a čas.

Zhao a spol. [47] navrhli model konvoluční neuronové sítě založený na vizuální dlouhodobé a krátkodobé paměti (visual long-short-term memory - VLSTM) pro zlepšení klasifikace defektů textilií. Model je složený ze tří částí, které pracují paralelně, a má napodobit lidský zrakový systém. Každá z částí byla vyhodnocena zvlášť i ve spojení s ostatními. Dohromady však kombinovaný model nazvaný VLSTM dosahoval nejlepší úspěšnosti 95,73%. Model VGG [36] pro porovnání dosahoval na stejném datasetu úspěšnosti 91,64%.

Jing a spol. [17] vytvořili model CNN nazvaný Mobile-Unet pro segmentaci vad v textiliích. V první fázi dochází k trénování sítě MobileNetV2, která provádí extrakci příznaků a následnou klasifikaci vad do několika tříd. Parametry sítě jsou potom použity v druhé fázi, kde na konvoluční vrstvy navazuje dekonvoluce, a jejímž výstupem je segmentovaný defekt.

Li a spol. [25] navrhli architekturu WACNet (wide-and-compact network) pro klasifikaci 4 základních kategorií vad. Model využívá několik mikroarchitektur viz 2.6. V porovnání s běžně používanými modely sítí dosáhla WACNet vysoké úspěšnosti s mnohem menším modelem.



Obrázek 2.6: Nákres WAC mikroarchitektury [25].

Mottalib a spol. [26] využívá prahování pro detekci defektů a Bayesův klasifikátor pro jejich klasifikaci. Vstupní snímek tkané textilie s vadou je převeden do odstínů šedi a následně jsou použity filtry pro vyhlazení a odstranění šumu. Pomocí histogramu jsou určeny hodnoty pro prahování a obraz je převeden do binární podoby. Z toho výsledku jsou spočítány geometrické příznaky defektu a na jejich základě je natrénován Bayesův klasifikátor. Defekty jsou klasifikovány do 6 tříd, které zahrnují i snímky bez vad. Průměrná úspěšnost na testovací sadě o 58 snímcích je 99,19%.

Většina ze zmíněných přístupů pracuje převážně s pletenými textiliemi a malým počtem druhů vad pro detekci nebo klasifikaci. V některých případech je pro učení algoritmů a vyhodnocení úspěšnosti použita malá datová sada takže výsledky mohou být zkreslené.

Kapitola 3

Umělé neuronové sítě

Umělé neuronové sítě patří mezi metody strojového učení. Jejich využití v mnoha oblastech stále roste a často svou úspěšností převyšují dosud používané algoritmy. V počítačovém vidění mají využití především pro detekci, segmentaci a klasifikaci objektů v obraze.

V úvodu kapitoly je stručně popsáno, co stálo za vznikem neuronových sítí. Dále je text zaměřen na strukturu dopředných sítí a jejich učení. Samostatná podkapitola je věnována konvolučním neuronovým sítím. V této kapitole jsou uvedeny informace bezprostředně související s touto prací. Nejedná se o detailní popis problematiky umělých neuronových sítí, protože rozsah práce je omezen.

3.1 Neuron

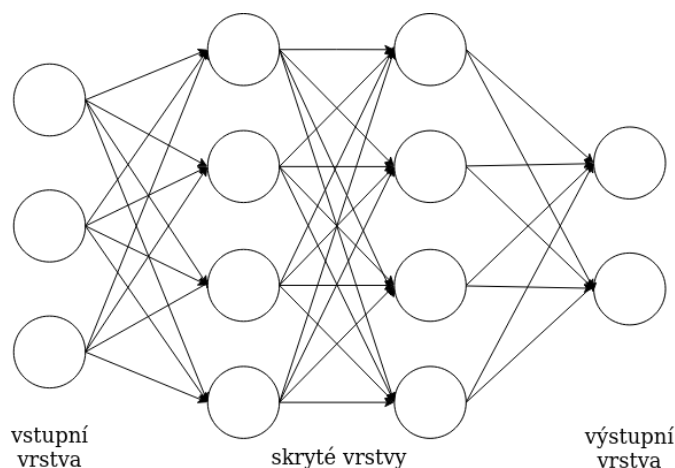
Vývoj umělých neuronových sítí byl inspirován biologickým nervovým systémem, jehož základní jednotkou je neuron. Zjednodušeně se neurony skládají ze tří základních částí: dendrity, jádro a axon. Mezi jednotlivými nervovými buňkami jsou spojení zvané synapse, které umožňují přenos vzruchu. Každý neuron má větší počet dendritů, které přijímají signály z okolních buněk. Vstupní signály ze všech dendritů jsou přenášeny do jádra, kde dochází k jejich zpracování. Pokud výsledek zpracování přesáhne určitý práh, dojde k vyslání výstupního signálu. Rozvětvený axon pak slouží k předávání výstupního signálu přes synapse na dendrity dalších nervových buněk [23].

Počítačová reprezentace neuronu pak vypadá následovně. Synapse jsou modelovány jako váhy (w_i), kterými jsou násobeny odpovídající vstupní signály (x_i) před tím, než jsou předány do pomyslného jádra. Tyto váhy jsou během učení neuronové sítě určitým způsobem měněny a díky tomu mohou rozhodovat, jak velký vliv má konkrétní vstupní signál na výstup. V jádře jsou signály ($w_i \cdot x_i$) sečteny dohromady a výsledek je vstupem aktivační funkce neuronu. Aktivační funkce může mít různou podobu a rozhoduje o výstupu, který je předán dalšímu neuronu [10, 23].

3.2 Dopředné neuronové sítě

Spojením více neuronů vzniká neuronová síť. Architektura sítě je složena z několika propojených vrstev, které udávají její hloubku. Vrstvy se skládají z určitého počtu neuronů, které pracují paralelně ale nejsou mezi sebou navzájem propojeny. Každý neuron přijímá na svých vstupech data z předchozí vrstvy a na výstup posílá výsledek ze své aktivační funkce. Hodnoty mezi vrstvami jsou typicky předávány ve formě vektoru. První a poslední

vrstva udávají, jaký formát dat je síť schopná přijímat na vstupu, a jaký je její výstup. Například pro úlohu klasifikace počet neuronů ve výstupní vrstvě odpovídá počtu tříd, do kterých klasifikujeme. Mezi vstupní a výstupní vrstvou je několik skrytých, plně propojených vrstev, které provádí samotné učení. U plně propojené vrstvy je výstup z každého neuronu napojený na vstup všech neuronů vrstvy následující. Dopřednou nazýváme takovou síť, u které informace putuje od vstupu, přes vnitřní výpočty na výstup. Nejsou zde žádné zpětné vazby, které by výstup propojovaly zpátky s modelem. Pokud takové propojení existuje, síť se nazývá rekurentní. Cílem dopředné sítě je aproximovat nějakou funkci pomocí parametrů tak, aby nejlépe řešila daný problém [8, 23]. Základní struktura dopředné neuronové sítě je zobrazena na obrázku 3.1.



Obrázek 3.1: Příklad dopředné neuronové sítě se vstupní, výstupní vrstvou a dvěma plně propojenými skrytými vrstvami.

3.3 Učení neuronové sítě

Učení umělého neuronu a tedy celé neuronové sítě spadá do kategorie učení s učitelem. To znamená, že je zapotřebí mít k dispozici datovou sadu, která je složena ze vzorků v podobě dvojic. Každou dvojici tvoří vstupní data, která mají být sítí zpracována (typicky značíme x), a výsledek, který očekáváme na výstupu (y). Učení má potom za cíl minimalizovat celkovou chybu, která je určena rozdílem mezi reálným výstupem sítě \hat{y} a očekávaným výsledkem y . Učení bez učitele se liší v tom, že je závislé pouze na vstupních hodnotách, u kterých není známý správný výsledek. Úspěch je obvykle určen tím, zda je síť schopna snížit nebo zvýšit související cenovou funkci [8].

Pro úspěšné natrénování neuronové sítě je důležitý návrh její architektury, datová sada, výběr cenové funkce a metoda optimalizace parametrů. Učení je založeno na snižování cenové funkce pomocí úprav parametrů sítě. Abychom zjistili, jakým způsobem provést změny, potřebujeme nejdříve vypočítat gradient cenové funkce. K tomu slouží algoritmus zpětného šíření. Metoda gradientního sestupu potom určuje, jak změnit samotné parametry, abychom dosáhli lepších výsledků. Oba algoritmy jsou popsány v této kapitole.

Na začátku učení jsou parametrům sítě přiřazeny náhodné malé hodnoty. Potom postupně zpracováváme vzorky x_i a na základě výsledku cenové funkce upravíme váhy, které ovlivňují jednotlivé vstupy neuronů. Epochou nazýváme jeden průchod celou datovou sa-

dou. Tento postup se opakuje dokud není dosaženo požadovaného cíle, například zvolené hodnoty chybové funkce na validačním datasetu, nebo když se začnou objevovat známky přetrénování [42].

Cenová funkce

K určení, jak dobře neuronová síť odhaduje výsledky, potřebujeme spočítat rozdíl mezi očekávaným a reálným výstupem. K tomu slouží cenová (cost) funkce $J(\theta)$, kde θ značí parametry sítě. Když počítáme rozdíl jen jednoho vzorku dat, používá se někdy označení chybová (loss, error) funkce. Při učení je cílem dosáhnout co nejmenší cenové funkce. V ideálním případě by to bylo $J(\theta) = 0$, což znamená, že mezi výstupy žádný rozdíl není. Ve většině případů se při učení používá metoda maximální věrohodnosti (maximum-likelihood). To znamená, že cenová funkce je záporná logaritmická věrohodnostní funkce 3.1 (negative log-likelihood), která může být ekvivalentně vyjádřena jako křížová entropie (cross entropy) [8].

$$L(y, \hat{y}) = - \sum_i^k y_i \cdot \log \hat{y}_i \quad (3.1)$$

Jiným příkladem cenové funkce je střední kvadratická chyba (mean squared error) nebo střední absolutní chyba (mean absolute error) [8]. Tyto funkce však mohou vytvářet velmi malé gradienty, což představuje komplikaci při optimalizaci metodou gradientního sestupu. Z tohoto důvodu se příliš často nepoužívají.

Aktivační funkce

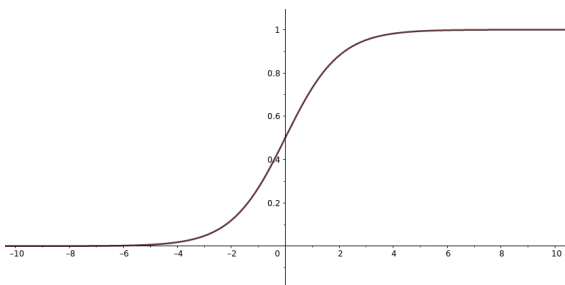
Při návrhu neuronové sítě je důležitý výběr aktivační funkce, která přináší do modelu nelinearitu. Díky tomu je možné, aby se síť naučila libovolnou nelineární funkci (podle množství neuronů). Teoreticky může mít každý neuron svou vlastní aktivační funkci, v praxi se však používá jedna funkce pro celou vrstvu, často i pro všechny skryté vrstvy modelu. Aktivační funkce zpracovávají jedno číslo složené ze vstupů neuronu a provádějí na něm určitou matematickou operaci. Existuje několik různých aktivačních funkcí, nejčastěji používané jsou zmíněny v této sekci, která čerpá informace z [2, 23].

Sigmoida

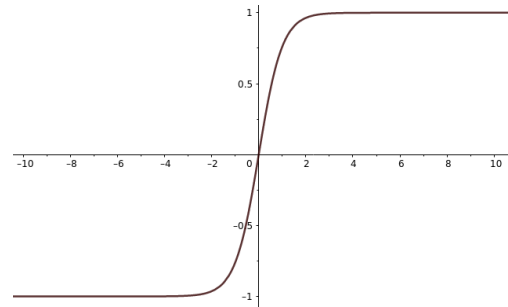
Sigmoida 3.2 má na vstupu reálné číslo, které přepočítá do intervalu mezi nulou a jedničkou. Dříve byla často využívána, ale dnes už není moc k vidění. Jednou z nežádoucích vlastností je, že dochází k saturaci funkce a gradient je v tomto místě téměř nulový. Důležitá je v tomto případě také inicializace vah neuronu, aby nedošlo k saturaci. Pokud budou váhy příliš velké, neurony budou saturované a síť se nebude schopná učit. Kromě toho může docházet k problému mizejícího gradientu (vanishing gradient). Ten se objeví, pokud je v daném místě gradient příliš malý a dalším násobením se ještě zmenšuje. To má za následek, že neuronem neprochází téměř žádný signál.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

Podobné vlastnosti má funkce hyperbolický tangens (tanh). Na rozdíl od sigmoidy přepočítává vstup do intervalu (-1,1). Díky tomu, že je výstup funkce centrován kolem nuly, je využívána častěji než sigmoida. Grafy funkcí můžeme vidět na obrázcích 3.2 a 3.3.



Obrázek 3.2: Sigmoida



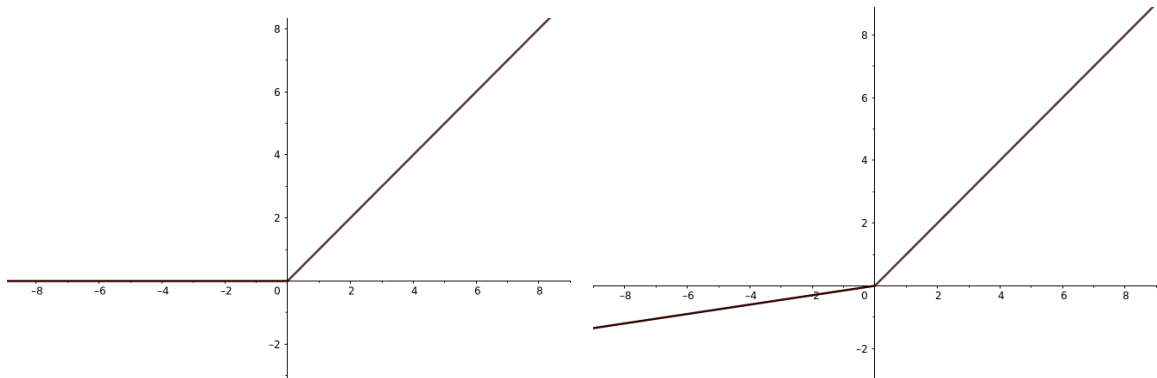
Obrázek 3.3: Hyperbolický tangens

ReLU

Asi nejvíce používaná aktivační funkce v poslední době je ReLU (z anglického Rectified Linear Unit), jejíž vzorec ukazuje rovnice 3.3. Jde o jednoduchou nelineární funkci, u níž pro kladné hodnoty zůstává výsledek stejný jako vstup, záporné hodnoty jsou nahrazeny nulou. Oblíbená je především proto, že je jednoduchá na implementaci a urychluje učení gradientním sestupem. Oproti funkcím sigmoid a tanh, které obsahují složitější výpočty (exponenty), je tato funkce proveditelná jednoduchým prahováním. Velké gradienty však mohou způsobit upravení vah neuronu takovým způsobem, že už nedojde k aktivaci neuronu při učení.

$$f(x) = \max(0, x) \quad (3.3)$$

Jedním z pokusů vyřešit tento problém je varianta Leaky ReLU. Od původní verze ReLU se liší v tom, že pokud je vstup menší než 0, funkce není nulová, ale má malý sklon, viz obrázek 3.4. Díky tomu gradient v záporných hodnotách nemizí.



(a) ReLU

(b) Leaky ReLU

Obrázek 3.4: Aktivační funkce ReLU a její varianta.

Backpropagation

Při učení sítě dochází v první fázi k tzv. dopřednému šíření. Informace v tomto případě putuje sítí od vstupu přes skryté vrstvy až na výstup, kde z výsledku může být spočítána hodnota cenové funkce. Algoritmus zpětného šíření chyby (backpropagation) je metoda, jakou lze vypočítat gradient cenové funkce v závislosti na parametrech sítě. Je to způsob,

kterým se hodnota cenové funkce šíří zpětně modelem. Vypočítaný gradient potom využívá metoda učení nazvaná gradientní sestup [8].

Gradient Descent

K minimalizaci cenové funkce se používá optimalizační algoritmus gradientní sestup (gradient descent) [8], který určuje způsob nalezení lokálního minima funkce. Gradient udává v konkrétním bodě směr a sílu, jakou funkce nejrychleji roste. Posouváním se po malých krocích proti směru gradientu můžeme snížit cenovou funkci. Krok posunutí se typicky definuje jako nějaká malá konstanta.

Problém při optimalizaci funkce představují tzv. stacionární body, u nichž nemůžeme určit, kterým směrem se posunout. Patří mezi ně lokální minima a maxima. Lokální minimum je bod, v němž má funkce menší hodnotu než všechny jeho okolní body a tedy už nejde provést krok níže. Další komplikací při učení je, že funkce může mít velké množství lokálních a globálních extrémů. Cílem proto není vždy najít globální minimum, ale alespoň velmi nízkou hodnotu.

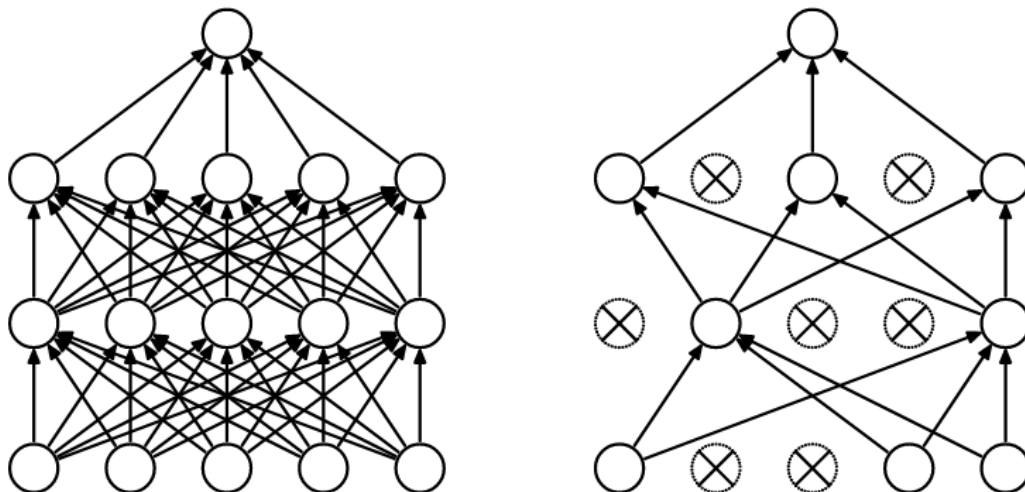
Pro dobrou generalizaci modelu je zapotřebí velké množství trénovacích dat. Výpočet gradientu a aktualizace parametrů při každém vzorku je ale výpočetně náročná a pomalá. Řešením je gradient odhadnout jako průměr z určitého malého souboru dat. Tato metoda se nazývá Stochastic Gradient Descent a jedná se o nejčastěji používanou variantu gradientního sestupu. Data se rozdělí na menší části o velikosti několika stovek vzorků, takzvané minibatch, a počítá se jejich průměr. Díky tomu můžeme model trénovat na obrovském množství dat pomocí aktualizací parametrů spočítaných pouze z malých skupin vzorků.

Generalizace a přetrénování

Pro naučený model je důležitá generalizace, tedy jeho schopnost mít dobré výsledky i na datech, které nikdy neviděl. Během učení počítáme chybu vzniklou na trénovacím datasetu (training error), která by se měla učením zmenšovat. Chyba na testovacích datech (test error, generalization error) je definována jako očekávaná hodnota chyby na novém vstupu a také chceme, aby byla co nejmenší. Určujeme ji měřením úspěšnosti modelu na testovacích datech, které jsou různé od trénovacího souboru. Míru naučení modelu určujeme podle toho, jak velká je chyba na trénovací množině a jak velký je rozdíl mezi chybou na trénovacích a testovacích datech [8, 10].

Neuronové sítě s velkým počtem neuronů jsou schopny vyjádřit složité funkce. Ačkoli se to může zdát výhodné, protože se síť může naučit komplikovanější data, častěji může docházet k přetrénování (overfitting). To znamená, že model se příliš přesně naučí příznaky trénovacích dat a jejich šum, a nedokáže potom správně klasifikovat data nová. K přetrénování dochází ve chvíli, kde je rozdíl mezi chybou na trénovací a testovací sadě příliš velký. Naopak pokud je velká chyba na trénovacím souboru, síť se nedokázala dostatečně naučit rozlišovat příznaky - underfitting. Pravděpodobnost přetrénování můžeme ovlivňovat množstvím parametrů sítě. Čím méně parametrů síť má, tím méně je pravděpodobné, že dojde k přetrénování [8, 10].

Jednou z metod jak předcházet přetrénování je tzv. dropout. Při jeho aplikaci jsou neurony v síti používány jen se zvolenou pravděpodobností. To znamená, že v určité fázi učení jsou aktivní jen některé neurony a ostatní jsou dočasně odstraněny i spolu s jejich propojeními, viz obrázek 3.5. Tento přístup má podobný efekt, jako bychom trénovali několik menších sítí, které sdílejí váhy. Ačkoli při použití dropout zabere učení sítě více času, ukázalo se, že tato metoda zlepšuje úspěšnost sítě a pomáhá zmírnit přetrénování [38].



Obrázek 3.5: Vlevo plně propojená neuronová síť, vpravo síť po aplikaci dropout [38].

Cross-Validation

Někdy se může stát, že pro danou úlohu nemáme dostatečné množství dat na to, abychom po rozdělení na trénovací a validační podmnožiny mohli provést dostatečně spolehlivé vyhodnocení. Tento problém pomáhá řešit křížová validace (cross-validation), která je sice výpočetně náročnější, ale umožňuje efektivně využít dostupná data. Nejčastěji používaná varianta je tzv. k -fold cross-validation. Hlavní myšlenka je náhodně rozdělit data do k skupin viz 3.6, které se navzájem nepřekrývají, a opakovaně na nich provádět trénování a validaci. V i -té iteraci je vždy i -tá skupina použita pro validaci a zbytek pro trénování. Chyba na validační množině dat pak může být určena jako průměr z chyb v jednotlivých iteracích [8].

	trénovací data				
$i = 1$	fold 1	fold 2	fold 3	fold 4	fold 5
$i = 2$	fold 1	fold 2	fold 3	fold 4	fold 5
$i = 3$	fold 1	fold 2	fold 3	fold 4	fold 5
$i = 4$	fold 1	fold 2	fold 3	fold 4	fold 5
$i = 5$	fold 1	fold 2	fold 3	fold 4	fold 5

Obrázek 3.6: Rozdělení dat při 5-fold cross-validation v jednotlivých iteracích. Modře jsou vyznačena validační data.

3.4 Konvoluční neuronové sítě

Konvoluční neuronové sítě (CNN) jsou jedním z typů umělých neuronových sítí. Zaměřeny jsou především na úlohy spojené s detekcí a klasifikací objektů v obraze. Jejich vstupem jsou nejčastěji přímo obrazová data, která mají podobu pixelů uspořádaných do 2D mřížky. Jak jejich název napovídá, charakteristickou operací je zde konvoluce. Kromě ní najdeme v architektuře CNN takzvané pooling vrstvy. Typická CNN architektura se skládá z několika konvolučních a pooling vrstev, které slouží pro extrakci příznaků, a na ty navazují

plně propojené vrstvy stejně jako v klasických neuronových sítích, které můžou provádět například klasifikaci.

Konvoluční vrstva

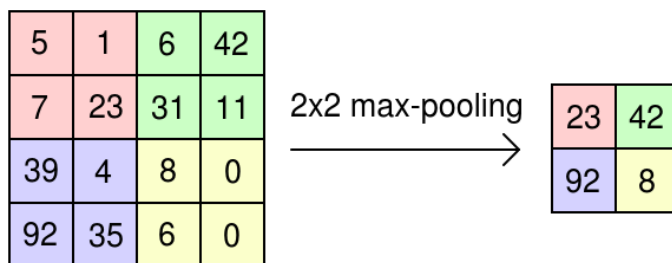
Konvoluční vrstvy [8, 23] jsou pojmenovány podle operace, kterou provádějí - konvoluci. Vstupem je vícerozměrné pole dat, které je násobeno s jádrem - vícerozměrné pole parametrů. Například pro dvourozměrný obraz používáme dvourozměrné jádro. Jádro konvoluce nazýváme také aktivační mapa. V každé vrstvě typicky používáme několik takových map, které se učí rozpoznávat v obraze různé znaky (přímky, rohy, barvy). Stejně jádro se vždy aplikuje na celý obraz, což významně snižuje počet potřebných parametrů. Na výstup konvoluční vrstvy je většinou aplikována některá z aktivačních funkcí, typicky se jedná o ReLU.

Konvoluční vrstvy můžou významně snížit složitost modelu prostřednictvím tří hyperparametrů: hloubky (depth), kroku (stride) a nastavení nulové výplně (zero-padding). Hloubka určuje počet filtrů, které chceme použít pro učení různých charakteristik v obraze. Nastavení kroku definuje, o kolik pixelů se posouvá jádro filtru. Nejčastější hodnoty jsou 1 nebo 2.

Nastavení nulové výplně umožňuje kontrolovat velikost výstupních dat. To je prováděno případným rozšířením okrajů obrazu o nulové pixely. Za zmínku stojí tři varianty této metody. Bez doplnění okrajů je možné konvoluci provést pouze v částech vstupních dat tak, aby celé konvoluční jádro překrývalo obraz. Pokud má v tomto případě obraz šířku m a jádro šířku k , výstup vrstvy bude mít šířku $m-k+1$. Protože při každé aplikaci konvoluční vrstvy dochází ke zmenšení obrazu, je omezený celkový počet těchto vrstev. Druhou variantou je doplnění tolika nul, aby byla zachována velikost vstupního obrazu. Síť tak může být tvořena libovolným počtem konvolučních vrstev. Krajiní vstupní pixely obrazu mají ve výsledku menší vliv na výstup než střední pixely. Další možností je přidání dostatek nulových pixelů tak, aby byl každý vstupní pixel ve výpočtu použit k -krát v každém směru. Dochází pak ke zvětšení a výstupní obraz bude mít šířku $m+k-1$. Použití nulových okrajů závisí na konkrétní architektuře, nejčastěji používané jsou první dvě zmíněné možnosti.

Pooling vrstva

Pooling vrstvy se typicky vkládají mezi vrstvy konvoluční. Jejich úkolem je mimo jiné zmenšit počet parametrů v síti a tedy snížit prostorovou a výpočetní náročnost. Pooling je realizován aplikací některé ze statistických funkcí na určité okolí pixelu. Mezi nejčastěji používané patří funkce maxima (max pooling) viz obrázek 3.7, průměr nebo vážený průměr závislý na vzdálenosti od středu.



Obrázek 3.7: Ukázka operace max-pooling s velikostí filtru 2x2.

Dalším efektem pooling vrstvy je zavedení invariance vůči malým posunům na vstupu. To znamená, že i když dojde k mírnému posunutí objektu v obraze, výstup z pooling vrstvy

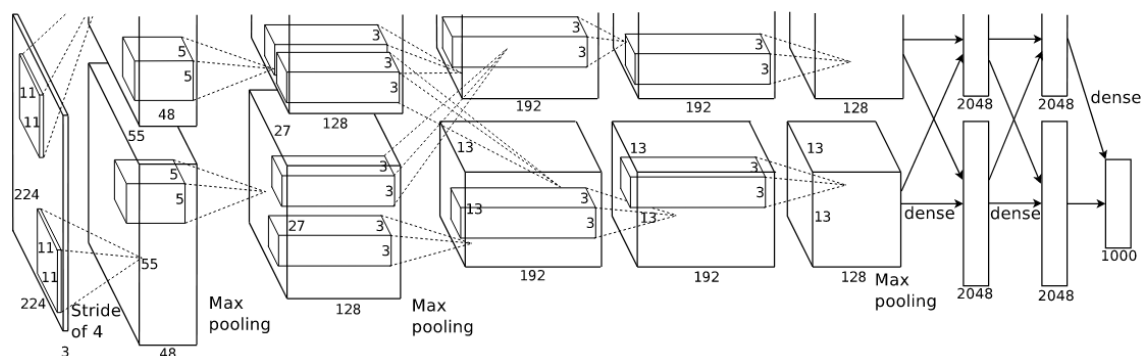
se téměř nezmění. Pooling vrstva sdružuje informaci z okolí určité velikosti a je tak možné použít menší počet pooling jednotek. Pro klasifikaci většinou nezáleží na přesné pozici pixelů objektu, ale jestli se v dané oblasti vyskytují hledané příznaky.

Pro ušetření množství výpočtů je možné pro pooling operaci nastavit velikost kroku (stride). Díky tomu dochází k přeskočení některých hodnot ze vstupu. I když je možné definovat různě velký krok pro různé směry pohybu, nejčastěji používaná kombinace je velikost filtru 2x2 s velikostí kroku 2 [8, 23, 28].

3.5 Architektury CNN

AlexNet

Alex Krizhevsky a spol. v roce 2012 navrhl architekturu AlexNet [21], která zpracovává barevné obrázky velikosti 224x224. Model se skládá z 5 konvolučních vrstev, mezi nimiž jsou 3 max-pooling vrstvy a na závěr 3 plně propojené vrstvy. Síť je zakončena výstupní vrstvou se softmax aktivační funkcí, která klasifikuje obrázky do 1000 tříd. První použité konvoluční jádro má velikost 11x11, nejmenší jádro 3x3. Jako aktivační funkce byla použita ReLU, která umožňuje rychlejší učení než dříve užívaná *tanh*. Síť byla trénována na několika GPU. Pro snížení míry přetrénování využili augmentaci dat a tehdy novou metodu dropout v prvních dvou plně propojených vrstvách. V roce 2012 model Alexnet vyhrál soutěž ILSVRC (ImageNet Large Scale Visual Recognition Challenge) v testu top-5 s chybovostí 15.3%, což je výrazně lepší výsledek oproti práci umístěné na druhém místě. Podrobný náčrt architektury je na obrázku 3.8.



Obrázek 3.8: Architektura sítě AlexNet [21].

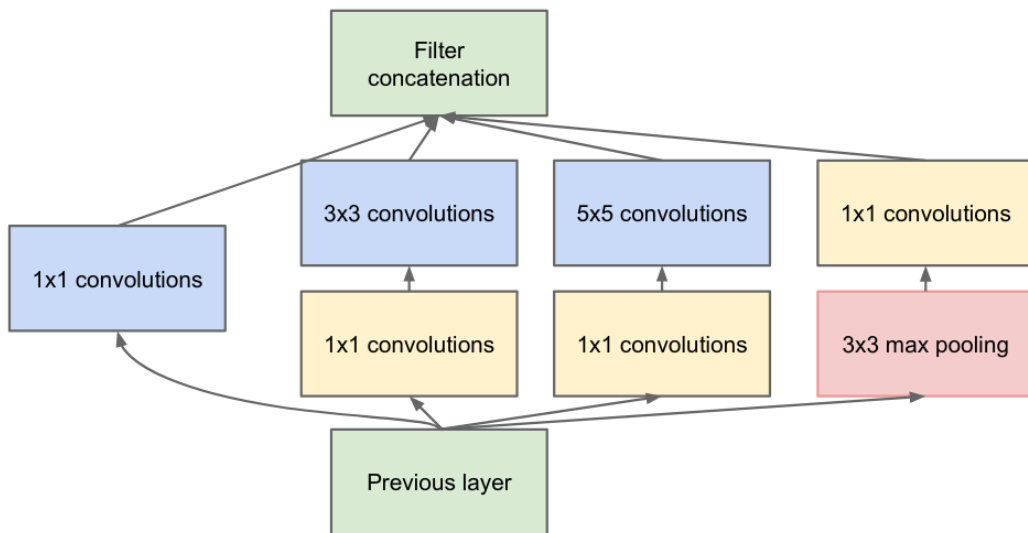
VGG

Simonyan a Zisserman [36] se zaměřili na prohloubení CNN pomocí malých konvolučních filtrů. Vytvořili několik modelů CNN s podobnou strukturou, které měly 11-19 vrstev ale různý počet konvolucí. Vstupem je RGB obraz o velikosti 224x224 pixelů. Následuje několik konvolučních vrstev mezi nimiž najdeme 5 max-pooling vrstev. Pro konvoluci jsou použita jádra velikosti 3x3 s rozšířením okrajů tak, že je zachována velikost obrazu. Na řadu konvolučních vrstev navazují tři plně propojené vrstvy, z nichž poslední provádí klasifikaci do 1000 tříd. Stejně jako v AlexNet jsou použity ReLU aktivační funkce a soft-max pro závěrečnou klasifikaci. Zmenšením konvolučních filtrů došlo k výraznému snížení počtu parametrů

v síti. Zároveň se ukázalo, že prohloubení sítě zvyšuje úspěšnost klasifikace v porovnání s menšími sítěmi, které využívají větší filtry.

Inception (GoogLeNet)

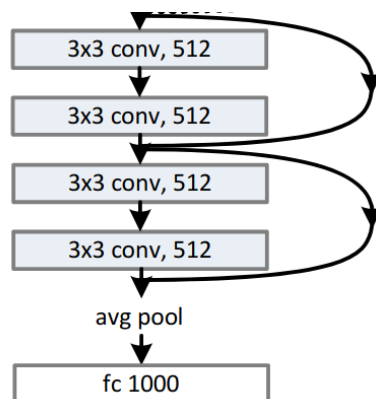
Zvýšit úspěšnost sítě je možné zvětšením její hloubky a šířky, to ale přináší problém přetrénování a velké výpočetní nároky. Szegedy a spol. [39] měli jako hlavní cíl vylepšit úspěšnost CNN a zároveň zachovat relativně malou výpočetní náročnost. Proto provedli experimenty s několika variantami sítě nazvané Inception. Konkrétní varianta pojmenovaná GoogLeNet vyhrála v roce 2014 soutěž ILSVRC. Dříve úspěšnou architekturu AlexNet překonala v testu top-5 s chybovostí 6.67% a přitom měla 12x méně parametrů. Architektura sítě je složena z několika konvolučních a pooling vrstev s aktivační funkcí ReLU. Využívá také dropout a softmax funkci v poslední vrstvě pro klasifikaci do jednotlivých tříd. Hlavní součástí jsou takzvané Inception bloky, které jsou tvořeny několika paralelně uspořádanými konvolučními a max-pooling vrstvou, jak můžeme vidět na obrázku 3.9. Důležitou funkci zastává konvoluce 1x1 ke snížení množství výpočtů.



Obrázek 3.9: Inception blok [39].

ResNet

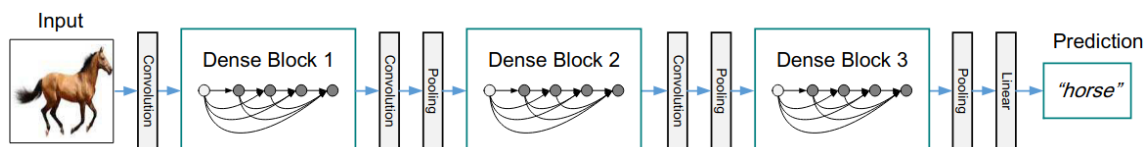
He a spol. [13] se snažili najít způsob, jak zjednodušit trénování sítě tak, aby mohli zvětšit její hloubku a tím dosáhnout větší úspěšnosti. Jako inspiraci použili architekturu hluboké VGG sítě. ResNet (z anglického residual network) je tak až 8x hlubší než síť VGG ale přesto má menší výpočetní náročnost. ResNet je podobně jako VGG složena z velkého množství malých konvolucí s jádrem 3x3. Základním principem reziduálního učení je propojení vrstev zkratkami, které umožňují přeskočit několik vrstev sítě. Zkratky tedy přenášejí výstup z jedné vrstvy na vstup vrstvy o několik úrovní dál viz obrázek 3.10. Díky tomu je redukován problém mizejících gradientů a zároveň je možné trénovat velmi hluboké sítě. ResNet nevyužívá metodu dropout a její závěr je tvořený avg-pooling vrstvou a plně propojenou vrstvou se softmax funkcí pro klasifikaci. V roce 2015 ResNet vyhrála klasifikaci v soutěži ILSVRC s chybovostí 3.57%.



Obrázek 3.10: Ukázka části ResNet propojení [13].

DenseNet

DenseNet [15] (Dense Convolutional Network) je architektura inspirovaná sítí ResNet, jejímž hlavním znakem je vytvoření zkratk mezi některými vrstvami. DenseNet tuto myšlenku posouvá dál a uvádí model, který propojuje každou jednu vrstvu se všemi ostatními vrstvami v síti, které jsou před ní. Každá vrstva má tak na vstupu všechny výstupy z předcházejících vrstev a zároveň svůj výstup posílá na vstup každé z následujících vrstev. Díky tomuto přístupu je zmírněn problém mizejících gradientů, více jsou využity příznaky z jednotlivých vrstev a zároveň má síť méně parametrů. Hlavním rozdílem mezi ResNet a DenseNet architekturou je, že DenseNet provádí konkatenci vstupů (ResNet je sčítá). DenseNet je složena z několika tzv. dense bloků, které jsou propojeny konvoluční a pooling vrstvou, jak můžeme vidět na obrázku 3.11. Tento model dosahuje lepších výsledků než dosavadní síť a zároveň má méně parametrů.



Obrázek 3.11: Dense bloky propojené konvucemi a pooling vrstvami [15].

EfficientNet

Tan a Le [41] se ve svém výzkumu zaměřili na to, jakým způsobem nejlépe rozšiřovat rozměry sítí tak, aby efektivně zvýšili jejich úspěšnost. Mezi způsoby, jak rozšířit architekturu CNN, patří především zvětšení její hloubky, šířky, nebo velikosti vstupního obrazu. Dříve zveřejněné práce se zaměřovaly na změnu pouze jedné z těchto dimenzí. Tan a Le přišli s novým přístupem, který upravuje všechny tři dimenze současně pomocí definovaných koeficientů. V rámci výzkumu také ukázali, že tento postup funguje i na modelu ResNet a že efektivita závisí na architektuře sítě. Proto se rozhodli vytvořit vlastní model nazvaný EfficientNet a několik jeho variant. Tato skupina modelů dosahuje ve srovnání s jinými architekturami vysoké úspěšnosti s podstatně menším množstvím parametrů. V porovnání se sítí ResNet-50 má model EfficientNet-B4 o 6,7% vyšší úspěšnost v testu top-1.

3.6 Knihovny pro počítačové vidění

Počítačové vidění a strojové učení je velmi rozšířená oblast a proto existuje velké množství knihoven, které efektivně implementují základní algoritmy i pokročilé funkce. Často podporují různé programovací jazyky a většinu operačních systémů. Mezi nejpoužívanější knihovny s otevřeným zdrojovým kódem patří například Tensorflow, Keras, PyTorch, OpenCV a další.

OpenCV

OpenCV [30] je široce používaná knihovna pro analýzu obrazu a videa. Napsána je v jazyku C a C++ a podporuje řadu programovacích jazyků jako je Python, Java nebo MATLAB. OpenCV je navrženo tak, aby provádělo efektivní a rychlé výpočty, které jsou užitečné pro programování sofistikovaných aplikací a zpracování obrazu v reálném čase. Knihovna obsahuje přes 500 funkcí rozdělených do několika modulů, který je každý zaměřený na jinou oblast počítačového vidění, např. filtrování, zpracování videa, uživatelské rozhraní, detekce klíčových bodů a objektů v obraze a mnoho dalších. Její využití je mimo jiné v oblastech pro kontrolu výroby, zpracování obrazu v lékařství, kalibrace kamery, stereofonní vidění a robotika. Knihovna obsahuje také modul pro strojové učení, který je zaměřený především na algoritmy pro statistické rozpoznávání vzorů a shlukování využitelné v různých problémech strojového učení. Knihovna je dostupná pro nejpoužívanější operační systémy Windows, Linux a Mac OS X a také pro mobilní aplikace na Android a iOS. Podporuje 32bitové a 64bitové architektury Intel a AMD (x86, x64) a také ARM [19].

Tensorflow

Tensorflow [1] je rozhraní pro implementaci algoritmů strojového učení. Jádro Tensorflow je implementováno v C++ kvůli přenositelnosti a podporuje tak řadu operačních systémů jako Linux, Windows, Mac OS X, Android a iOS. Knihovnu lze využít k realizaci široké škály algoritmů pro trénování modelů hlubokých neuronových sítí. Nejčastěji je používána ve spojení s programovacími jazyky Python a C++. Výpočet Tensorflow je reprezentován orientovaným grafem, který se skládá ze sady uzlů s různými vlastnostmi. Každý uzel má několik vstupů a výstupů a představuje instanci operace. Výpočty mohou být realizovány na různých systémech, jako jsou rozsáhlé distribuované systémy z mnoha výpočetních jednotek GPU, nebo mobilní telefony a tablety. Systém je zaměřený na to, aby byl flexibilní pro rychlé experimentování s novými modely pro výzkumné účely a měl i dostatečně vysoký výkon. Je používán při nasazování systémů strojového učení do výroby v odvětví jako počítačové vidění, rozpoznávání řeči, získávání informací, zpracování přirozeného jazyka a v robotice.

Keras

Keras [6] je vysoko-úrovňové rozhraní pro řešení problémů hlubokého učení. Implementované je v jazyku Python jako nadstavba platformy Tensorflow pro strojové učení. Keras obsahuje implementace vrstev sítí, aktivačních funkcí, nástrojů pro načítání vstupních dat a dalších běžně používaných částí pro budování neuronových sítí. Rozhraní poskytuje základní abstrakce a stavební bloky a umožňuje tak uživateli soustředit se na provádění experimentů. Keras má také rozsáhlou dokumentaci s příklady a průvodce pro vývojáře. Rozhraní umožňuje také vývoj modelů pro spuštění na iOS, Android, nebo přímo v prohlížeči.

PyTorch

PyTorch [31] je knihovna napsaná v jazyku Python pro programování algoritmů strojového učení. Je odvozena od knihovny Torch a zaměřená jak na použitelnost tak na rychlost. PyTorch je efektivní a podporuje hardwarovou akceleraci na GPU. Tato knihovna na rozdíl například od TensorFlow nebo Caffe umožňuje libovolně měnit způsob, jakým se síť chová, s nulovým zpožděním nebo režii a okamžitě provádí tenzorové výpočty. Uživatelé nemusí před spuštěním čekat na kompilaci a lze sledovat průběžné výpočty pomocí standardních ladicích nástrojů. Její výkon je přitom srovnatelný s ostatními knihovnami pro strojové učení, které mají staticky definovanou strukturu sítě. Hlavním cílem je snadné a produktivní psaní modelů, načítání dat a optimalizace. Vrstvy sítě jsou obvykle vyjádřeny jako Python třídy jejichž konstruktory vytvářejí a inicializují své parametry. Podobně mohou být i modely reprezentovány jako třídy, které skládají jednotlivé vrstvy. Návrh neuronových sítí je možný jak v Pythonu, tak v C/C++ a jsou podporovány všechny standardní operační systémy.

Caffe

Caffe [16] je C++ knihovna implementující algoritmy pro hluboké učení s podporou jazyků Python a MATLAB. Díky oddělení reprezentace modelu od jeho implementace umožňuje snadný vývoj a experimentování na různých platformách. Caffe je vyvíjena především skupinou BVLC (Berkeley Vision and Learning Center) s důrazem na rychlost a modularitu. Její aplikace je v různých výzkumných projektech a rozsáhlé škále průmyslových aplikací. I když byla knihovna původně zaměřena na počítačové vidění, dnes je využívána v různých odvětvích jako je zpracování řeči, robotika nebo astronomie. Caffe poskytuje sadu nástrojů pro učení, testování a ladění modelů, včetně zdokumentovaných příkladů modelů některých sítí. Knihovna podporuje operační systémy Linux, OS X a Windows.

Kapitola 4

Návrh řešení

Cílem práce je navrhnout vhodnou metodu pro klasifikaci defektů netkaných textilií. V této kapitole je provedeno shrnutí a zhodnocení současného stavu a na základě toho zvolena metoda pro klasifikaci. Popsána je podoba dodaného datasetu a rozdělení vad do jednotlivých kategorií. Následně je uveden návrh práce a vyhodnocení.

4.1 Analýza současného stavu

Pro detekci defektů v textiliích jsou často používány metody využívající spektrální analýzu obrazu. Konkrétně se jedná o Gaborovy filtry, Fourierovy a vlnkové transformace. Pomocí těchto metod je možné určit oblast defektu, ze které jsou následně vypočítány vybrané příznaky. Takto lze získat informace o tvaru defektu jako je výška, šířka, plocha, nebo lze určit průměrné hodnoty jasu, odchylku, atd. Tyto příznaky jsou následně použity pro trénování klasifikátorů. Práce, které využívají tento přístup, často uvádějí vysokou úspěšnost detekce i klasifikace vad (přes 90%), ale vzorek dat, na kterém jsou testy prováděny, je poměrně malý.

Například v [35] je provedena klasifikace defektů tkanin do 6 kategorií na základě příznaků získaných FFT nebo statistických hodnot. Každá třída obsahovala pouze 30 testovacích snímků. Tvarem podobné kategorie bylo obtížné spolehlivě rozlišit. V [46] jsou příznaky získané pomocí Gaborových filtrů použity pro trénování GMM klasifikátoru s úspěšností 85%. V práci [9] dosahovala nejlepších výsledků klasifikace na základě příznaků získaných vlnkovou transformací. Klasifikovány byly však pouze 3 odlišné třídy.

Obecně lze shrnout, že ačkoli práce implementující tyto metody uvádějí vysokou úspěšnost klasifikace, pro problém definovaný v této práci by byly nedostatečné. Zároveň četnost dat použitých pro vyhodnocení je poměrně malá pro získání dostatečně objektivního výsledku. Klasifikace je prováděna do několika tříd, které jsou dobře vizuálně rozlišitelné. Většinou je lze odlišit už na základě jejich rozměrů a poměrů stran, případně barvy. Tento přístup by pravděpodobně nebyl schopný klasifikovat náročné kategorie defektů v této práci, které se liší jen v malých detailech.

Konvoluční neuronové sítě často dosahují vyšší úspěšnosti v oblasti klasifikace než jiné metody založené na ruční extrakci příznaků. Zároveň jsou schopny klasifikovat více druhů tříd, které se od sebe výrazně neliší. Ukázalo se, že hlubší sítě mohou mít lepší výsledky, avšak za cenu vyšších nároků na paměť a čas. Některé přístupy byly proto zaměřeny na vylepšování menších CNN, například využitím metadat.

V [3] bylo porovnáno několik architektur CNN pro rozdělení defektů do dvou tříd na závažné a nezávažné. V této práci využili také další informace o obrazu pro rozšíření příznaků. Ukázalo se, že zapojení metadat je schopné zvýšit úspěšnost sítě. V [47] vytvořili paralelní model ze 3 menších částí, který měl lepší úspěšnost než jednotlivé části zvlášť. V práci [25] se inspirovali sítí InceptionNet a vytvořili vlastní podobu mikroarchitektury, s níž dosahovala síť úspěšnosti 96%. Přehled vlastností některých postupů je uveden v tabulce 4.1.

	kategorie	celkem dat	úspěšnost v %	použité metody
Zhang [46]	9	720	85,0	Gabor + stat. + GMM
Jing [18]	5	225	98,7	Gabor + stat. + NN
Shady [35]	6	180	88,6	stat. + NN
Shady [35]	6	180	91,4	FFT + NN
Çelik [48]	4	180	96,3	GLCM + NN
Li [24]	6	180	87,8	SVM
Li [24]	6	180	85,9	BPNN
Wei [45]	10	500	97,9	CS-CNN
Calderisi [3]	2	71878	88,3	CNN+META
Zhao [47]	10	10500	95,7	VLSTM
Li [25]	3	27200	96,3	WACNet-3MA
Mottalib [26]	6	128	99,2	Bayes

Tabulka 4.1: Shrnutí některých vlastností vybraných přístupů.

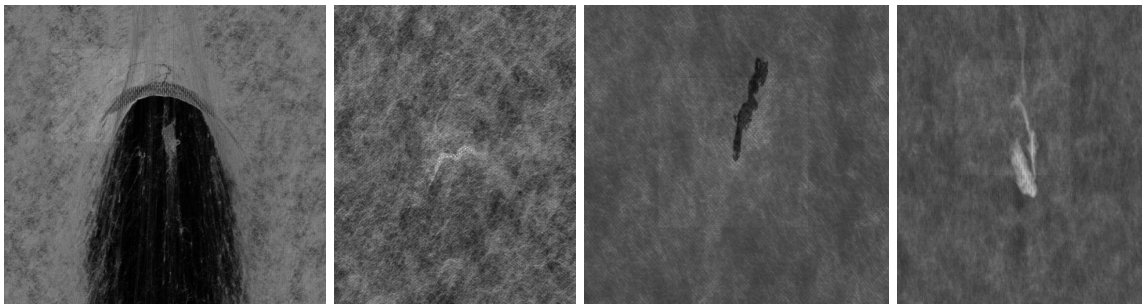
4.2 Popis dat

Sada dat k této práci byla poskytnuta firmou CAMEA spol. s r.o. Jedná se o snímky defektů, které vznikají při výrobě netkaných textilií, zachycené vysokorychlostními kamerami. Data jsou sesbírána z několika pracovišť, kde byl aplikován tento systém kontroly výroby. Snímky jsou před uložením zpracovány algoritmem, který analyzuje obraz, kompenzuje nerovnoměrné osvětlení a vytvoří výřez s detekovanou vadou.

Obrázky v datasetu jsou roztrženy do 26 složek podle druhu vady a typu osvětlení při jejím zachycení kamerou. Nenachází se zde snímky bez defektů. V datasetu se celkem vyskytuje 15 kategorií vad (viz tabulka 4.2), které mohou být zachyceny různými kamerami s odlišným osvětlením. Obecně se jedná o čtyři skupiny kategorií označené identifikačními čísly 9-12 podle druhu snímání:

- 9 - kamera se šikmým podsvitem, která produkuje tmavé vady na světlejším pozadí
- 10 - kamera se šikmým podsvitem, která produkuje světlé vady na tmavším pozadí
- 11 - kamera s nadsvitem, která produkuje tmavé vady na světlejším pozadí
- 12 - kamera s nadsvitem, která produkuje světlé vady na tmavším pozadí

Porovnání snímků z různých kamer ukazuje obrázek 4.1. Některé vady mohou být detekovány i více kamerami současně, jiné lze zaznamenat jen s určitým osvětlením. Ne vždy je na obrázku zachycen celý defekt. Komplikaci při klasifikaci vad představují především kategorie, které jsou si vzhledově velmi podobné, nebo tvarově velmi rozmanité defekty v rámci jedné kategorie.



Obrázek 4.1: Ukázka vad z jednotlivých způsobů snímání defektů. Zleva 9 - podsvit (úkap s dírou), 10 - podsvit (vlašťovka), 11 - nadsvit (nečistota), 12 - nadsvit (úlet).

Celkem je v souboru 7083 dat, které jsou nerovnoměrně rozloženy mezi jednotlivé kategorie. Mezi nejpočetnější defekty patří *zeslabené místo* (2299) a *prášení* (1324), nejméně se pak vyskytuje kategorie *muška* (13). Rozměry obrázků se pohybují v rozmezí 331-1241 pixelů na šířku a 554-1231 pixelů na výšku. Snímky mají různý poměr stran a jsou uloženy ve formátu JPEG.

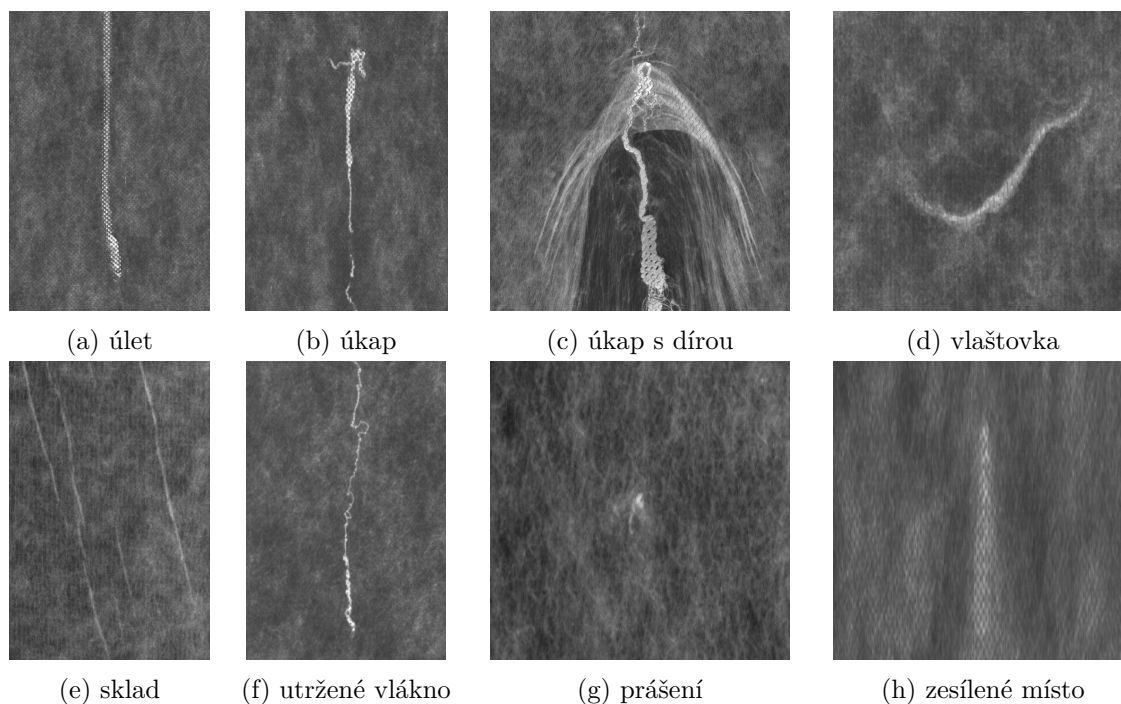
Kategorie vady	Typ snímání obrazu				Celkový počet výskytů vady
	9	10	11	12	
aviváž		14			14
díra	97				97
dlouhý tenký úlet				134	134
muška			13		13
nečistota	44		66		110
prášení				1324	1324
průlet	606		66		672
sklad		4		44	48
úkap	30	56		250	336
úkap s dírou	67			92	159
úlet	96			267	363
utržené vlákno		18		461	479
vlašťovka		39		413	452
zesílené místo		2		581	583
zeslabené místo	1732		567		2299
Celkový počet vad	2672	133	712	3566	7083

Tabulka 4.2: Přehled počtu defektů v jednotlivých kategoriích.

K dispozici jsou také metadata o každém snímku ve formátu csv, která ukládá detekční algoritmus. Jedná se například o informace o jasových hodnotách obrazu, odhadovaná pozice a velikost vady, typ osvětlení, a další. Tato data byla ve firmě analyzována a bylo zkoumáno, zda je vhodné využít je pro klasifikaci. Bylo však vyhodnoceno, že na základě těchto vypočtených parametrů nelze dostatečně spolehlivě klasifikovat, jelikož jsou příliš nepřesné.

4.3 Návrh řešení

Po konzultaci s vedoucím práce a zástupcem firmy CAMEA s .r. o. bylo rozhodnuto, že práce bude zaměřena na klasifikaci světlých vad zachycených kamerou s nadsvitem (viz skupina 12 v tabulce 4.2). Ostatní skupiny jsou dostatečně spolehlivě rozlišeny stávajícím systémem. Klasifikace defektů ze skupiny 12 však dosahuje úspěšnosti jen okolo 68% a je potřeba se na ni zaměřit. Jejich ukázka je na obrázku 4.2. Hlavním cílem práce je tedy vytvořit systém, který tuto úspěšnost překoná. Důraz je přitom kladen na to, aby byly co nejlépe klasifikovány kritické vady *úlet*, *úkap* a *utržené vlákno*, které dosahují v průměru f1-score jen 39%.



Obrázek 4.2: Ukázky vad z kategorií skupiny 12. Zvětšeno pro lepší viditelnost.

Na základě zadání a požadavků potenciálního odběratele a po nastudování dosud používaných metod a zhodnocení jejich výsledků, jsem se rozhodla řešit uvedený problém využitím konvolučních neuronových sítí. Ukázalo se, že CNN vykazují velmi dobré výsledky jak v úlohách klasifikace vad textilií, tak v mnoha dalších oblastech. Dodaný dataset obsahuje kategorie defektů, které jsou si velmi podobné a liší se jen v malých detailech. Tyto by pravděpodobně nebylo možné rozlišit pouze na základě zvlášť vypočtených příznaků statistickými a spektrálními metodami, které používají některé práce. CNN jsou oproti tomu schopné naučit se extrahovat takové příznaky z obrazu, které jsou nejužitečnější pro klasifikaci.

Rozhodla jsem se, že pro klasifikaci defektů bude vybrán jeden ze základních modelů CNN, který dosáhne nejvyšší úspěšnosti na vybraných kategoriích. Navrženou CNN by mělo být možné zapojit do stávajícího systému kontroly výroby firmy CAMEA s .r. o. Rozhodla jsem se klasifikaci vad textilií zpracovat s následujícími vlastnostmi a parametry:

- úspěšnost klasifikace nad 68% (vyhodnocení viz níže)
- použitý dataset pro trénování a hodnocení: dodaný firmou CAMEA s .r. o.

- použitý programovací jazyk: Python¹
- použitý framework pro neuronové sítě: Keras²
- použitý systém pro testování:
 - OS: Microsoft Windows 10 Pro
 - GPU: NVIDIA Quadro P2200
 - CPU: Intel Core i5-4570S
 - RAM: 8 GB

4.4 Způsob vyhodnocení

Pro vyhodnocení úspěšnosti klasifikace do více tříd lze použít několik metrik [37]. Pro vyhodnocení práce budou použity především metriky f1-score a accuracy. K jejich definování je nejprve zapotřebí výsledky rozdělit do následujících skupin podle toho, jak byly klasifikovány:

- TP (true positive) - snímky, které byly správně označeny jako vybraná třída
- TN (true negative) - snímky, které byly správně označeny jako jiná třída
- FP (false positive) - snímky, které byly chybně označeny jako vybraná třída
- FN (false negative) - snímky, které byly chybně označeny jako jiná třída

Z těchto hodnot lze vytvořit tzv. matici záměn (confusion matrix), viz 4.3, a vypočítat jednotlivé metriky.

	výstup klasifikátoru	
	třída1	třída2
třída1	TP	FN
třída2	FP	TN

Tabulka 4.3: Matice záměn pro dvě třídy.

Metrika precision 4.1 počítá podíl správně klasifikovaných vzorků a všech vzorků označených jako vybraná třída. V ideálním případě se precision rovná 1. Toho dosáhneme v případě, že žádný obrázek není chybně označený jako vybraná třída (FP=0). I v tomto případě ale nemusí být správně klasifikovány všechny snímky vybrané vady. Metrika totiž nezohledňuje počet chybně označených snímků jako jiná než vybraná třída (FN). Pokud by model označil správně pouze 1 z 10 snímků dané třídy, i přesto by byl výsledek 1.

$$precision = \frac{TP}{TP + FP} \quad (4.1)$$

Recall 4.2 je metrika, která na rozdíl od precision zohledňuje počet FN, tedy takové vzorky, kdy je vybraná třída chybně označena jako jiná. I u této metriky chceme, aby se

¹<https://www.python.org/>

²<https://keras.io/>

v ideálním případě rovnala. Nedostatek této metriky spočívá v tom, že jako vybraná třída mohou být chybně označeny i snímky jiných tříd (FP) a i přesto bude výsledek blízký 1.

$$recall = \frac{TP}{TP + FN} \quad (4.2)$$

Pro dobrý klasifikátor chceme, aby se precision i recall rovnaly jedné. Metrika, která stejnou vahou kombinuje obě hodnoty se nazývá f1-score 4.3. Existují i její varianty, kde lze příslušným koeficientem dát metrice recall větší váhu.

$$f1\text{-score} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (4.3)$$

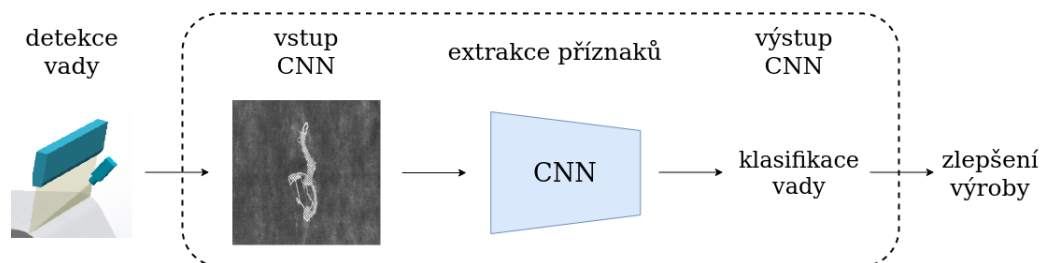
Accuracy 4.4 je metrika, která počítá podíl celkově správně klasifikovaných vzorků všech tříd a celkového počtu snímků. Určuje tak celkovou efektivitu klasifikátoru ale bez bližších informací o úspěšnosti klasifikací.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.4)$$

Kapitola 5

Implementace řešení a vyhodnocení

Cílem práce je navržení metody, která bude vhodná pro klasifikaci defektů netkaných textilií. Tato metoda má dosahovat větší úspěšnosti než stávající klasifikátor firmy CAMEA spol. s r. o. Pro řešení tohoto problému bude implementována konvoluční neuronová síť, která provádí klasifikaci a může být následně zapojena do systému kontroly výroby. Schéma takového systému ukazuje obrázek 5.1. Vstupem sítě je snímek s výřezem defektu netkané textilie, který byl vytvořen detekčním systémem. Jejím výstupem je klasifikace do jedné z vybraných tříd. Tato informace může být dále využita pro zlepšení výrobního procesu. Konvoluční síť bude klasifikovat světlé vady zachycené kamerou s nadsvítem (skupina 12). Na základě požadavku firmy CAMEA spol. s r. o. bude při vyhodnocení zvláštní pozornost věnována vadám *úkap*, *úlet* a *utržené vlákno*.



Obrázek 5.1: Schéma navrhovaného systému.

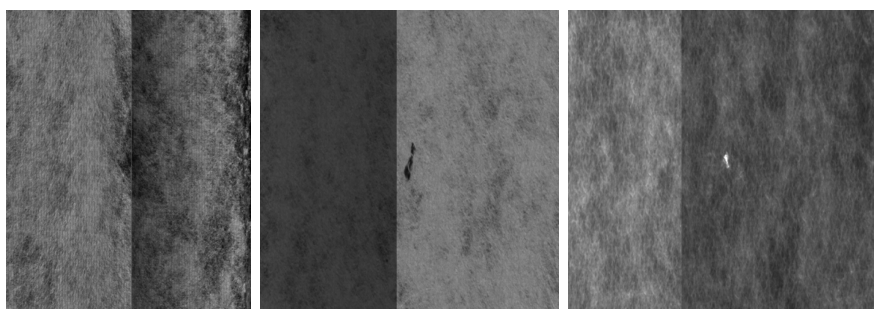
Za účelem vytvoření a natrénování co nejúspěšnější CNN je navrženo a realizováno několik experimentů, které jsou popsány dále v této kapitole. Pokusy jsou zaměřeny na různé složení vstupních dat, jejich augmentaci a velikost vstupního obrazu. Jejich cílem je vyhodnotit vliv jednotlivých úprav na výslednou klasifikaci. Následně je natrénováno několik základních modelů CNN a porovnána jejich úspěšnost. Architektura nejúspěšnějšího modelu je poté použita k vytvoření nového modelu, který na vstupu přijímá další informace o obrazu (původní velikost snímku) a využívá je pro klasifikaci. Postup práce lze shrnout do následujících kroků:

1. Kontrola dat a vyrovnání počtu snímků v jednotlivých kategoriích
2. Porovnání vlivu různého složení a velikosti vstupních dat
3. Natrénování základních modelů CNN pro klasifikaci vad textilií

4. Zapojení metadat do vybrané architektury
5. Vyhodnocení a srovnání vytvořené CNN s referenčním řešením

5.1 Předzpracování dat

Bližší prohlédnutí datasetu ukázalo, že se v něm nacházejí i snímky, které jsou výrazně pozměněné detekčním algoritmem. Při detekci vad je algoritmem upravován jas obrazu, což v něm může zanechat tmavé pásy různé velikosti, které někdy překrývají samotnou vadu. V některých kategoriích byly takové snímky častější a mohly by tak negativně ovlivnit průběh učení neuronové sítě. Celý dataset byl proto zkontrolován a výrazně pozměněné snímky byly odstraněny. Jejich příklad je na obrázku 5.2. Několik snímků bylo také zařazeno do špatné kategorie, což bylo opraveno.



Obrázek 5.2: Pozměněné snímky detekčním algoritmem.

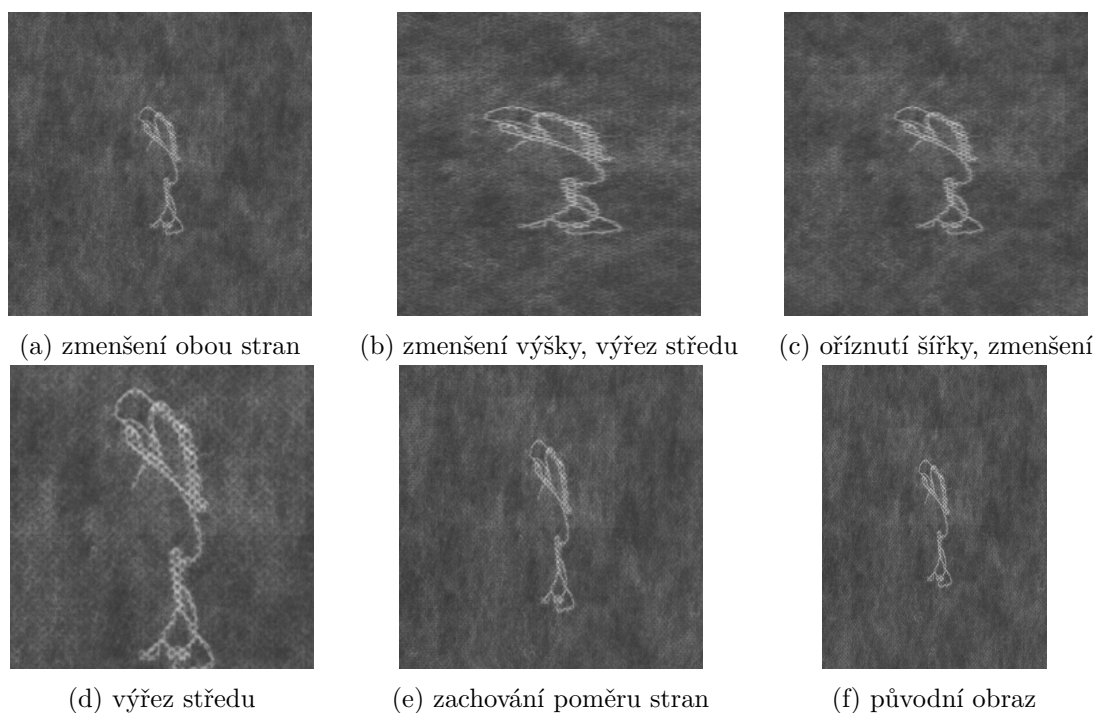
Protože je v některých kategoriích velmi málo snímků, dataset byl rozdělen do dvou částí. V té první je 7 kategorií, které obsahují alespoň 100 snímků, v druhé skupině zůstávají dvě méně početné kategorie. Pro experimenty je použita pouze větší skupina vad. Následně je zapotřebí data rozdělit na trénovací a validační část. Dělení v poměru 7:3 se po provedení testu ukázalo být vhodnější než poměr 8:2.

Dalším krokem je upravení počtu trénovacích snímků v jednotlivých kategoriích tak, aby všechny měly stejné zastoupení. Vyrovnání počtu snímků bylo dosaženo pomocí augmentace a podvzorkování. Málo početné kategorie byly pomocí horizontálního a vertikálního překlopení rozšířeny až 4x. V některých kategoriích (*prášení*) bylo snímků oproti ostatním vadám příliš mnoho, a tak byla tato část podvzorkována. K učení bylo využito jen tolik dat, kolika dosahovaly ostatní kategorie po augmentaci.

5.2 Rozměry obrazu

Snímky v datasetu se různí jak svou velikostí, tak poměrem stran. Proto je zapotřebí jejich velikost sjednotit tak, aby bylo možné je použít jako vstupní data do sítí, které přijímají data na vstupu v konkrétní velikosti. Jelikož jsou snímky vad příliš velké a jejich zpracování v původní velikosti by trvalo velmi dlouho, snímky byly podvzorkovány a/nebo ořezány.

Značná část defektů v datasetu je orientovaná v obraze na výšku, tedy ve směru, jakým se textilie pohybuje při výrobě. Často se stává, že dlouhé vady nejsou na snímku zachyceny celé a nahoře a/nebo dole obraz přesahují. Naopak na šířku je ve většině případů výřez až zbytečně velký. Jen malé množství snímků není dostatečně širokých, aby obsáhly celou plochu poškození. Mnohem častější bývá situace, kdy velkou část snímku zabírají pouze pixely



Obrázek 5.3: Různé podoby trénovacích snímků.

pozadí a uprostřed se nachází malý či úzký defekt. Z těchto důvodů bylo experimentováno s různými možnostmi zmenšení a ořezání snímků pro učení sítě. Cílem bylo odstranění pro úlohu zbytečných pixelů a zachování co největšího množství užitečných informací. Pro síť, které přijímají na vstupu obrázky velikosti $size \times size$ byly připraveny následující úpravy snímků (viz obrázek 5.3):

- (a) zmenšení obou stran obrazu na velikost $size \times size$
- (b) zmenšení výšky obrazu na velikost $size$, následně výřez ze středu o šířce $size$
- (c) redukce původní šířky obrazu na polovinu oříznutím obou okrajů, následně zmenšení na velikost $size \times size$
- (d) zmenšení velkých snímků na velikost 500×500 px, následně vytvoření čtvercového výřezu ze středu o velikosti $size \times size$
- (e) zmenšení obrazu se zachováním poměru stran tak, aby kratší strana odpovídala velikosti $size$ a následně oříznutí delší strany

Při zmenšování obrazu je zapotřebí vypočítat nové hodnoty pixelů. Způsob, jakým je to provedeno, může mít vliv na výsledky klasifikace. Proto byly testovány následující způsoby interpolace:

- nejbližší soused
- bilineární interpolace
- bikubická interpolace

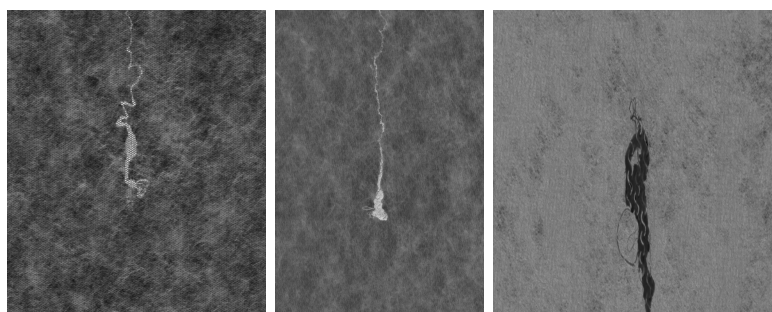
5.3 Složení trénovacích dat

Cílem této práce je správně klasifikovat defekty ze skupiny označené identifikačním číslem 12 (nadsvit, světlé vady). Některé kategorie těchto vad se však vyskytují i v jiných skupinách, viz 5.4. Například *úkap* nebo *úlet* bývá zachycen i kamerou s podsvitem (skupina 9). *Utržené vlákno* nebo *vlaštovka* se zase vyskytují i ve skupině 10. Ve skupině 11 se žádné z vybraných vad nenachází.

Jelikož je v datasetu poměrně málo dat, nabízí se myšlenka snímků defektů z různých skupin sloučit dohromady. Díky tomu dojde ke zvětšení vzorku dat pro učení sítě a zároveň to může přispět ke schopnosti sítě přizpůsobit se novým datům. Z tohoto důvodu bylo navrženo několik experimentů se sloučením kategorií defektů z různých skupin.

Při prvním pokusu byla využita data pouze skupiny 12. Dalším krokem bylo sloučení skupin 12 a 10, které mají vizuálně podobné snímky - světlá vada na tmavším pozadí. Následně byly přidány i vady ze skupiny 9 - tmavý defekt na světlém pozadí.

Další pokus byl proveden se sloučením kategorií *úlet* a *dlouhý tenký úlet*. Jedná se o rozlišení jednoho druhu vady do dvou kategorií. Obě mají stejný způsob vzniku, ale měly by se odlišovat tvarem. Po prohlédnutí datasetu se však ukázalo, že jsou si třídy velmi podobné. Několik snímků se dokonce nacházelo v obou kategoriích. Po konzultaci bylo dohodnuto, že tyto třídy není zapotřebí navzájem rozlišovat.

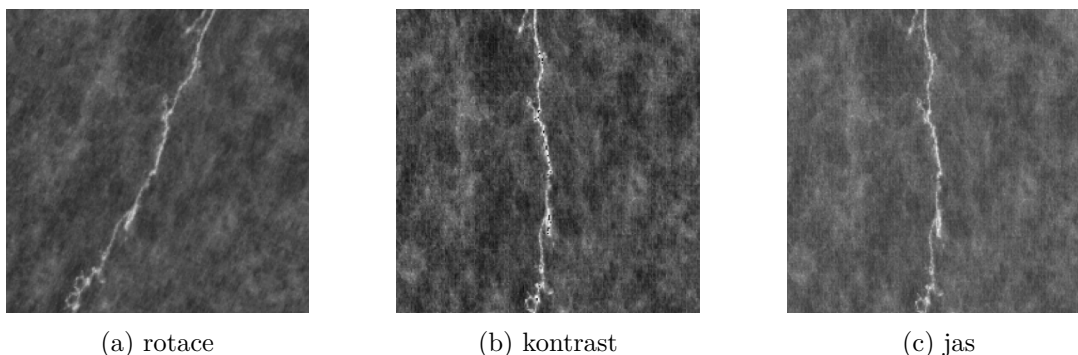


Obrázek 5.4: Úkap zachycený různými kamerami.

5.4 Augmentace

Augmentace je vhodná metoda pro rozšíření datasetu, pokud je v něm nedostatek dat pro trénování sítě. Úprava obrázků také přispívá ke schopnosti sítě správně klasifikovat nová data a snižuje míru přetrénování. Existuje několik způsobů, jak pozměnit snímek, mezi které patří například rotace, posun, úprava jasu, kontrastu, sytosti, přidání šumu. Ne vždy je však vhodné použít všechny metody. Například dlouhé vady jako *utržené vlákno* jsou na snímcích orientovány vertikálně (směr pohybu látky při výrobě) a nemá tak smysl rotovat je o 90 stupňů.

Kromě převrácení obrazu popsaného výše pro vyrovnání počtu dat v jednotlivých třídách, bylo dále experimentováno s ostatními transformacemi obrazu, a byl zkoumán jejich vliv na výslednou přesnost detekce. Testovány byly jak prostorové transformace, tak úpravy jasu pixelů. Příklady augmentace ukazuje obrázek 5.5. Augmentace snímků byla realizována pomocí vrstev zařazených před konvoluční vrstvy sítě. Koeficienty použité pro úpravu snímků byly určeny náhodně z definovaného intervalu.



Obrázek 5.5: Ukázky augmentace.

5.5 Modely CNN

Provedení experimentů s různým složením a velikostí dat umožňuje vybrat takovou podobu vstupních obrázků, která nejvíce zvyšuje úspěšnost klasifikace. Tyto snímky jsou následně použity pro trénování vybraných modelů konvolučních neuronových sítí.

Natrénováno bylo několik standardních modelů: VGG-16, Inception-V3, DenseNet-121, ResNet-50, EfficientNet-B0, jejichž vlastnosti jsou shrnuty v tabulce 5.1. Jejich architektury jsou blíže popsány v kapitole 3.5 a implementace byla využita z knihovny Keras. Protože dataset obsahuje málo dat na dostatečné natrénování celé sítě, použity jsou již sítě předtrénované. Aplikována je metoda přeneseného učení (transfer learning) a jsou využity váhy parametrů natrénované na datasetu ImageNet¹. Trénuje se pouze klasifikační část sítě.

Parametry modelů jsou vyladěny tak, aby dosahovaly co nejvyšší úspěšnosti na validační sadě. Testován je různý koeficient učení, velikost batch, vliv různých augmentací a metody dropout.

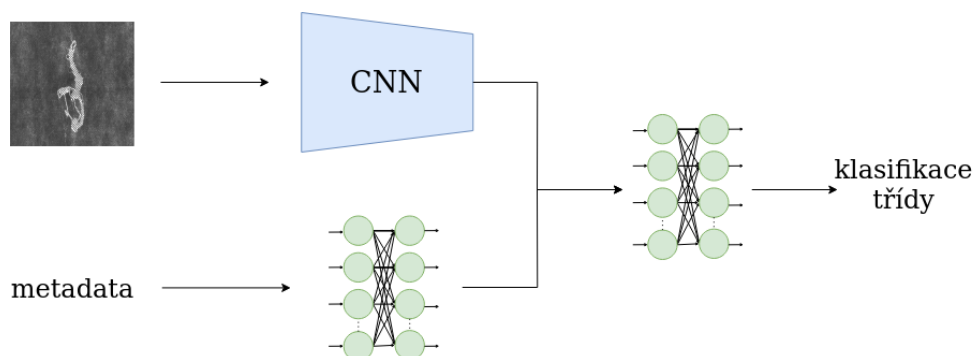
model	velikost (MB)	parametry (miliony)	hloubka
VGG-16	528	138,4	16
Inception-v3	92	23,9	189
ResNet-50	98	25,6	107
DenseNet-121	33	8,1	242
EfficientNet-B0	29	5,3	132

Tabulka 5.1: Parametry vybraných sítí uvedené v dokumentaci knihovny Keras [6].

Architektura sítě s nejvyšší úspěšností je následně použita k vytvoření nového modelu, který na vstupu kromě obrázků přijímá také další data. Konkrétně se jedná o šířku a výšku obrázku před jeho zmenšením a ořezáním. Data s velikostmi snímků jsou uložena v souboru csv, který byl vytvořen pomocným skriptem.

Vytvořeno bylo několik variant tohoto modelu s různým počtem plně propojených vrstev. Nejprve je provedena pouze konkaténace těchto hodnot a příznaků z konvoluční sítě. Takto složený vstup je poslán do plně propojené vrstvy, která provede klasifikaci. Druhou možností je přidat plně propojenou vrstvu ještě před konkaténací vstupů. Další vrstvy lze také připojit před klasifikační vrstvou. Znázornění navržených modelů je na obrázku 5.6. Byly provedeny experimenty s počtem plně propojených vrstev i počtem neuronů.

¹<https://www.image-net.org/>



Obrázek 5.6: Návrh zapojení metadat do architektury CNN modelu.

5.6 Vyhodnocení experimentů

V této části jsou shrnuty výsledky jednotlivých experimentů se vstupními daty a augmentací. Data z pokusů, které přinesly nejlepší výsledky jsou použita pro trénování modelů CNN. Porovnány jsou úspěšnosti klasifikace vybraných modelů konvolučních sítí a nových modelů vytvořených přidáním metadat. Na závěr je provedeno srovnání nejlepší sítě s referenčním řešením.

Pro porovnání pokusů se vstupními daty byla využita předtrénovaná síť DenseNet-121, která byla postupně učena s různým složením vstupních dat. Velikost batch byla 32 a koeficient učení 0,001. Vstup sítě má velikost 224×224 px. Vyhodnocení sítí bylo provedeno metodou 3-fold cross validation. Testovány byly také varianty 5-fold a 10-fold, ale zvolené rozdělení dat v poměru odpovídajícím variantě 3-fold bylo nejvhodnější.

Rozměry obrazu

Protože snímky v datasetu nemají stejnou velikost ani poměr stran, bylo zapotřebí je upravit. Kromě standardního zmenšení obou stran obrazu bylo experimentováno i s dalšími variantami. Použity byly kategorie defektů ze skupiny 12.

V tabulce 5.2 můžeme vidět, že největší zlepšení oproti variantě a), která jen zmenší obě strany obrazu, přinesla varianta b). Zde byla výška obrazu zmenšena na cílovou velikost 224 px a proveden výřez ze středu obrazu také o šířce 224 px. Díky této transformaci byla zachována celá délka defektu, ale ořezány jsou nepotřebné okraje obrazu, které byly ve většině snímcích nadbytečné a obsahovaly pouze textilii bez poškození. Tato úprava obrazu zlepšila přesnost klasifikace o 3,9%.

Experiment	accuracy	precision	recall	f1-score
a) zmenšení obou stran	0,684	0,739	0,735	0,723
b) zmenšení výšky, výřez středu	0,723	0,752	0,767	0,751
c) oříznutí šířky, zmenšení	0,714	0,752	0,761	0,746
d) výřez středu	0,699	0,718	0,735	0,715
e) zmenšení, zachování poměru stran	0,681	0,725	0,730	0,714

Tabulka 5.2: Výsledky experimentů s různou transformací vstupních dat.

Druhou nejlepší úspěšnost má varianta c), u které byla šířka ořezána na polovinu původní velikosti a obraz následně zmenšen. K mírnému zhoršení o 0.3% došlo při zmenšení

obrazu se zachováním poměru stran v experimentu e). U většinu snímků došlo k ořezání části defektu a tedy ztrátě důležitých dat.

Po určení nejlepší transformace vstupního obrazu byl také testován vliv různé interpolace, která je použita při zmenšení. V tabulce 5.3 můžeme vidět, že výsledky tří testovaných interpolací jsou si podobné. Pokus s interpolací metodou nejbližšího souseda měl o 0.2% lepší úspěšnost klasifikace než interpolace bilineární.

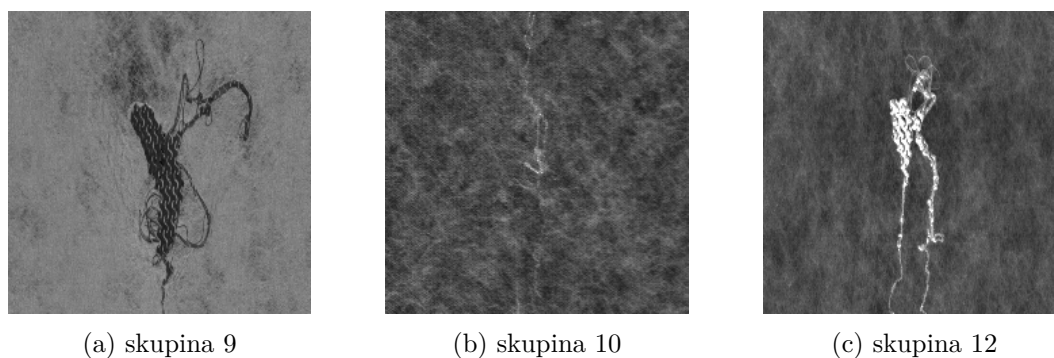
Experiment	accuracy	precision	recall	f1-score
nejbližší soused	0,725	0,757	0,773	0,757
bilineární interpolace	0,723	0,752	0,767	0,751
bikubická interpolace	0,724	0,757	0,772	0,755

Tabulka 5.3: Výsledky experimentů s různou interpolací vstupních dat.

Složení trénovacích dat

Experimenty s různým složením trénovacích dat proběhly s využitím stejné architektury DenseNet-121 jako u testování rozměrů obrazu. Velikost batch je 32 a koeficient učení 0,001. Vstupní obrázky byly zmenšeny na velikost 224×224 pomocí bilineární interpolace. Postupně bylo testováno, jaký vliv bude mít přidání skupin defektů 9 a 10 ke skupině 12, kterou je potřeba klasifikovat.

Prvním pokusem bylo sloučení skupin 12 a 10. Ačkoli má skupina 10 světlé vady na tmavším pozadí podobně jako je tomu ve skupině 12, došlo ke zhoršení přesnosti klasifikace o 2,6%. Naopak po sloučení se skupinou 9 se přesnost zlepšila o 2,1%. Zhoršení se skupinou 10 lze vysvětlit tím, že defekty jsou na těchto snímcích nevýrazné, naopak ve skupině 9 jsou velmi dobře odlišitelné od pozadí, viz 5.7.



Obrázek 5.7: Ukázky úkapu z různých skupin.

Dále bylo provedeno sloučení kategorií *úlet* a *dlouhý tenký úlet*, které obsahují často velmi podobné vady a bylo konzultováno, že je není potřeba rozlišovat. Po sloučení se projevilo očekávané zlepšení klasifikace této třídy. Zároveň tento krok zlepšil klasifikaci (především metriku recall) i důležitých tříd *úkap* a *utržené vlákno*, které byly spolu s třídou *úlet* vzájemně sítí zaměňovány. Po spojení tříd dosahovala síť úspěšnosti 78%. Proveden byl také experiment se spojením tříd *úlet* a zároveň skupiny 9 a 12. Tato kombinace dosahovala úspěšnosti 78,2%.

Po porovnání všech metrik jednotlivých experimentů, viz tabulka 5.4, jsem se rozhodla pro trénování konvolučních sítí použít pouze skupinu 12 se sloučením tříd *úlet* a *dlouhý tenký*

úlet. Ačkoli přidání skupiny 9 přineslo malé zlepšení v metrice accuracy, naopak f1-score kleslo.

Experiment	accuracy	precision	recall	f1-score
12	0,684	0,739	0,735	0,723
12+10	0,658	0,671	0,674	0,657
12+9	0,705	0,750	0,763	0,743
12+9+10	0,676	0,690	0,706	0,679
12+úlet	0,780	0,800	0,799	0,799
12+9+úlet	0,782	0,799	0,801	0,798

Tabulka 5.4: Výsledky experimentů se sloučením různých skupin dat.

Augmentace dat

Vstupní data pro experimenty s augmentací byla upravena podle nejlepších výsledků z předchozích pokusů. Obrázky jsou zmenšeny na výšku 224 px a šířka je ořezána na stejnou velikost. Pro trénování jsou použity defekty skupiny 12, kde byly sloučeny kategorie *úlet* a *dlouhý tenký úlet*.

Testované úpravy obrazu zahrnují změnu jasu, kontrastu, rotaci snímku, zvětšení a jejich kombinace. Transformace, které ovlivňují barvy (odstín, sytost), nejsou testovány, protože nemají u šedotónového obrázku význam. Výsledky některých experimentů ukazuje tabulka 5.5. Lze zhodnotit, že i malá rotace snímků do 5° má negativní vliv na úspěšnost klasifikace. Zhoršení bylo v průměru o 2,5%. Podobný efekt má také přiblížení či oddálení obrazu. Mírné zlepšení (0,3%) se ukázalo po úpravě jasu a kontrastu.

Transformace obrazu byly testovány také na sítích Inception-v3 a VGG-16, kde měly podobný vliv na výslednou klasifikaci. Dá se shrnout, že u všech testovaných sítí měla mírná změna jasu a kontrastu buď pozitivní efekt na úspěšnost, nebo ji téměř nezměnila. Výrazné transformace obrazu naopak způsobily ve všech případech zhoršení úspěšnosti.

Experiment	accuracy	precision	recall	f1-score
bez augmentace	0,801	0,808	0,812	0,809
jas 1	0,802	0,811	0,814	0,812
jas 2	0,804	0,813	0,816	0,814
jas 3	0,802	0,808	0,813	0,810
kontrast 1	0,801	0,813	0,805	0,809
kontrast 2	0,802	0,813	0,806	0,810
kontrast 3	0,802	0,813	0,804	0,808
rotace 1	0,776	0,778	0,802	0,788
rotace 2	0,782	0,787	0,811	0,796
rotace 3	0,772	0,780	0,804	0,787
přiblížení 1	0,789	0,794	0,801	0,797
přiblížení 2	0,790	0,792	0,804	0,797
přiblížení 3	0,787	0,797	0,813	0,802

Tabulka 5.5: Porovnání vlivu některých transformací obrazu na úspěšnost klasifikace sítě DenseNet-121.

Modely CNN a zapojení metadat

Po vyhodnocení experimentů s různou velikostí a složením vstupních dat byly nejlepší varianty použity pro trénování modelů různých konvolučních sítí. Jako vstupní data byly použity kategorie vad ze skupiny 12, kde byl sloučen *úlet* a *dlouhý tenký úlet*. U snímků byla zmenšena výška na velikost 224 px, a ořezána šířka. Pro trénování byly použity kategorie, které měly alespoň 100 snímků.

Základní model VGG-16 měl po učení na nových datech úspěšnost 76,1%. Vliv na zlepšení klasifikace sítě měla především přidaná vrstva dropout před plně propojenou vrstvou, která provádí klasifikaci. Čím větší je míra dropout, tím lepší je úspěšnost sítě. Augmentace vstupních dat také přinesla zvýšení přesnosti asi o 0,3%. Nejlepší úspěšnosti 76,8% bylo dosaženo s mírou augmentace 25% v kombinaci s úpravou jasu a kontrastu vstupních snímků. Model VGG-16 dosahoval z testovaných sítí nejnižší úspěšnosti.

Při trénování modelu Inception-v3 měla augmentace a vrstva dropout také pozitivní vliv na klasifikaci, ačkoli ne tak výrazný jako u sítě VGG-16. Po vyzkoušení různých kombinací bylo dosaženo zlepšení o 0,4% na přesnost 79,4%. Model ResNet-50 naopak dosahoval nejlepších hodnot pouze samotný bez úprav. Přidání míry dropout nebo augmentace zhoršilo výsledky klasifikace. I přesto tato architektura dosahovala jedné z nejvyšších dosažených přesností 81%.

U sítě DenseNet-121 přineslo zlepšení klasifikace snížení koeficientu učení na 0,0009. Metoda dropout s mírou 0.1 také zvedla přesnost sítě o 0,004. Nejvyšší úspěšnosti u této sítě bylo dosaženo kombinací augmentace dat a vrstvy dropout. Výsledná úspěšnost této sítě dosahovala 80,7%. Síť EfficientNet-B0 měla z testovaných modelů nejvyšší úspěšnost 81,1%. Té bylo dosaženo přidáním vrstvy dropout před s mírou 0,1. Augmentace neměla na výsledky příliš velký vliv.

U všech sítí byla testována různá velikost batch a koeficientu učení. Jako nejlepší hodnoty se ukázaly batch 32 a koeficient 0,001. Pouze u sítě DenseNet-121 měl lepší výsledek koeficient učení 0,0009. I přesto, že natrénované modely ResNet-50 a EfficientNet-B0 mají podobnou hodnotu accuracy, EfficientNet-B0 má o 1% vyšší hodnotu f1-score. Proto byl tento model vyhodnocen jako nejúspěšnější.

Model	accuracy	precision	recall	f1-score
VGG-16	0,768	0,778	0,776	0,775
Inception-v3	0,794	0,786	0,796	0,791
ResNet-50	0,810	0,815	0,812	0,813
DenseNet-121	0,807	0,820	0,819	0,819
EfficientNet-B0	0,811	0,822	0,827	0,823

Tabulka 5.6: Výsledky klasifikace vybraných modelů CNN na validační datové sadě.

Architektura EfficientNet-B0 byla následně použita pro vytvoření nového modelu, který na vstupu přijímá i další informace o obrázku. Jeho popis je uveden výše v této kapitole. Testovány byly různé jeho varianty, kdy byl měněn počet a velikost plně propojených vrstev. Jako nejúspěšnější se ukázalo přidání plně propojené vrstvy o velikosti 64 neuronů před klasifikační vrstvou sítě. Takto upravená síť dosahovala úspěšnosti 81,5%, viz tabulka 5.7.

Model	accuracy	precision	recall	f1-score
Efficient+Meta1	0,814	0,831	0,809	0,818
Efficient+Meta2	0,813	0,831	0,807	0,817
Efficient+Meta3	0,806	0,818	0,805	0,808

Tabulka 5.7: Výsledky klasifikace některých variant sítě vytvořené spojením modelu EfficientNet-B0 a vstupu s původní velikostí obrázku.

Ačkoli nově vytvořená architektura byla schopna dosáhnout o 0,2% vyšší hodnoty accuracy než samotný model bez zapojení metadat, byla jako nejúspěšnější vyhodnocena samotná síť EfficientNet-B0. Po zapojení metadat totiž došlo k výraznému snížení f1-score třídy *úkap* o 5,2%, což je nežádoucí. U ostatních tříd se rozdíl v f1-score pohyboval okolo 1%. Výsledky sítě EfficientNet-B0 při vyhodnocení metodou 3-fold cross validation ukazují tabulka 5.8.

Třída	precision	recall	f1-score
91203 úkap	0,799	0,863	0,830
91205 úlet	0,939	0,850	0,892
91206 utržené vlákno	0,839	0,808	0,823
91207 vlašťovka	0,879	0,915	0,897
91208 zesílené místo	0,640	0,734	0,684
91213 prášení	0,836	0,793	0,814
průměr	0,822	0,827	0,823

Tabulka 5.8: Výsledky sítě EfficientNet-B0 při vyhodnocení metodou 3-fold cross validation.

Porovnání s referenčním řešením

Jako nejúspěšnější model byla vyhodnocena natrénovaná síť EfficientNet-B0, která dosahovala accuracy 81,1% na datové sadě o velikosti 3421 snímků. Vyhodnocení proběhlo metodou 3-fold cross validation. Průměrná hodnota f1-score je 82,3%. Následně bylo provedeno otestování na nové datové sadě defektů. Tato sada obsahovala 1059 snímků, které byly rozděleny do jednotlivých tříd ve stejném poměru jako u dříve dostupného datasetu. Klasifikace na nové sadě dat dosahovala accuracy 74,3% a f1-score 72,2%, viz 5.9. Tento pokles lze vysvětlit tím, že data byla získána v jiném prostředí a snímky se mohou lišit od dat, na kterých byla síť trénovaná. V obou případech vyhodnocení však klasifikátor dosahuje větší úspěšnosti než referenční řešení, viz 5.10. Konvoluční síť také klasifikuje lépe všechny tři kritické vady. Jejich průměrné f1-score se zvýšilo o 18,5%. Největšího zlepšení dosáhla třída *úlet*. Pro další zlepšení klasifikace by bylo zapotřebí větší množství správně označených dat k trénování.

Třída	precision	recall	f1-score
91203 úkap	0,486	0,909	0,633
91205 úlet	0,870	0,844	0,857
91206 utržené vlákno	0,539	0,722	0,617
91207 vlašťovka	0,730	0,977	0,836
91208 zesílené místo	0,886	0,432	0,581
91213 prášení	0,893	0,735	0,806
průměr	0,734	0,770	0,722

Tabulka 5.9: Výsledky klasifikace sítě EfficientNet-B0 na nové sadě dat.

Třída	precision	recall	f1-score
91203 úkap	0,371	0,573	0,450
91204 úkap s dírou	0,263	0,179	0,213
91205 úlet	0,494	0,513	0,503
91206 utržené vlákno	0,968	0,435	0,600
91207 vlašťovka	0,970	0,516	0,674
91208 zesílené místo	0,973	0,632	0,767
91210 sklad	0,529	0,692	0,600
91213 prášení	0,792	0,970	0,872
91214 dl. tenký úlet	0,256	0,275	0,265
průměr	0,624	0,532	0,549

Tabulka 5.10: Výsledky klasifikace referenčního řešení.

Kapitola 6

Závěr

Cílem práce bylo implementovat metodu, která klasifikuje defekty netkaných textilií s úspěšností větší než 68,1%. Zároveň bylo požadováno, aby byly úspěšně klasifikovány tři kritické kategorie vad. Toho bylo dosaženo s využitím konvoluční neuronové sítě EfficientNet-B0.

V rámci práce byly nastudovány a shrnuty existující přístupy automatické analýzy obrazu pro detekci a klasifikaci vad v textiliích. Popsán byl také současný kontrolní systém a defekty, které při výrobě netkaných textilií vznikají. Jako klasifikátor byla zvolena a implementována konvoluční síť EfficientNet-B0. Realizována byla série experimentů s cílem zvýšit úspěšnost klasifikace testovaných CNN. Byly provedeny pokusy s velikostí snímku vady a složením vstupních dat. Testován byl také nový model, který jako vstup kombinoval snímek vady a informaci o jeho původní velikosti. Ačkoli model dosahoval mírně vyšší přesnosti, došlo k nežádoucímu zhoršení klasifikace jedné z významných tříd.

Jako nejlepší z testovaných modelů byla vyhodnocena síť EfficientNet-B0, která dosahovala přesnosti 81,1% při vyhodnocení metodou křížové validace s celkovým počtem 3421 snímků. Při testování na novém souboru dat o velikosti 1059 bylo dosaženo úspěšnosti 74,3%. V obou případech tak klasifikace dosahovala větší úspěšnosti než referenční řešení. Klasifikace kritických vad byla zlepšena v průměru o 18,5%. Nejlépe se povedlo klasifikovat kategorii úlet. Vliv na zlepšení klasifikace mělo mimo jiné ořezání šířky snímku.

V práci by bylo možné pokračovat získáním většího množství správně označených dat a navržením nového modelu sítě. Uvedený rozšířený model by bylo možné dále využít v případě, že budou k dispozici lepší metadata. Myslím, že přesnější informace o pozici a velikosti vady by mohly přispět ke zlepšení klasifikace. Toho by šlo dosáhnout vylepšením způsobu detekce vad.

Literatura

- [1] ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Dostupné z: <https://www.tensorflow.org/>.
- [2] AGHDAM, H. H. a HERAVI, E. J. Guide to convolutional neural networks. *New York, NY: Springer*. Springer. 2017, sv. 10, 978-973, s. 51.
- [3] CALDERISI, M., GALATOLO, G., CEPPA, I., MOTTA, T. a VERGENTINI, F. Improve Image Classification Tasks Using Simple Convolutional Architectures with Processed Metadata Injection. In: *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. 2019, s. 223–230. DOI: 10.1109/AIKE.2019.00046. ISBN 978-1-7281-1488-0.
- [4] CHAPMAN, U. *Applications of nonwovens in technical textiles*. 1. vyd. Elsevier, June 15 2010. 224 s. ISBN 9781845699741.
- [5] CHISTI, M. K. M., KUMAR, S. S. a PRASAD, G. Defects Identification, Localization, and Classification Approaches: A Review. *IETE Journal of Research*. Taylor & Francis. 2021, sv. 0, č. 0, s. 1–14. DOI: 10.1080/03772063.2021.1958073. Dostupné z: <https://doi.org/10.1080/03772063.2021.1958073>.
- [6] CHOLLET, F. et al. *Keras: Deep Learning for humans*. 2015. Dostupné z: <https://keras.io>.
- [7] COHEN, F. S., FAN, Z. a ATTALI, S. Automated inspection of textile fabrics using textural models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. IEEE Computer Society. 1991, sv. 13, č. 08, s. 803–808. DOI: 10.1109/34.85670. ISSN 0162-8828.
- [8] GOODFELLOW, I., BENGIO, Y. a COURVILLE, A. *Deep Learning*. MIT Press, 2016. 800 s. ISBN 194872232. <http://www.deeplearningbook.org>.
- [9] GÜLER, H. a GÜNEŞ, M. Performance comparison of spectral based approaches on fabric defect detection. *Global Journal on Technology*. 2015, č. 7, s. 42–57.
- [10] GURNEY, K. *An introduction to neural networks*. London and New York: Taylor & Francis e-Library, 2017. 234 s. ISBN 9781315273570.
- [11] HAJIMOWLANA, S., MUSCEDERE, R., JULLIEN, G. a ROBERTS, J. 1D autoregressive modeling for defect detection in web inspection systems. In: *1998 Midwest Symposium on Circuits and Systems (Cat. No. 98CB36268)*. IEEE, 1998, s. 318–321. DOI: 10.1109/MWSCAS.1998.759496. ISBN 0-8186-8914-5.

- [12] HANBAY, K., TALU, M. F. a FARUK ÖZGÜVEN Ömer. Fabric defect detection systems and methods—A systematic literature review. *Optik*. 2016, sv. 127, č. 24, s. 11960–11973. DOI: <https://doi.org/10.1016/j.ijleo.2016.09.110>. ISSN 0030-4026. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0030402616311366>.
- [13] HE, K., ZHANG, X., REN, S. a SUN, J. Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. ArXiv, June 2016. DOI: <https://doi.org/10.48550/arXiv.1512.03385>.
- [14] HU, G.-H. Optimal ring Gabor filter design for texture defect detection using a simulated annealing algorithm. In: *2014 International Conference on Information Science, Electronics and Electrical Engineering*. 2014, sv. 2, s. 860–864. DOI: 10.1109/InfoSEEE.2014.6947789. ISBN 978-1-4799-8164-9.
- [15] HUANG, G., LIU, Z., MAATEN, L. van der a WEINBERGER, K. Q. Densely Connected Convolutional Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [16] JIA, Y., SHELFHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J. et al. Caffe: Convolutional Architecture for Fast Feature Embedding. *ArXiv preprint arXiv:1408.5093*. 2014.
- [17] JING, J., WANG, Z., RÄTSCH, M. a ZHANG, H. Mobile-Unet: An efficient convolutional neural network for fabric defect detection. *Textile Research Journal*. 2020, sv. 0, č. 0, s. 0040517520928604. DOI: 10.1177/0040517520928604. Dostupné z: <https://doi.org/10.1177/0040517520928604>.
- [18] JING, J., ZHANG, H., WANG, J., LI, P. a JIA, J. Fabric defect detection using Gabor filters and defect classification based on LBP and Tamura method. *The Journal of The Textile Institute*. Taylor & Francis. 2013, sv. 104, č. 1, s. 18–27. DOI: 10.1080/00405000.2012.692940. Dostupné z: <https://doi.org/10.1080/00405000.2012.692940>.
- [19] KAEHLER, A. a BRADSKI, G. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. 1. vyd. Sebastopol: O’Reilly, 2016. ISBN 978-1491937990.
- [20] KAI LING MAK, K.-F. C. Y. Fabric defect detection using multi-level tuned-matched Gabor filters. *Journal of Industrial & Management Optimization*. 2012, sv. 8, č. 2, s. 325–341.
- [21] KRIZHEVSKY, A., SUTSKEVER, I. a HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012, sv. 25, s. 1097–1105.
- [22] KUMAR, A. Computer-Vision-Based Fabric Defect Detection: A Survey. *IEEE Transactions on Industrial Electronics*. 2008, sv. 55, č. 1, s. 348–363. DOI: 10.1109/TIE.1930.896476.
- [23] LI, F.-F. *CS231N convolutional neural networks for visual recognition*. Dostupné z: <https://cs231n.github.io/>.

- [24] LI, W. a CHENG, L. Yarn-dyed woven defect characterization and classification using combined features and support vector machine. *The Journal of The Textile Institute*. Taylor & Francis. 2014, sv. 105, č. 2, s. 163–174. DOI: 10.1080/00405000.2013.833689. Dostupné z: <https://doi.org/10.1080/00405000.2013.833689>.
- [25] LI, Y., ZHANG, D. a LEE, D.-J. Automatic fabric defect detection with a wide-and-compact network. *Neurocomputing*. 2019, sv. 329, s. 329–338. DOI: <https://doi.org/10.1016/j.neucom.2018.10.070>. ISSN 0925-2312. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0925231218313109>.
- [26] MOTTALIB, M. M., ROKONUZZAMAN, M., HABIB, M. T. a AHMED, F. Fabric defect classification with geometric features using Bayesian classifier. In: *2015 International Conference on Advances in Electrical Engineering (ICAEE)*. IEEE Computer Society, 2015, s. 137–140. DOI: 10.1109/ICAEE.2015.7506815. ISBN 978-1-4673-9695-0.
- [27] NGAN, H. Y., PANG, G. K. a YUNG, N. H. Automated fabric defect detection—A review. *Image and Vision Computing*. 2011, sv. 29, č. 7, s. 442–458. DOI: <https://doi.org/10.1016/j.imavis.2011.02.002>. ISSN 0262-8856. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0262885611000230>.
- [28] NIELSEN, M. A. *Neural networks and deep learning*. Determination press San Francisco, CA, 2015. Dostupné z: <https://books.google.cz/books?id=STDBswEACAAJ>.
- [29] OJALA, T., PIETIKAINEN, M. a MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Institute of Electrical and Electronics Engineers (IEEE). 2002, sv. 24, č. 7, s. 971–987. DOI: 10.1109/TPAMI.2002.1017623. ISSN 0162-8828.
- [30] OPENCV. *Open Source Computer Vision Library*. 2015. Dostupné z: <http://opencv.org/>.
- [31] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: WALLACH, H., LAROCHELLE, H., BEYGEZIMER, A., ALCHÉ BUC, F. d', FOX, E. et al., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019, sv. 32, s. 8026–8037. Dostupné z: <https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>.
- [32] PATIL, M., VERMA, S. a WAKODE, J. A review on fabric defect detection techniques. *Int. Res. J. Eng. Technol.* 2017, sv. 4, č. 9, s. 131–136.
- [33] RAMOLA, A., SHAKYA, A. K. a VAN PHAM, D. Study of statistical methods for texture analysis and their modern evolutions. *Engineering Reports*. Wiley. 2020, sv. 2, č. 4, s. 24. DOI: <https://doi.org/10.1002/eng2.12149>. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/eng2.12149>.
- [34] SAKHARE, K., KULKARNI, A., KUMBHAKARN, M. a KARE, N. Spectral and spatial domain approach for fabric defect detection and classification. 2015, s. 640–644. DOI: 10.1109/IIC.2015.7150820.

- [35] SHADY, E., GOWAYED, Y., ABOUHANNA, M., YOUSSEF, S. a PASTORE, C. Detection and Classification of Defects in Knitted Fabric Structures. *Textile Research Journal*. 2006, sv. 76, č. 4, s. 295–300. DOI: 10.1177/0040517506053906. Dostupné z: <https://doi.org/10.1177/0040517506053906>.
- [36] SIMONYAN, K. a ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv 1409.1556*. arXiv. Zář 2014. DOI: <https://doi.org/10.48550/arXiv.1409.1556>.
- [37] SOKOLOVA, M. a LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*. 2009, sv. 45, č. 4, s. 427–437. DOI: <https://doi.org/10.1016/j.ipm.2009.03.002>. ISSN 0306-4573. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0306457309000259>.
- [38] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I. a SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*. JMLR. org. 2014, sv. 15, č. 1, s. 1929–1958.
- [39] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S. et al. Going Deeper With Convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2015. DOI: <https://doi.org/10.1109/CVPR.2015.7298594>. ISBN 978-1-4673-6964-0.
- [40] TAMURA, H., MORI, S. a YAMAWAKI, T. Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics*. 1978, sv. 8, č. 6, s. 460–473. DOI: 10.1109/TSMC.1978.4309999.
- [41] TAN, M. a LE, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv. 2019. DOI: 10.48550/ARXIV.1905.11946. Dostupné z: <https://arxiv.org/abs/1905.11946>.
- [42] THEODORIDIS, S. Chapter 18 - Neural Networks and Deep Learning. In: THEODORIDIS, S., ed. *Machine Learning (Second Edition)*. Second Edition. Academic Press, 2020, s. 901–1038. DOI: <https://doi.org/10.1016/B978-0-12-818803-3.00030-1>. ISBN 978-0-12-818803-3. Dostupné z: <https://www.sciencedirect.com/science/article/pii/B9780128188033000301>.
- [43] TONG, L., WONG, W. a KWONG, C. Differential evolution-based optimal Gabor filter model for fabric inspection. *Neurocomputing*. 2016, sv. 173, s. 1386–1401. DOI: <https://doi.org/10.1016/j.neucom.2015.09.011>. ISSN 0925-2312. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0925231215013193>.
- [44] VOMELA, M. a HONEC, P. *Continuous Strip Inspection - SURFSCAN*. Brno: CAMEA, spol. s r.o. and Brno University of Technology, FEEC, 2016.
- [45] WEI, B., HAO, K., TANG, X. a DING, Y. A new method using the convolutional neural network with compressive sensing for fabric defect classification based on small sample sizes. *Textile Research Journal*. 2019, sv. 89, č. 17, s. 3539–3555. DOI: 10.1177/0040517518813656. Dostupné z: <https://doi.org/10.1177/0040517518813656>.

- [46] ZHANG, Y., LU, Z. a LI, J. Fabric Defect Detection and Classification Using Gabor Filters and Gaussian Mixture Model. In: ZHA, H., TANIGUCHI, R.-i. a MAYBANK, S., ed. *Computer Vision – ACCV 2009*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, s. 635–644. ISBN 978-3-642-12304-7.
- [47] ZHAO, Y., HAO, K., HE, H., TANG, X. a WEI, B. A visual long-short-term memory based integrated CNN model for fabric defect image classification. *Neurocomputing*. 2020, sv. 380, s. 259–270. DOI: <https://doi.org/10.1016/j.neucom.2019.10.067>. ISSN 0925-2312. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S092523121931505X>.
- [48] ÇELİK, H., DÜLGER, L. a TOPALBEKIROĞLU, M. Development of a machine vision system: real-time fabric defect detection and classification with neural networks. *The Journal of The Textile Institute*. Taylor & Francis. 2014, sv. 105, č. 6, s. 575–585. DOI: 10.1080/00405000.2013.827393. Dostupné z: <https://doi.org/10.1080/00405000.2013.827393>.
- [49] ČÍP, P. *PFN – Rozhodovací strom klasifikátoru defektů*. Brno: CAMEA, spol. s r.o., 2020.