

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informatiky a kvantitativních metod**

**Počítačové vidění v průmyslové výrobě**  
**Bakalářská práce**

Autor: Josef Málek

Studijní obor: aplikovaná informatika, kombinovaná forma

Vedoucí práce: Mgr. Jan Vaněk, Ph.D.

Katedra informatiky a kvantitativních metod

Hradec Králové

listopad 2017

**Prohlášení:**

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 15. 11. 2017

Josef Málek

**Poděkování:**

Děkuji panu Mgr. Janu Vaňkovi, Ph.D. za odborné vedení, vstřícnost a cenné rady při vypracování bakalářské práce.

**Anotace:**

Bakalářská práce popisuje postupy a metody počítačového vidění a zaměřuje se na oblast průmyslové výroby a využití strojového vidění při kontrole kvality. Snaží se nalézt odpověď na otázku, zda by některé z metod byly použitelné pro automatickou kontrolu při výrobě tištěných topných těles. Pro klasifikaci vadných a bezvadných výrobků je využito algoritmů strojového učení, konkrétně stromů a neuronových sítí.

**Klíčová slova:**

počítačové vidění, zpracování obrazu, kontrola kvality, strojové učení, tištěná topná tělesa

**Annotation:**

**Title:** Computer Vision in Industrial Production

The bachelor thesis describes computer vision methods and algorithms and focuses on the area of industrial production and the use of machine vision in quality control. It is trying to find the answer to the question whether some of the methods could be useful for automatic control in the production of thick film heating elements. The algorithms of machine learning, namely trees and neural networks, are used to classify defective and faultless products.

**Keywords:**

computer vision, image processing, quality control, machine learning, thick film heating elements

# OBSAH

<b>1</b>	<b>Úvod</b> .....	<b>1</b>
<b>2</b>	<b>Počítačové vidění</b> .....	<b>2</b>
2.1	<i>Historie</i> .....	3
2.2	<i>Obraz</i> .....	4
2.3	<i>Obecný postup zpracování obrazu</i> .....	5
2.3.1	Snímání obrazu a digitalizace .....	5
2.3.2	Předzpracování obrazu .....	7
2.3.3	Segmentace obrazu na objekty.....	13
2.3.4	Popis objektů .....	16
2.3.5	Porozumění obsahu obrazu, klasifikace objektů .....	17
2.4	<i>Strojové učení</i> .....	17
2.5	<i>Algoritmy strojového učení</i> .....	19
2.5.1	Stromy.....	19
2.5.2	Neuronové sítě .....	23
2.6	<i>Senzitivita, specificita, <math>F_1</math> skóre a přesnost</i> .....	25
2.7	<i>Typické úlohy strojového vidění</i> .....	27
2.7.1	Odhalování povrchových vad .....	28
2.7.2	Identifikace, vyhledání polohy a počítání kusů.....	28
2.7.3	Metrologické úlohy a kontrola tolerancí.....	28
2.7.4	Rozpoznávání barev .....	29
2.7.5	Načítání a ověřování textů a kódů.....	29
2.8	<i>Praktické využití počítačového vidění</i> .....	29
2.8.1	Kontrola elektrických kontaktů.....	29
2.8.2	Detekce globálních a lokálních vad pružných tištěných obvodů.....	30
2.9	<i>Knihovna OpenCV</i> .....	31
<b>3</b>	<b>Navržené řešení</b> .....	<b>32</b>
3.1	<i>Tištěná topná tělesa</i> .....	32
3.2	<i>Výroba a kontrola kvality</i> .....	33
3.3	<i>Fotodokumentace</i> .....	34
<b>4</b>	<b>Průběh a výsledky výzkumu</b> .....	<b>36</b>
4.1	<i>Sběr informací</i> .....	36

4.2	<i>Aplikace pro zpracování obrazu.....</i>	<i>37</i>
4.3	<i>Testování datových sad.....</i>	<i>39</i>
4.4	<i>Shrnutí výsledků a další rozvoj aplikace.....</i>	<i>44</i>
<b>5</b>	<b>Návrhy a doporučení .....</b>	<b>48</b>
<b>6</b>	<b>Závěr.....</b>	<b>49</b>
<b>7</b>	<b>Seznam použité literatury.....</b>	<b>50</b>
<b>8</b>	<b>Seznam obrázků.....</b>	<b>53</b>
<b>9</b>	<b>Seznam tabulek.....</b>	<b>54</b>
<b>10</b>	<b>Přílohy.....</b>	<b>55</b>

# 1 ÚVOD

Bakalářská práce se zabývá problematikou kontroly kvality při výrobě tištěných topných těles. V dnešní době jsou kladeny vysoké požadavky odběratelů na precizní a bezchybné zhotovení odebíraného zboží. Nezřídka je požadována 100% kvalita, a proto každý kus musí být při výrobě kontrolován. Společnost Backer Elektro a. s. začala před několika lety vyrábět tištěná topná tělesa. Výroba se nyní rozrostla do takové míry, že byla zřízena zcela nová výrobní linka a další je ve stádiu příprav.

Kontrola výrobků nyní probíhá čistě manuálně. Kontroloři kvality prohlížejí každý výrobek a sledují, zda neobsahuje nějakou výrobní vadu. Pokud je shledán bezvadným, pokračuje dále výrobním procesem. Osobní kontrola zaměstnancem má svá neoddiskutovatelná negativa, ať se jedná o nedbalost, únavu, subjektivní posouzení apod. Již nějakou dobu se uvažuje o automatizaci kontroly pomocí techniky. Cílem bakalářské práce je zjistit, zda by se metody strojového vidění daly reálně a efektivně použít při této činnosti.

## 2 POČÍTAČOVÉ VIDĚNÍ

Počítačové vidění (*computer vision*) je relativně nové odvětví výpočetní techniky a vývoje softwaru, které se zabývá vytvářením systémů schopných získávat informace ze snímaného obrazu. Počítačové vidění je v určitém směru opakem počítačové vizualizace. Zatímco počítačová vizualizace vytváří obrazová data z informací popisujících zachycené objekty, počítačové vidění je opačný proces. Vyskytuje se trend obě disciplíny kombinovat, například v systémech rozšířené reality. Počítačové vidění zažívá v současné době obrovský rozmach díky zpřístupnění a zlevnění moderních technologií. Autonomní vozidla, která jsme mohli vídat ve vědecko-fantastických filmech na začátku století, získávají v současnosti reálné obrysy. Počítačové vidění se využívá i v medicíně, kdy ze získaných informací z mikroskopů, RTG nebo tomografie lze určovat diagnózu pacienta. Využití v armádě je pravděpodobně jednou z největších oblastí využití počítačového vidění a pokrok v této oblasti může být mnohem dál, než je obecně prezentováno. Zde se setkáme se systémy detekce nepřátelských vojáků a vozidel, systémy navádění raket, řízení bezpilotních vozidel, letounů a dronů či analýze družicových snímků.

Bakalářská práce pojednává o využití počítačového vidění v průmyslu. Zde je nespočet možností, kde počítačové vidění (v průmyslu častěji používané spojení strojové vidění) nalezne své uplatnění, ať se jedná o úlohy vizuální inspekce předepsaných viditelných parametrů, počítání objektů nebo hledání defektů a vad. Strojové vidění ideálně podporuje dvě základní a nevyhnutelné konkurenční strategie každého výrobního podniku, snižování nákladů a zvyšování kvality. Firma pak může svým zákazníkům nabídnout vyšší kvalitu za nižší cenu. Příkladem toho, jak tento přínos funguje, je automobilový průmysl. V případě zjištění jediného vadného dílu dnes automobilka vrací celou dodávku a často i uplatňuje vysoké smluvní penále. Jedinou možností, jak se vyhnout těmto ztrátám, je kontrolovat každou provedenou operaci a každý vyrobený kus. A prakticky jediným způsobem, jak to udělat bez neúměrného růstu nákladů, jsou právě systémy strojového vidění.

Počítačové vidění se rozvíjí již od počátku výpočetní techniky, významnějších pokroků je však dosahováno až od devadesátých let minulého století (podrobněji viz kapitola 2.1. Základem pro všechny metody je reprezentace obrazu ve formě funkce prostorových souřadnic, viz kapitola 2.2. Samotné zpracování obrazu je tvořeno



posloupností navazujících kroků, které jsou blíže popsány v kapitole 2.3. Společně s počítačovým viděním se rozvíjí obor strojového učení (viz kapitola 2.4) a díky propojení obou vědních disciplín je dosahováno velice zajímavých výsledků. Základem strojového učení je řada rozhodovacích a učících algoritmů, mezi které patří stromy a neuronové sítě, o kterých pojednává kapitola 2.5. Úspěšnost učení je posuzována za pomoci numerických metrik, z nichž ty relevantní pro tuto práci jsou uvedeny v kapitole 2.6. Počítačové vidění lze využít v mnoha typických úlohách průmyslové výroby obecně popsaných v kapitole 2.7 a na konkrétních příkladech v kapitole 2.8. Při implementaci vlastních systémů počítačového vidění je velmi často využívána knihovna OpenCV, volně dostupná pro řadu programovacích jazyků (viz kapitola 2.9).

## 2.1 Historie

Jedním z prvních pokusů řešení problému počítačového vidění byl projekt Saymoura Paperta z MIT publikovaný v červenci 1966. Hlavním cílem studie, příhodně nazvané The Summer Vision Project, bylo vytvořit systém, který bude schopen v obraze rozpoznat oblasti objektu, oblasti pozadí a oblasti chaosu a následně rozpoznané objekty správně pojmenovat (1). Projekt však nebyl tak úspěšný, jak se zamýšlelo, neboť počítačové vidění je velmi složitá problematika, což dokládá už jen fakt, že lidský mozek spotřebuje až 70% svojí kapacity na zpracování obrazových signálů (2). Po prvních neúspěších se výzkumníci zaměřili na úlohy zpracování obrazu. Zpracování obrazu zahrnovalo operace na úrovni pixelů, řešila se detekce hran objektů, vznikaly různé filtry na zvýšení kvality obrazu, řešila se komprese obrazu, aniž by docházelo ke ztrátě informace. Stále ale nebyla vyřešena základní myšlenka počítačového vidění, a to interpretovat vizuální svět. Na začátku devadesátých let se objevil systém konvolučních neuronových sítí, který byl schopen rozpoznávat čísla na bankovních šecích (3).

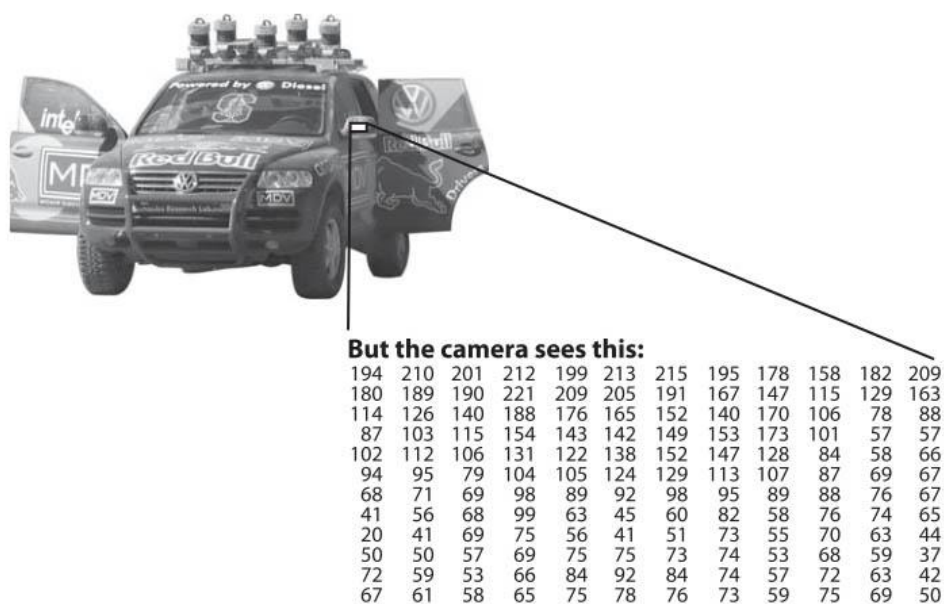
Od devadesátých let dvacátého století zažíval obor počítačového vidění zrychlený rozvoj. Objevily se způsoby, jak řešit problematiku klasifikace objektu, detekce a segmentace objektu či rozpoznávání obličejů. Jedním z prvních úspěchů bylo představení algoritmu SIFT (4) pro detekci a popis místních specifik v obrazech. Z tréninkových obrazů jsou získány zajímavé body pro popis objektu a tato data jsou použita na vyhledávání objektu v testovaném obraze. Společně s metodou Bag of

Words (5), která shromažďuje získaná data, nyní tvoří celkem přesné systémy pro vyhledávání objektů. Velmi důležité bylo představení silného učícího algoritmu, tzv. podpůrného vektorového stroje (SVM) (6), a zásadní pokroky v oblasti neuronových sítí.

Významným impulsem pro vývoj je systém otevřených veřejných výzev, kde se každý může účastnit řešení vyvěšených problémů, viz (7). A v neposlední řadě je to současný vývoj internetu a sociálních sítí, který slouží jako nekonečný zdroj informací a tréninkových dat a dostupnost a výkon současné výpočetní techniky.

## 2.2 Obraz

Dvourozměrný obraz je popsán obrazovou funkcí  $f(x, y)$ , jejímiž argumenty jsou dvě souřadnice v rovině. Takto lze popsat monochromatický obraz. V případě barevného obrazu každé dvojici souřadnic  $(x, y)$  odpovídá vektor hodnot, většinou hodnot jasu červené, zelené a modré složky obrazu (RGB model). Při zpracování počítačem se pak pracuje s diskretizovanými digitálními obrazovými funkcemi, které jsou reprezentované dvourozměrnými maticemi, viz Obr. 1. Pro člověka a lidský mozek je zpracování obrazu díky mnoha létům zkušeností zakódovaných ve složitých strukturách mozku rutinní záležitostí, v systémech počítačového vidění ale počítač obdrží řadu čísel, kde neexistuje nějaký vzorec pro rozpoznání ani pravidla, jak z těchto čísel zjistit, co skutečně znamenají.

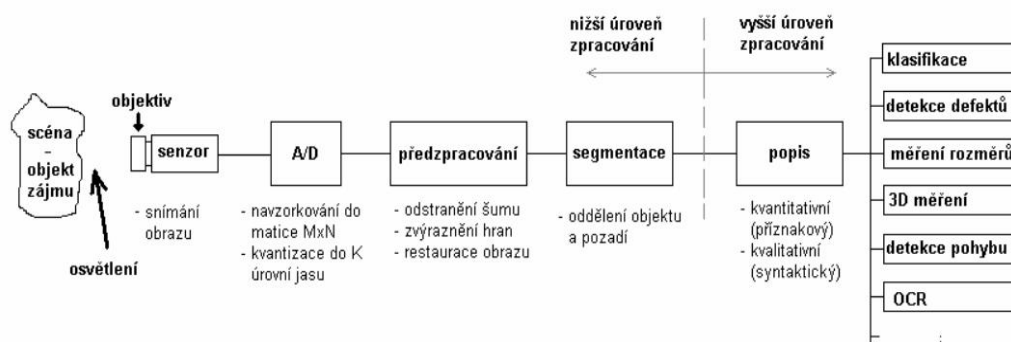


Obr. 1 – Obraz jako matice čísel, převzato z (8)

A to je jeden z důvodů, proč je počítačové vidění tak složité, protože neexistuje spolehlivá cesta, jak z dvourozměrného obrazu snadno rekonstruovat trojrozměrný svět. Obraz může reprezentovat libovolnou kombinaci zobrazení stejného objektu, protože výslednou podobu ovlivňuje řada faktorů, které způsobují šum a zkreslení. Může se jednat o rozdíly ovlivněné prostředím (počasí, osvětlení, odrazy, pohyb), nedokonalosti v objektivu, různé nastavení objektivu, elektrický šum na senzoru nebo kompresní chyby vznikající po zachycení a při ukládání obrazu (8).

## 2.3 Obecný postup zpracování obrazu

Vlastní průběh zpracování a rozpoznávání obrazu reálného světa se obvykle skládá z několika základních kroků. Jednotlivé kroky jsou znázorněny na Obr. 2.

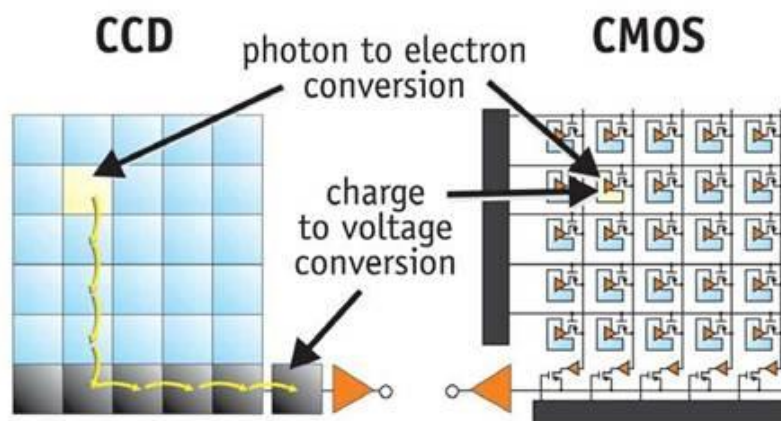


Obr. 2 – Sekvence úkonů zpracování obrazu, převzato z (9)

### 2.3.1 Snímání obrazu a digitalizace

Celý proces zpracování a rozpoznávání obrazu začíná zachycením obrazu skutečného světa, následuje převod snímku do digitální podoby vhodné pro uložení a následné zpracování v počítači či jiném systému. Při snímání obrazu dochází k převodu optické veličiny na elektrický signál, který je spojený v čase i úrovni. Pro snímání obrazu je většinou využívána kamera nebo fotoaparát, která obsahuje obrazový senzor. Zjednodušeně si lze obrazový senzor přestavit jako matici uspořádaných mikroskopických jamek, které zachycují fotony světla. V současnosti se většinou setkáme se senzory CCD nebo CMOS. Princip CCD (Charge – Coupled Device) čipu je založen na dvourozměrné matici senzorů a každý senzor obsahuje prvek citlivý na světlo, tzv. fotodiodu, která dopadající proud fotonů přeměňuje na elektrický signál. Elektronika v CCD senzoru zajišťuje následné postupné vyčítání obsahu senzoru. Ze

senzoru vystupuje analogový elektrický signál, který je dále digitalizován pomocí AD převodníku a zpracován logickými obvody kamery nebo jiného zařízení. CMOS (Complementary Metal Oxide Semiconductor) je obrazový senzor, který je postaven na podobném principu jako CCD čip. Jedná se o dvourozměrné pole světlocitlivých prvků – fotodiod, kde opět dochází k převodu světelné energie na elektrickou. Elektronika je součástí každé buňky s fotodiodou, která určuje aktuální hodnotu osvětlení. Každá jednotlivá buňka má oproti CCD čipu své vlastní připojení, jak je názorně zobrazeno v Obr. 3. Dochází k přímému zesílení signálu z každé buňky, což je paradoxně konstrukčně mnohem jednodušší. Navíc se zde také objevuje menší počet nežádoucích artefaktů, jako je například šum. Z výrobního hlediska jsou technologické procesy podobné procesům při výrobě elektrotechnických součástek, jako jsou posuvné registry apod., které celý proces výroby velmi usnadňují. I když se zdá technologie CMOS čipů mnohem složitější, není výroba nijak nákladná. Ačkoli je technologie CMOS senzorů o více než dvě desetiletí mladší, nelze ji kategoricky považovat za dokonalejší. Oba principy jsou dle autorů v (10) považovány za dobré.



Obr. 3 – CCD vs. CMOS, převzato z (11)

Druhou fází procesu získávání obrazu vhodného pro další zpracování v počítačích je převedení spojitého analogového signálu na signál digitální. Tento proces se nazývá digitalizace. Digitální obraz je diskretizovaným ekvivalentem spojitě obrazové funkce  $f(x, y)$ , kde  $x$  a  $y$  jsou souřadnice v prostoru. Obraz se získá pomocí vzorkování do matice  $M \times N$  bodů a kvantováním do  $K$  úrovní. Vzorkování se řídí obecně známým Shannonovým (Nyquistovým) teorémem. „Pro jednorozměrné signály musí být vzorkovací frekvence minimálně dvakrát větší, než nejvyšší frekvence ve vzorkovaném signálu. Pro dvojrozměrné signály je zapotřebí zvolit

*interval vzorkování tak, aby byl minimálně dvakrát menší než velikost nejmenšího detailu v obraze.*“ (12) Pro dosažení optimálního zpracování obrazového signálu se většinou používá velikost vzorkovacího elementu pětikrát menší, než je teoretická mez daná vzorkovací větou. Právě volba vhodného rozlišení obrazu je jedním ze zásadních kroků digitalizace. Pokud je zvoleno nízké rozlišení, dojde ke ztrátě informací o detailech v obraze, a naopak při velkém rozlišení bude docházet k neúměrné výpočetní náročnosti při dalším zpracovávání obrazu. Velikost obrazu je většinou udávána v obrazových bodech – pixelech (picture element). Pro označení reálné velikosti obrazu se využívá jednotka bodů na palec – DPI (dots per inch). Při kvantování úrovní jasu je volen jejich počet tak, aby nevznikaly falešné obrysy v obraze. Ty začínají být člověku patrné, pokud počet úrovní klesne pod padesát (13). Další důležitou součástí digitalizace bývá volba vzorkovací mřížky. Většinou jsou používány mřížky čtvercové nebo hexagonální. Čtvercová vzorkovací mřížka vychází z konstrukce většiny snímacích prvků a proto je velmi snadno realizovatelná. Čtvercová mřížka má však i své nevýhody, které se týkají především měření vzdáleností a spojitosti objektů. Hexagonální mřížka většinu těchto problémů řeší, ale není zase vhodná pro některé operace, jakými jsou např. Fourierova transformace, jak uvádějí autoři v (10). Při řešení bude využita pouze čtvercová vzorkovací mřížka.

### **2.3.2 Předzpracování obrazu**

Předzpracování obrazu si klade za cíl získat z původního obrazu obraz vhodnější pro další zpracování. Metody předzpracování obrazu je možné rozdělit do několika základních skupin, kterými jsou bodové jasové transformace, geometrické transformace, či metody lokálního předzpracování. Používají se operace jako potlačení šumu, odstranění zkreslení, dochází ke zvýraznění nebo potlačení některých prvků v obraze. Z hlediska Shannonova teoremu není možné v průběhu předzpracování získat žádnou novou informaci. Skutečně lze pouze některou z obrazových informací potlačit a jinou zvýraznit. Úpravy, které jsou využívány během předzpracování a jejich výsledek, je podmíněn naší znalostí základních charakteristik obrazu. Pokud je celkový informační obsah obrazu nízký, nezbyvá než zlepšit postup použitý při pořizování, protože kvalita zachyceného obrazu je klíčem k vytvoření fungujících systémů počítačového vidění.

## Bodové jasové transformace

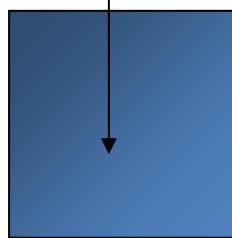
Bodové jasové transformace je možné rozdělit do dvou skupin. První skupinou jsou jasové korekce, druhou modifikace jasové stupnice. V případě jasových korekcí je jas v bodě výstupního obrazu závislý pouze na jasů bodu ve vstupním obraze s totožnými souřadnicemi. Jasová korekce je využívána při poruchách hardwaru, kdy jednotlivé světlocitlivé prvky snímače mohou vykazovat jinou citlivost, nebo při nerovnoměrném osvětlení při snímání. Výpočet je definován rovnicí (2.1). Při korekci je třeba určit degradační funkci a předpokládat multiplikatívni charakter poruchy.

$$f(x, y) = e(x, y) * g(x, y) \quad (2.1)$$

$f(i, j)$	...	<i>obraz po korekci</i>
$e(i, j)$	...	<i>degradační funkce</i>
$g(i, j)$	...	<i>původní obraz</i>

Samotná degradační funkce modeluje poruchu a bývá stanovena buď analyticky, například aproximací poruchy analytickou plochou, nebo empiricky pomocí etalonového snímku. Analytický model poruchy je dán obecně nelineární rovnicí určenou z apriorní explicitní informace. Příklad analytického modelu degradační funkce může být korekce nerovnoměrného osvětlení, kdy scéna je osvětlena z levého horního rohu a korekce je tedy opačná (9), viz rovnice (2.2).

$$e(i, j) = 0.0005(x + y)^2 + 0.4x + 16 \quad (2.2)$$



Pro empirický model poruchy, pokud je definovaná scéna, pořídíme snímek  $g(x, y)$  při explicitně určené scéně  $g'(x, y)$ . Degradační funkce je pak určena rovnicí (2.3):

$$e(x, y) = \frac{g(x, y)}{g'(x, y)} \quad (2.3)$$

Pokud není k dispozici přesně definovaná scéna, je možné pořídit prázdný snímek bez objektu  $g_0(x, y)$  a snímek za tmy se zakrytým objektivem  $g_B(x, y)$ . Degradáční funkce je pak definována rovnicí (2.4):

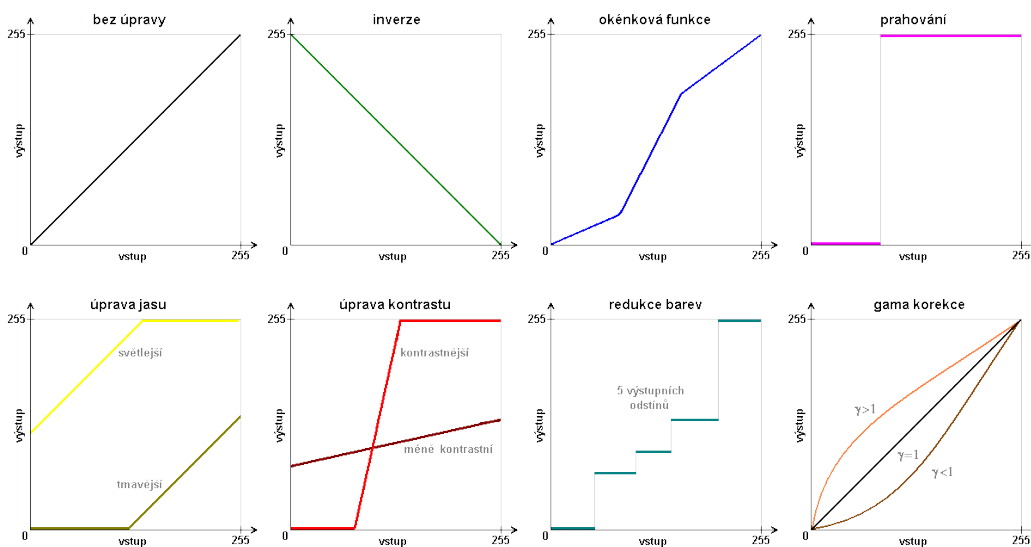
$$e(x, y) = \frac{1}{g_0(x, y) - g_B(x, y)} \quad (2.4)$$

a výpočet korigovaného snímku je modifikován, viz (2.5):

$$f(x, y) = e(x, y) * g(x, y) - g_B(x, y) \quad (2.5)$$

### Modifikace jasové stupnice

U modifikací jasové stupnice bývá pouze některá hodnota jasu ve vstupním obraze přetransformována na jinou hodnotu ve výstupním obraze, a to bez ohledu na pozici v obraze. Příklady modifikací jasové stupnice ilustrují grafy v Obr. 4.



Obr. 4 – Modifikace jasové stupnice, převzato z (9)

Zajímavou transformací jasové stupnice je ekvalizace histogramu. „Histogram je grafické znázornění distribuce dat pomocí sloupcového grafu se sloupci stejné šířky, vyjadřující šířku intervalů (tříd), přičemž výška sloupců vyjadřuje četnost sledované veličiny v daném intervalu.“ (14) Pokud budeme uvažovat o obraze ve stupních šedi, na ose x budou znázorněny stupně intenzity (0 – 255), na ose y počet pixelů odpovídajících danému stupni šedi. Algoritmus ekvalizace mění rozložení intenzit v obraze tak, aby se v něm vyskytovaly intenzity přibližně se stejnou četností. Metoda je dobře použitelná pro obrazy, které jsou buď příliš tmavé, nebo naopak

příliš světlé. Dochází k lokálnímu zvýšení kontrastu a ke zvýraznění detailů, jak uvádějí autoři v (10).

### **Geometrické transformace**

Jak je z názvu patrné, geometrické transformace slouží k odstranění geometrických zkreslení vzniklých při pořízení obrazu nebo ke standardním operacím, jakými jsou změna rozlišení obrazu, posunutí, otočení, či různé deformace. Transformaci je možné opět rozdělit do dvou kroků, transformaci souřadnic bodů a následnou interpolaci jasu několika sousedních bodů. Geometrické transformace obecně řeší problém, kdy jsou diskrétním celočíselným souřadnicím ve vstupním obraze přiřazovány celočíselné souřadnice v obraze výstupním tak, aby nevznikaly díry, nebo aby docházelo ke správnému mapování několika pixelů na stejné místo. Obecně využívanými metodami bývají interpolace metodou nejbližšího souseda, či váženým průměrem čtyř nebo více nejbližších sousedů, nebo bilineární, či bikubická interpolace, jak je uvedeno v (10).

### **Lokální předzpracování**

V metodách lokálního předzpracování je pro výpočet jasu pixelu ve výstupním obraze využíváno pouze lokálního okolí odpovídajícího pixelu ve vstupním obraze. Některé z operací lokálního předzpracování se také nazývají filtrace. Při lokálním předzpracování se lze typicky setkat se dvěma skupinami operací, vyhlazováním obrazu a gradientními operacemi pro detekci hran.

Při vyhlazování obrazu dochází k utlumení vyšších frekvencí obrazové funkce. Výsledkem vyhlazování je obvykle potlačení náhodného šumu. Negativní vlastností vyhlazování je naneštěstí potlačování všech ostatních vysokých změn jasové funkce, zejména ostrých čar a hran, které jsou nositeli významných informací. Gradientní operace bývají využívány k ostření obrazu a na rozdíl od vyhlazování vedou ke zdůraznění vyšších frekvencí. Zvýrazněny jsou tedy ty obrazové elementy, kde dochází k náhlé změně jasové funkce, a která má zde tedy vysoký modul gradientu. Požadovaným výsledkem gradientních operací je zvýraznění hran v obraze. Bohužel dochází ke zvýraznění nežádoucích artefaktů v obraze, jakými je například šum. Pokud lze porovnat vyhlazovací a gradientní operace, je možné si všimnout jejich

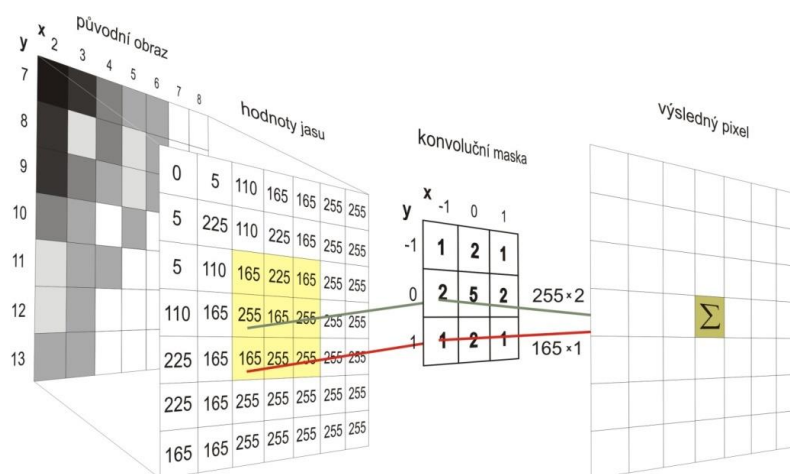


protichůdnosti. Naštěstí existují některé algoritmy, které využívají a kombinují oba zmíněné postupy a umožňují, aby byl obraz vyhlazován a ostřen současně, viz (10).

Metody lokálního předzpracování lze navíc na základě transformačních vlastností rozdělit na lineární a nelineární. Konvoluční lineární operace (filtry) vypočtou výslednou hodnotu výstupního pixelu  $g(x, y)$  jako lineární kombinaci jasů v lokálním sousedství  $O$  pixelu  $f(x, y)$  ve vstupním obraze. Příspěvek pixelu v sousedství  $O$  je násoben koeficientem  $h$ :

$$f(x, y) = \sum_{(m,n) \in O} h(x - m, y - n)g(x, y) \quad (2.6)$$

Rovnice (2.6) je ekvivalentem k diskrétní konvoluci s jádrem  $h$ , které je nazýváno konvoluční maska. Obdélníkové sousedství  $O$  je tvořeno lichým počtem pixelů v řádcích i sloupcích, aby bylo možné specifikovat střed sousedství. Konvoluční filtr názorně ilustruje Obr. 5.



Obr. 5 – Konvoluční filtry, převzato z (13)

Velmi často je vyžadováno snížení diference jasu uvnitř oblastí, které způsobují šum. Volba transformace je obvykle závislá na velikostech objektů. Pokud lze očekávat, že objekty zájmu budou relativně velké, a naopak objekty šumu budou odpovídat pouze velmi malým oblastem, které se od objektů liší jasnem, pak je možné šum v obraze celkem snadno odfiltrovat. Tyto metody jsou založené na průměrování ve vybraném okolí a dochází k zanedbatelnému rozmazání objektů.

## Vyhlazování obrazu

Filtraci šumu odpovídá ve frekvenčním spektru potlačení vysokých plošných frekvencí, které jsou výsledkem dvourozměrné Fourierovy transformace v obraze. Během filtrace obrazu dochází přirozeně kromě potlačení šumu také k rozmazání hran. V případech, kdy jsou objekty vzhledem k oblastem šumu poměrně malé, může přinést dobré výsledky transformace, která vyhledá a odstraní relativně velké rozdíly jasů v obraze.

Metoda obyčejného průměrování filtruje obraz tak, že hodnota jasů bodu je vypočtena jako aritmetický průměr jasů bodů jeho obdélníkového okolí. Dochází k potlačení hodnot šumu menších, než je nejmenší významný detail nacházející se v obraze. Metoda obyčejného průměrování má však svoji nevýhodu, kterou je rozmazávání hran. Pro vyhlazování obrazu lze využít i filtr s Gaussovým rozložením, kdy je konvoluční jádro reprezentováno maticí 5x5, příkladem může být konvoluční maska definovaná rovnicí (2.7):

$$\mathbf{h} = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad (2.7)$$

Ke zrychlení výpočtu je možné poměrně snadno vytvořit specializované technické prostředky. Urychlení výpočtů je obzvlášť podstatné pro speciální tvary filtrů, kde příkladem mohou být rekurzivní a separabilní filtry, viz (10).

Medián je další zajímavou možností vyhlazování obrazu, kdy se pro výsledný jas využívá prostřední hodnota zvoleného okolí. Výhodou je redukce rozmazání hran, nevýhodou poškození tenkých čar a ořez ostrých rohů (15).

Poslední vyhlazovací metodou, která bude zmíněna, je vyhlazování rotující maskou. Metoda podle homogenity jasů hledá k filtrovanému bodu část jeho okolí, ke které pravděpodobně patří (kde má jas nejmenší rozptyl) a tu pak využije pro výpočet. Metoda má obvykle mírně ostřící charakter a využívá se osmi různých pozic stejné masky, viz (10).

## Hranové detektory

Hranové detektory jsou kolekcí velmi důležitých metod lokálního předzpracování, založených na gradientních operacích. Hrany jsou místa, kde se skokově mění hodnota jasové funkce a přítomnost hran, jako nositelů důležitých informací, vede k pochopení obsahu obrazu. Pro detekci hran vzniklo několik konvolučních masek, invariantních i neinvariantních vůči rotaci. Neinvariantní vznikly aproximací první derivace a lze se setkat s Robertsovým, Sobelovým, Prewittovým či Kischovým operátorem (10). Operátory invariantní vůči rotaci vycházejí z aproximace druhé derivace a zde je možné zmínit Laplaceův operátor (10). Hranové detektory lze navíc velmi dobře využít i k segmentaci obrazu.

### 2.3.3 Segmentace obrazu na objekty

Segmentace obrazu je proces zpracování a rozdělení digitálního obrazu na významné části, které mají podobné vlastnosti (16). Typickými vlastnostmi mohou být např. barva, tvar nebo textura. Významnou část obrazu nazýváme objekt nebo oblast zájmu, zbytek obrazu je tzv. pozadí. Výsledkem segmentace jsou skupiny vzájemně se nepřekrývajících oblastí a vzniká zjednodušená reprezentace obrazu, která je smysluplnější a snáze zpracovatelná při pozdější analýze.

K nejstarším a nejjednodušším metodám segmentace obrazu patří tzv. prahování (*thresholding*) (10). Základem metody je hodnocení jasu každého pixelu. Díky skutečnosti, že objekty či oblasti jsou charakterizovány podobnými hodnotami reflexních či absorpčních vlastností svého povrchu, lze tyto hodnoty jasu využít k oddělení objektů od pozadí. Velmi významnou roli při využití metody *thresholdingu* představují histogramy.

Potřebujeme-li oddělit objekty od pozadí, je třeba zvolit práh, který rozdělí obraz na barvu pozadí a na barvu objektů. Metoda je obecně velmi dobře použitelná, pokud se v obraze vyskytují tmavé objekty na světlém pozadí. Potom každý pixel, který má menší hodnotu než zvolený práh, je nazýván bod objektu a ostatní pixely jsou body pozadí. Pokud máme stanovený práh  $T$ , pak platí rozdělení (2.8):

$$g(x, y) = \begin{cases} 1, & f(x, y) < T \\ 0, & f(x, y) \geq T \end{cases} \quad (2.8)$$

Pomocí prahování se otestují všechny pixely  $(x, y)$  obrazu  $f$ . Pixely, které mají po segmentaci hodnotu 1, představují objekty a body s hodnotou 0 jsou pixely pozadí. Hodnotu prahu je možné určit empiricky (manuálně z histogramu) nebo za pomoci některých automatizovaných metod určování prahu. Výsledkem je binární obraz, tedy obraz, který obsahuje pouze dvě barvy, zpravidla černou a bílou. Zřídka je možné pracovat pouze s jedním prahem, kvůli změnám jasu objektu i pozadí způsobeným nerovnoměrností osvětlení. V tomto případě lze využít metodu prahování s proměnným prahem, přičemž hodnota prahu je určována dle lokálních vlastností obrazu. Dochází k rozdělení obrazu do několika samostatných podobrazů a u těch je následně postupováno jako při běžném prahování. Modifikovaná metoda thresholdingu s globálním prahem je prahování s více prahy, pro které platí rozdělení (2.9):

$$g(x, y) = \begin{cases} 1, f(x, y) \in A_1 \\ 2, f(x, y) \in A_2 \\ \dots \\ n, f(x, y) \in A_n \\ 0, jinak \end{cases} \quad (2.9)$$

$A_i$  jsou podmnožiny definovaných jasových úrovní. Výsledkem tak není binární obraz, ale obraz s omezeným počtem jasových úrovní. Běžně dochází k situacím, že dochází k výraznému překryvu jasu pixelů popředí a pozadí. Příčinou může být například výrazný podíl šumu v samotném obrazu nebo samotná kompozice snímku je velice složitá. Přímé použití prahování bývá v takových případech prakticky nemožné.

Jak je zmiňováno výše, kromě manuální volby prahu existují metody, které hodnotu vypočítají automaticky na základě rozložení odstínů vstupního obrazu. Jednou z nejznámějších je Otsu metoda prahování (17). Algoritmus opět předpokládá, že se obraz skládá ze dvou tříd pixelů – popředí a pozadí. Optimální hodnota prahu je určena jako ta, která minimalizuje společný rozptyl popředí a pozadí.

Jednou ze základních segmentačních úloh je detekce hran. Hranou rozumíme souvislou linii oddělující dvě jasově výrazně odlišné oblasti. Jednotlivé segmenty v segmentovaném obraze jsou tvořeny oblastmi ohraničenými právě takto vzniklými hranami. Výsledným obrazem těchto algoritmů je opět binární obraz, kde jedna z barev znázorňuje hrany v původním obraze. K populárním metodám detekce hran

patří metody založené na první či druhé derivaci funkce obrazového signálu, a použití konvoluce s vhodnou konvoluční maskou jak je zmiňováno v kapitole 2.3.2, popřípadě velmi oblíbený a robustní Cannyho detektor hran (18).

Další možností, jak segmentovat obraz, je využití regionálních metod segmentace (*region-based methods*), jak je uvedeno v (19). Tyto metody jsou založeny na zjišťování, zda existuje podobnost obrazových bodů ve zvolené vlastnosti, kterou může být například jasová hodnota nebo statistické vlastnosti okolí pixelu. Metoda je založena na postupné konstrukci segmentu od jednoho pixelu, až je vytvořena celá oblast, kterou spojuje nějaká společná vlastnost. Na začátku jsou vhodným algoritmem v obraze rozmístěny inicializační, nebo také semínkové (*seed*) pixely. Rozložení bývá obvykle rovnoměrné, ale může být i náhodné, obrazové segmenty pak vznikají postupným rozrůstáním inicializačních pixelů do svého okolí. Při použití metod, které jsou založeny na postupném růstu segmentu, není zaručeno, že při jiném počtu a rozmístění semínkových pixelů bude výsledek segmentace totožný. Dalšími negativními skutečnostmi jsou celkem vysoká výpočetní náročnost, citlivost na větší množství šumu v obraze nebo fakt, že jde především o lokální metodu bez ohledu na celkovou kompozici obrazu.

Sledování hranice (*boundary tracking*) je postup prováděný na snímcích objektů, aby bylo možné získat informace o jejich obecném tvaru. Po extrakci obrysu daného objektu jsou zkoumány jeho vlastnosti, které mohou být později použity při klasifikaci. Proto správné vytažení obrysu vytvoří přesnější funkci, která zvýší pravděpodobnost správné klasifikace daného objektu. Sledování hranice může opět selhávat u obrazů se značným množstvím šumu, ale také v případě, že se v obraze vyskytují objekty příliš komplikovaných tvarů. Příkladem jsou algoritmy Square tracing, Moore – Neighborhood (20) nebo algoritmus Thea Pavlidise (21). Sledování hranice často významně přispívá k efektivitě procesu extrakce vlastností – základního procesu v oblasti rozpoznávání.

Aktivní kontura (*active contour, snake*) je jednou z pokročilých metod segmentace obrazu. Metoda vyžaduje, aby na jejím vstupu byla uzavřená křivka ohraničující segment, který je nutné ohraničit s vysokou přesností. Využívá se fyzikálních analogií, metoda pak definuje energie a síly, které vedou k deformaci vstupní uzavřené křivky tak, že se stane hranicí segmentu, jak uvádějí autoři v (22). Metoda

je úspěšně využívána v aplikacích jako sledování objektu, rozpoznávání tvaru nebo detekci hran.

Segmentace rozvodím (*watershed segmentation*) je metoda založená na principu postupného zaplavování obrazu vodou, přičemž jas pixelu si lze představit jako nadmořskou výšku a celý obraz jako trojrozměrnou mapu, viz (23). Každé lokální minimum a související oblast je chápána jako povodí. Každý segment je pak vlastně oblast zaplněná vodou a hranice segmentů jsou místa, kde dochází ke styku dvou zaplavených oblastí, které jsou oddělené nepřerušenou skupinou pixelů s vyšší hodnotou jasu, která tvoří pomyslnou hráz. Segmentace rozvodím má však své nevýhody. Metoda je velice citlivá na šum v obraze a šum v originálním obraze přímo zhoršuje gradient obrazu a může snadno způsobit odchylky segmentovaných obrysů. Vlivem šumu, kvantifikačních chyb či existenci nadměrného množství detailů může vzniknout neúměrně velké množství malých regionů. V neposlední řadě, u obrazu s nízkým kontrastem se některé důležité obrisy mohou úplně ztratit.

### **2.3.4 Popis objektů**

Popis jednotlivých objektů je proces, který následuje po segmentaci obrazu. Je možné popisovat primitivní vlastnosti jako je tvar, velikost, barva, umístění v obraze. Pokud se budeme zabývat tvarem, lze sledovat vlastnosti jako kompaktnost, podlouhlost, pravoúhlost, směr nebo výstřednost. Základem je získat dostatečné množství příznaků pro začlenění objektů do tříd a zároveň výběr správných příznaků, které přesně popisují charakteristické rysy. Kritérií pro správnou volbu příznaků je celá řada, ať se jedná o neměnnost, spolehlivost, či efektivitu výpočtu. Výsledkem snažení je vytvoření příznakového vektoru, který může mít libovolnou dimenzi a kde každý prvek představuje konkrétní vlastnost. Příznaky lze navíc rozdělit do několika skupin. Fotometrické popisovače například vyjadřují optické vlastnosti objektů, další deskriptory zachycují geometrické vlastnosti, některé jsou založené na základě reprezentace hranice, např. pomocí Freemanova kódu (24) a jiné mohou být založené na regionu, kdy je hodnota příznaku počítána na základě hodnot pixelů objektu. Využit lze i lokální deskriptory, kdy je hodnota získána pouze z konkrétní lokální části objektu. Příkladem mohou být hrany a rohy objektů nebo různá zakřivení.

### 2.3.5 Porozumění obsahu obrazu, klasifikace objektů

Posledním krokem při zpracování obrazu bývá klasifikace objektů a porozumění obsahu obrazu. Ve většině případů dochází k zařazení objektů nalezených v obraze do předem známých a definovaných tříd. Klasifikační metody se dělí do dvou základních skupin, které jsou úzce spojeny se způsobem popisu objektů a jedná se o příznakové nebo strukturální rozpoznávání. Základem příznakových metod je princip využití příznaků, což může být vektor číselných charakteristik objektu. Definice vlastního klasifikátoru se provádí za pomoci metod strojového učení, které může být s trénovací množinou, ale i bez ní, s využitím principu shlukové analýzy. Mechanismy strojového učení jsou blíže popsány v kapitole 2.4. Strukturální rozpoznávání definuje jednotlivé obrazy jako množinu základních popisných elementů – tzv. primitiv, sleduje jejich vlastnosti a vztahy mezi nimi. Dále je vytvořena abeceda, jazyk popisu a gramatiky jednotlivých tříd. Vlastní rozpoznávání je pak založeno na principu rozboru vět na slova a výhodou této analýzy je kromě klasifikace do tříd i získání vztahů mezi nimi.

## 2.4 Strojové učení

Strojové učení (machine learning) je podoblastí umělé inteligence, zabývající se algoritmy a technikami, které dávají počítačovému systému schopnost učit se, aniž by byl explicitně programován. Základním předpokladem jsou algoritmy, které přijímají vstupní data, využívají statistických analýz a předpovídají výstupní hodnoty v přijatelném rozsahu. Cílem je vlastně změnit data v informace. Algoritmy pracují s datovými sadami, které se skládají z tisíců příkladů, přičemž každý příklad může mít desítky i stovky atributů. V obraze 20 x 20 pixelů nalezneme 400 hodnot jasové funkce, pokud budeme uvažovat obraz ve stupních šedi a 1200 hodnot, pokud se bude jednat o obraz barevný. Datové sady se zpracují, jednotlivým atributům jsou nastaveny váhy a příklad je klasifikován. Algoritmy pro strojové učení lze rozdělit podle způsobu učení do dvou základních skupin, učení s učitelem (supervised learning) a učení bez učitele (unsupervised learning). V prvním případě existuje pro vstupní data správný výstup, který bývá označován jako label, ve druhém případě máme pouze množinu vstupních dat, ve kterých hledáme nějakou spojitost. Existují i kombinace obou zmíněných metod, kdy pro část dat existují známé výstupy. Dobrých výsledků dosahuje i tzv. zpětnovazebné učení, viz (25).

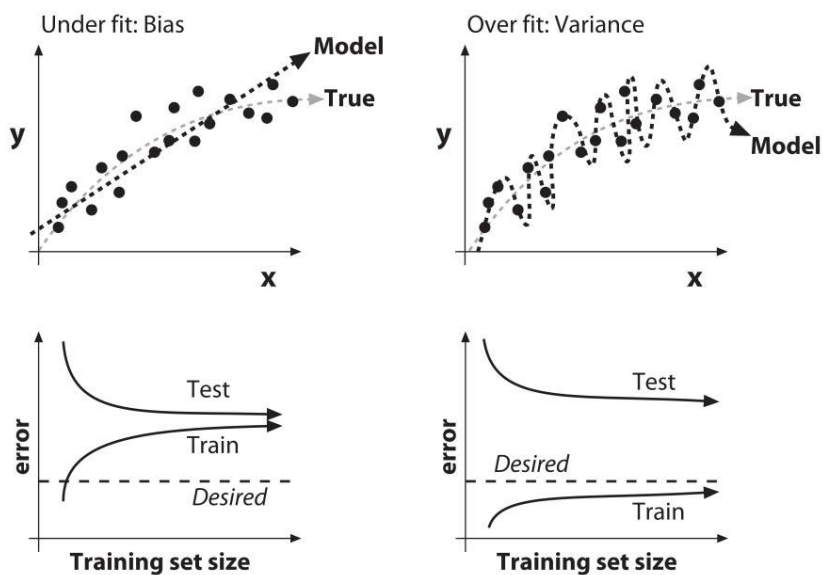
Klasifikace je jedna ze základních úloh strojového učení. Vstupní data rozděluje do několika tříd. Pokud jsou vstupní data rozdělena pouze do dvou tříd, jedná se o binární klasifikaci. Dalšími typickými úlohami strojového učení je regrese, kdy je odhadována číselná hodnota výstupu na základě vstupních dat (např. odhad ceny nemovitosti ze zadaných atributů) nebo shluková analýza, kdy jsou vstupní data rozdělena do skupin s podobnými vlastnostmi. Shlukování je typické pro učení bez učitele (8).

Proces strojového učení začíná sběrem dat. Už při samotném sběru dat je nutné rozmyslet způsoby a metody, aby bylo možné data později označit a rozdělit do tříd. Navíc každý příklad může mít desítky a stovky atributů a je důležité rozhodnout, který z atributů je pro klasifikaci důležitý a který nikoliv. Následuje označení a rozdělení dat do tříd. Z označených dat je vybrána množina příkladů, na kterých se provádí trénování a testování. Ve fázi testování je ideální využít postup, kterému se říká křížová validace. Data jsou rozdělena do několika skupin, trénovací proces se spustí několikrát a v každém procesu jsou využity jiné skupiny dat pro trénink a testování.

Fungující model může být často více typem umění, než skutečnou vědou. Výsledky často nemusejí být takové, jak by člověk očekával, a je nutné zjistit, proč model nepracuje správně. Existuje několik rad a doporučení, které pomohou vytvořit spolehlivější modely. Více dat je lepší, než méně dat a kvalitnější data, ve smyslu správně volených atributů, jsou lepší, než vhodněji zvolený algoritmus. Pokud jsou data připravena správně, maximalizujeme nezávislost jednotlivých příkladů a minimalizujeme odlišnosti v různých podmínkách, většina algoritmů bude fungovat správně.

I tak se lze setkat se dvěma běžnými problémy. Prvním je bias, kdy je model příliš jednoduchý a algoritmus ztrácí důležité vztahy mezi atributy a cílovými výstupy. Výkon takového modelu je špatný jak na trénovacích, tak testovacích datech. Druhým problémem je variance, kdy naopak máme příliš atributů a model neumí generalizovat výstupy z testovacích příkladů, tj. má vynikající výsledky na trénovacích datech, ale velmi slabé výsledky na datech testovacích, jak zachycuje Obr. 6. Tento jev je také nazýván přeučení modelu, neboli overfitting, viz (8).





Obr. 6 – Underfitting a overfitting, převzato z (8)

Základní chyby v modelech a možnosti jejich odstranění jsou uvedeny v Tabulce 1.

Tabulka 1 - Chyby v modelech a možná řešení

Problém	Možné řešení
Bias	Využít více atributů Použít výkonnější algoritmus
Variance	Využít více trénovacích dat Snížit počet atributů a redukovat možnost přeučení Použít méně výkonný algoritmus
Dobré výsledky na trénovacích i testovacích datech, špatné na reálných datech	Nasbírat realističtější data
Model se není schopen učít z trénovacích a testovacích dat	Pozměnit, či vybrat jiné atributy pro zachycení invariance v datech Nasbírat nová, významnější data Použít výkonnější algoritmus

## 2.5 Algoritmy strojového učení

Existuje celá řada algoritmů, které se využívají pro strojové učení. V následujících podkapitolách budou uvedeny dva běžně užívané algoritmy, včetně jejich variací, které byly následně využity pro klasifikaci a predikci v praktické části.

### 2.5.1 Stromy

Rozhodovací strom je stromový graf nebo model. Je to spíše obrácený strom, protože má kořen nahoře a roste dolů. Toto zobrazení dat má výhodu ve srovnání s jinými

přístupy, neboť je lze snadno interpretovat. Cílem je vytvořit klasifikační model, který předpovídá hodnotu cílového atributu (často označovaného jako třída nebo štítek – label) na základě několika vstupních atributů. Každý vnitřní uzel stromu odpovídá jednomu ze vstupních atributů. Každý list stromu představuje hodnotu atributu štítku vzhledem k hodnotám vstupních atributů reprezentovaných cestou z kořene k listu.

Rozhodovací stromy jsou generovány rekurzivním rozdělením. Rekurzivní dělení znamená opakované rozdělení na základě hodnot atributů. Nejznámějším klasifikačním algoritmem pro generování rozhodovacích stromů metodou odshora dolů je ID3 (26), který lze popsat následujícím způsobem:

1. Strom začíná jako jeden samostatný uzel, který reprezentuje všechny trénovací příklady.
2. Pokud všechny příklady patří do stejné třídy, pak se takový uzel stane listovým uzlem a je označen touto třídou.
3. V opačném případě algoritmus použije metodu označovanou jako informační zisk pro výběr atributu, který bude schopen nejlépe rozdělit příklady do jednotlivých tříd. Takový atribut se stane buď testovacím, nebo rozhodnutím – třídou. Algoritmus pracuje pouze s diskrétními hodnotami, pokud se na vstupu objevují jiné hodnoty, musí být vždy převedeny na diskrétní hodnoty.
4. Pro každou existující hodnotu atributu dochází k vytvoření samostatné větve a příklady jsou přiřazeny jednotlivým větvím.
5. Algoritmus totožným způsobem vytváří rozhodovací strom v každé jeho části. V případě, že se atribut použije jako testovací kritérium, nesmí být znovu použitý pro testování v této části stromu.
6. Opakované dělení může skončit v momentě, kdy je splněna některá z následujících podmínek:
  - všechny příklady vybraného uzlu náleží do stejné třídy
  - neexistují žádné další atributy, podle kterých by mohlo dojít k dalšímu dělení. V takovém případě je uskutečněna tzv. „většinová volba“, která zahrnuje převod daného uzlu na list a označí se třídou, která se vyskytuje ve většině příkladů tohoto uzlu

- neexistují žádné další vzorky, které by patřily k dané větvi. V tomto případě je list vytvořený z většinové třídy v příkladech

Algoritmus ID3 garantuje generování minimálního rozhodovacího stromu, pokud platí:

- podmínka nekontradikčnosti trénovacích příkladů – ID3 není odolný vůči zašuměným vstupním údajům
- podmínka neredundantnosti trénovacích příkladů – není možný vícenásobný výskyt stejného trénovacího příkladu v trénovací množině
- podmínka vzájemné nezávislosti atributů

Algoritmy náhodných stromů (random trees) byly představeny Leo Breimanem a Adele Cutlerovou a jsou využitelné jak pro klasifikační, tak i regresní problémy. (27) Model náhodného stromu pracuje podobně jako model rozhodovacího stromu s jednou výjimkou, v každém uzlu každého trénovaného stromu se nepoužívají všechny atributy k nalezení nejlepšího rozdělení, ale náhodná podmnožina. S každým uzlem je generována nová podmnožina. Velikost podmnožiny je však pevná pro všechny uzly.

Náhodný les (random forest) generuje množinu náhodných stromů. Náhodné stromy jsou generovány přesně stejným způsobem, jakým algoritmus náhodného stromu generuje strom. Klasifikace funguje jako volební model, klasifikátor náhodných stromů vezme vektor vstupního prvku, klasifikuje ho všemi stromy v lese a zařadí do třídy, která získala většinu „hlasů“. V případě regrese je odpověď klasifikátoru průměrem odpovědí všech stromů v lese.

Gradient boosting je metoda typicky používaná v rozhodovacích stromech a lze ji opět využít jak pro klasifikační, tak i regresní úlohy. Oba postupy využívají dopředného učení, kdy získávají výsledky prostřednictvím postupně zlepšených odhadů. Boosting je pružný postup nelineární regrese, který pomáhá zvýšit přesnost stromů. Postupným aplikováním slabých algoritmů klasifikace na postupně změněné údaje se vytváří řada rozhodovacích stromů, které vytvářejí soubor slabých modelů předpovědí. Boosting stromů zvyšuje jejich přesnost, ale také snižuje rychlost a lidskou interpretaci.

Nejznámějším algoritmem pro boosting se nazývá Adaboost, odvozené varianty jsou známé jako Discrete Adaboost, Real AdaBoost, LogitBoost, and Gentle AdaBoost (28). Základní algoritmus předpokládá klasifikaci do dvou tříd  $\{-1, 1\}$ . V pseudokódu lze Adaboost popsat následujícím způsobem, jak je uvedeno v (28):

Mějme vstupní trénovací množinu vzorů  $(x_1, y_1), \dots, (x_m, y_m)$ , kde  $x_i \in X, y_i \in Y = \{-1, 1\}$

Nastavíme váhu všech vzorů v trénovací množině na  $D_1(i) = \frac{1}{m}$ , kde  $i$  je index vzoru

Pro  $t = 1, \dots, T$ :

- Nauč slabý klasifikátor za použití rozdělení  $D_t$ .
- Získej slabou hypotézu  $h_t: X \rightarrow \{-1, 1\}$  s chybou  $e_t = P_{i \sim D_t}[h_t(x_i) \neq y_i]$ .
- Zvol  $\alpha_t = \frac{1}{2} \ln \left( \frac{1-e_t}{e_t} \right)$ .
- Změň:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & h_t(x_i) = y_i \\ e^{\alpha_t}, & h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

kde  $Z_t$  je normalizační faktor.

Výstup finální hypotézy je definován rovnicí (2.10):

$$H(x) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x) \right) \quad (2.10)$$

Tento postup řeší otázku binární klasifikace a rozdělení do dvou možných tříd. Mnoho, možná většina problémů reálného světa si s binární klasifikací nevystačí, proto existují rozšíření boosting algoritmů pro vícetřídové klasifikátory. Obecně se provádí převod vícetřídových klasifikátorů na skupiny jednodušších binárních klasifikátorů. Řešení je popsáno v algoritmech AdaBoost.MH a AdaBoost.M2, viz (29).

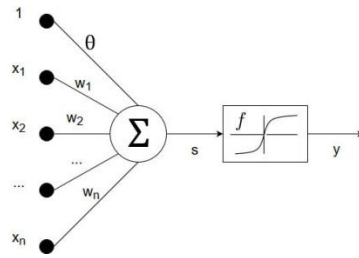
V modelech stromů s výjimkou náhodných stromů a lesů se setkáváme s technikou prořezávání (pruning), ve které jsou odstraněny uzly listů, které nepřispívají k diskriminační síle rozhodovacího stromu. Proveďte se převod stromu na obecnější formu, aby se zvýšila jeho prediktivní síla a snížila možnost přeučení. Prepruning je typ prořezávání prováděného paralelně s procesem vytváření stromu, kdy se růst stromu zastaví dříve, než je strom schopen perfektně klasifikovat trénovací data. Postpruning se provádí po dokončení procesu vytvoření stromu a tato metoda je využívána mnohem častěji, protože není snadné odhadnout, kdy skončit s růstem stromu. Postpruning využívá dvou operací, subtree replacement, kdy je vybrán podstrom a nahrazen listem, což může snížit přesnost na trénovacích datech, ale zlepšit predikci na testovacích datech. Postupuje se od listů ke kořeni, v každém uzlu buď podstrom nahradíme, nebo ponecháme. Druhou operací je subtree raising, kdy jeden uzel nahradíme jeho podstromem a překlasifikujeme příklady, které patřily do zbylých podstromů, tento princip je využitý v algoritmu C4.5, jak je uvedeno v (30). V praxi je tento princip zkoušen pouze pro nejnavštěvovanější větve.

## 2.5.2 Neuronové sítě

Umělá neuronová síť (artificial neural network), obvykle stručně nazývaná jen neuronová síť (neural network), je matematický model nebo výpočetní model, který je inspirován strukturou a funkčními aspekty biologických neuronových sítí. Neuronová síť se skládá z vzájemně propojené skupiny umělých neuronů a zpracovává informace pomocí spojovacího přístupu k výpočtu. Ve většině případů je neuronová síť adaptivní systém, který mění svou strukturu na základě externích nebo interních informací, které prochází sítí během fáze učení. Moderní neuronové sítě se obvykle používají k modelování komplexních vztahů mezi vstupy a výstupy nebo k nalezení vzorků v datech. Základní jednotkou sítě je neuron, který má  $n$  vstupů  $x_1, x_2, \dots, x_n$ , které jsou ohodnoceny vahami  $w_1, w_2, \dots, w_n$ . Každý neuron má také posun  $\theta$ , který supluje roli prahu u binárních klasifikátorů. Vnitřní potenciál neuronu lze chápat jako:

$$\xi = \sum_{i=1}^n w_i x_i \quad (2.11)$$

Struktura neuronu je znázorněna na Obr. 7.



Obr. 7 – Struktura neuronu s logistickou aktivační funkcí

Transformace vstupů neuronu  $x_1, x_2, \dots, x_n$  na výstup  $y$  je definována vzorcem,

$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right) = f\left(\sum_{i=0}^n w_i x_i\right) = f(s) \quad (2.12)$$

kde se  $\theta$  nahradí vahou  $w_0$  na vstupu  $x_0 = 1$ . Funkce  $f$  se nazývá aktivační funkce a pro klasifikační úlohy se často využívá funkce sigmoida definovaná jako:

$$f(s) = \frac{1}{1 + e^{-s}} \quad (2.13)$$

Učení neuronu probíhá v pěti krocích, jak popisuje autor v (31):

1. Inicializace vah na malé náhodné hodnoty.
2. Předložení vstupu  $X = x_1, x_2, \dots, x_n$  a odpovídajícího výstupu  $d$ .
3. Výpočet výstupu

$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right)$$

4. Upravení vah, pokud je výstup  $y$  různý od očekávaného výstupu  $d$

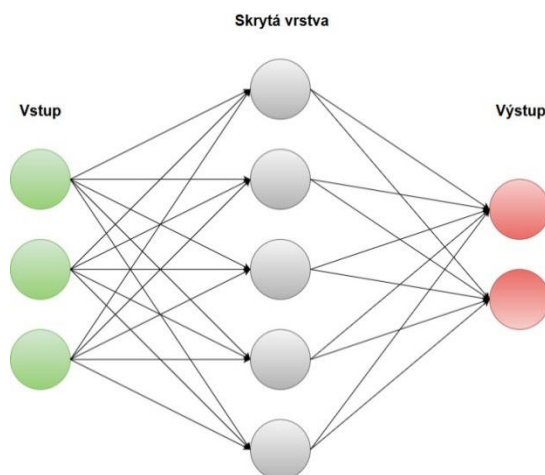
$$w_i(t+1) = w_i(t) + \alpha * (d - y) * x_i(t),$$

kde  $w_i(t)$  je hodnota váhy  $i$ -tého vstupu v čase  $t$  a  $\alpha$  je tzv. parametr učení.

5. Opakování od kroku č. 2, pokud nejsme spokojeni s výsledkem.

Vícevrstvý perceptron (MLP) je model umělé neuronové sítě, který se skládá z několika vrstev uzlů v orientovaném grafu bez kružnic, přičemž každá vrstva je plně spojena s další vrstvou. S výjimkou vstupních uzlů je každý uzel sítě samostatným neuronem. MLP funguje jako neuronová síť dopředného šíření vstupního signálu (feed-forward). V této síti se informace pohybují pouze v jednom směru dopředu ze vstupních uzlů skrze skryté uzly do výstupních uzlů, jak ilustruje

Obr. 8. V síti nejsou žádné cykly nebo smyčky. MLP využívá algoritmu zpětného šíření chyby pro učení sítě.



Obr. 8 – Schéma neuronové sítě s jednou skrytou vrstvou

Algoritmus zpětného šíření chyby (back propagation) je metodou učení s učitelem, která může být rozdělena do dvou fází: propagace a aktualizace vah. Obě fáze se opakují, dokud není výkon sítě dostatečně dobrý. V algoritmech zpětného šíření jsou výstupní hodnoty porovnávány se správnou odpovědí pro výpočet hodnoty některých předdefinovaných chybových funkcí. Různými technikami je chyba přenášena zpět přes síť. Pomocí těchto informací algoritmus upravuje váhy každého spojení, aby se snížila hodnota chybové funkce o malé množství. Po opakování tohoto procesu pro dostatečně velký počet tréninkových cyklů se síť obvykle stabilizuje do nějakého stavu, kde je chyba výpočtů malá. V takovém případě by se dalo říct, že se síť naučila určitou cílovou funkci. Algoritmus zpětného šíření chyby je podrobně popsán například v (32).

## 2.6 Senzitivita, specifita, $F_1$ skóre a přesnost

V binární klasifikaci se často setkáme s množinami dat, kdy jsou jednotlivé třídy nerovnoměrně zastoupeny. Ve výrobě je běžné, že např. 98% sledovaných vzorků jsou bez vad a pouze u 2% se nějaká výrobní vada vyskytuje. Pokud by klasifikátor označoval všechny vzorky jako bezvadné, bude mít přesnost 98%, což je velice dobrá hodnota, ale užitečnost takového klasifikátoru je nulová. Procento chybných predikcí tedy není dobrou mírou kvality klasifikátoru. Řešením je použití metrik vycházejících z matice záměn, kterou ukazuje Tabulka 2.

Tabulka 2 - Matice záměn

	Pozitivní skutečnost	Negativní skutečnost
Pozitivní predikce	Skutečně pozitivní (TP)	Falešně pozitivní (FP)
Negativní predikce	Falešně negativní (FN)	Skutečně negativní (TN)

- Skutečně pozitivní (TP – true positives) – vzorky, které model správně označil jako pozitivní (tj. vzorky, které jsou se skutečností vadné a model je také označil jako vadné)
- Skutečně negativní (TN – true negatives) – vzorky, které model správně označil jako negativní (tj. vzorky, které jsou se skutečností bezvadné a i model je označil jako bezvadné)
- Falešně negativní (FN – false negatives) – vzorky, které model mylně označil jako negativní (tj. vzorky, které jsou se skutečností vadné, ale model je označil jako bezvadné)
- Falešně pozitivní (FP – false positives) – vzorky, které model mylně označil jako pozitivní (tj. vzorky, které jsou se skutečností bezvadné, ale model je označil jako vadné)

Z hodnot zanesených v matici záměn získáme vzorce pro výpočet senzitivity, specificity a dalších metrik, jak uvádí autor v (33):

$$\text{senzitivita (recall, true positive rate, TPR)} \quad \frac{TP}{TP + FN} \quad (2.14)$$

$$\text{specificita (true negative rate)} \quad \frac{TN}{TN + FP} \quad (2.15)$$

$$\text{precision (positive predictive value, PPV)} \quad \frac{TP}{TP + FP} \quad (2.16)$$

$$\text{negative predictive value (NPV)} \quad \frac{TN}{TN + FN} \quad (2.17)$$

$$\text{přesnost (accuracy)} \quad \frac{TN + TP}{TN + FN + TP + FP} \quad (2.18)$$

V binární klasifikaci se pro výpočet přesnosti testovacího modelu využívá  $F_1$  skóre, někdy označované krátce jako F skóre, definované rovnicí (2.19):



$$F_1 = 2 * \frac{PPV * TPR}{PPV + TPR} = \frac{2 * TP}{2 * TP + FP + FN} \quad (2.19)$$

## 2.7 Typické úlohy strojového vidění

Hlavním úkolem strojového vidění ve výrobě bývá kontrola produktů tak, aby na konci výrobního procesu vznikl naprosto bezchybný výrobek. Výrobní proces se většinou skládá z několika výrobních operací, které na sebe navazují. Vzniklo mnoho teoretických prací, zabývající se vlivem organizace a optimalizace výrobního procesu na kvalitu výroby, téměř vždy však obsahují předpoklad precizního provedení alespoň některých výrobních operací tak, aby byl vyroben bezvadný produkt, jak uvádí autor v (34). Právě strojové vidění bývá nástrojem, který umožňuje tuto podmínku splnit. Typickým použitím strojového vidění ve výrobním průmyslu se stává kontrola korektního provedení výrobní operace.

Techniky strojového vidění se proto převážně používají v úlohách, které jsou úzce spojeny s kontrolou výrobní operace. Výrobní operaci lze jednoduše namodelovat jako posloupnost čtyř navazujících fází:

1. shromáždění materiálů a součástí – do této fáze by měla patřit kontrola materiálů vstupujících do výrobní operace
2. příprava materiálů a součástí ke konkrétní operaci – nastavení dílů do operační polohy, příprava na provedení operace
3. samotné provedení operace – tím může být např. nanesení barvy na výrobek, napájení konektoru apod.
4. ukončení operace a vyzvednutí výsledného produktu – zde by opět mělo docházet k provedení výstupní kontroly

V každé fázi se může vyskytovat jedna nebo několik úloh, kdy správné provedení výrobní operace vede k vytvoření bezchybného vyráběného produktu. Úlohy se ve výrobních operacích často opakují v různých obměnách a při určitém zobecnění je lze rozdělit do několika základních kategorií:

- odhalování povrchových vad
- identifikace, vyhledání polohy a počítání kusů
- metrologické úlohy a kontrola tolerancí

- rozpoznávání barev
- načítání a ověřování textů a kódů

Každá z těchto úloh by našla své uplatnění při kontrole tištěných topných těles, jak bude uvedeno dále.

### **2.7.1 Odhalování povrchových vad**

Tato úloha je typická pro výrobní operace, kdy je sledována jakákoliv povrchová úprava produktu, ať se jedná o identifikaci poškozených povrchů, vad nátěrů nebo jiných povrchových úprav, neúplných potisků atd. Kontrola může probíhat jednak ve fázi shromažďování materiálu k operaci prováděné vstupní kontrolou. Je-li nátěr nebo jiná povrchová úprava sama o sobě výrobní operací, je možné tento typ úlohy efektivně využít také ve fázi vyzvednutí výsledného produktu. Touto úlohou se zabývá tato bakalářská práce.

### **2.7.2 Identifikace, vyhledání polohy a počítání kusů**

Tato úloha obvykle mívá mnoho podob, nejčastěji dochází ke zjištění, zda je správná součást na správném místě. Například při provádění operace zalisováním je nutné se ujistit, jestli jsou připraveny všechny díly k zalisování a zda se nacházejí ve správné poloze. Stejnou operaci je třeba udělat po skončení operace. S úlohou se lze rovněž setkat při vstupní kontrole. Dochází k identifikaci deformovaných částí, otřepů po stříhání nebo lisování, poškozené hrany nebo nevyvrtaného otvoru. Počítání kusů se uplatňuje při kontrole správně naplněné přepravky nebo blistru. V případě výroby tištěných topných těles by se tento postup dal využít ke kontrole osazení termistoru a správnému osazení jednotlivých konektorů.

### **2.7.3 Metrologické úlohy a kontrola tolerancí**

Úloha měření obecně prostupuje celým výrobním procesem a lze ji považovat za naprostý základ při provádění mnoha operací. Měří se tolerance vstupních dílů, poloha dílů před operací i tolerance hotového výrobku. Úloha by se opět dala uplatnit při kontrole napájení konektorů (správná rozteč a rovnoběžnost napájení).

#### **2.7.4 Rozpoznávání barev**

Barvy je třeba identifikovat jak při vstupní, ale i při výstupní kontrole, může jít například o jeden ze znaků při rozpoznání typu materiálu, barevná odlišnost může označovat skrytou výrobní vadu. Barva potravin může v potravinářském průmyslu signalizovat sníženou kvalitu produktu, vhodné je tak provádět tuto úlohu při procesu vyskladňování. Tato úloha je úzce spojena s odhalováním povrchových vad, které se mohou projevit právě jinou barvou oproti zbytku povrchu.

#### **2.7.5 Načítání a ověřování textů a kódů**

Kódy, ať jednorozměrné nebo dvourozměrné, se načítají zejména v první a druhé fázi výrobní operace, kdy je kontrolováno, nachází-li se správný díl na správném místě a zda je připraven k provedení operace. V průběhu operace někdy bývá vytvářen další kód, který se kontroluje v poslední fázi operace. Čtení i verifikace kódů většinou nebývají samostatnou úlohou strojového vidění ve výrobním procesu, ale je-li tomu tak, většinou bývají řešeny využitím specializovaných čteček. Pokud však systém strojového vidění provádí v dané operaci více úloh, může být výhodně využít i pro čtení kódů a textů. Typicky se čtení textů objevuje v první a poslední fázi operace. V první části se používá pro rozpoznávání dílů a v poslední části ke kontrole vytvořených identifikátorů, např. potisků s typem a datem expirace výrobku. I topná tělesa jsou značena maticovým kódem typu Data Matrix, který je čten speciální čtečkou. Při většině operací je přečteno sériové číslo výrobku a informace z aktuální části výrobního postupu jsou ukládány do databáze.

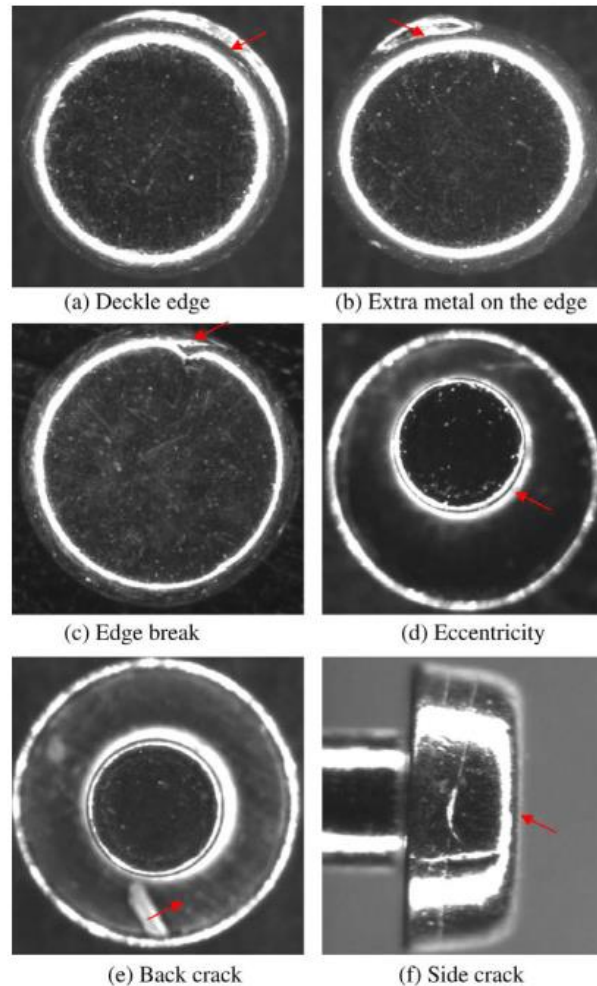
### **2.8 Praktické využití počítačového vidění**

V následujících podkapitolách je představeno několik řešení z různých průmyslových odvětví, kde počítačové vidění nachází své uplatnění. Vybrané studie využívají postupy, které mohou být inspirací v případě zjišťování kvality tištěných topných těles.

#### **2.8.1 Kontrola elektrických kontaktů**

Studie se zabývá kontrolou elektrických kontaktů (35). Elektrické kontakty jsou důležité elektronické komponenty, které spojují a rozpojují elektrické obvody v elektrických zařízeních. Studie popisuje systém pro trojsměrnou kontrolu (horní,

spodní, boční pohled) elektrických kontaktů a sleduje vady různých typů: hrubé okraje, otřepy, poškozené okraje, excentricitu či nejrůznější rýhy a škrábance, viz Obr. 9.



Obr. 9 – Povrchové vady elektrických kontaktů, převzato z (35)

Horní pohled zahrnuje předzpracování obrazu, blob analýzu a kontrolu kulatosti, spodní pohled sleduje excentricitu, rýhy a praskliny, boční pohled se také převážně zabývá sledováním prasklin. Implementace systému využívá nasnímaných obrazů v odstínech šedi v rozlišení 640 x 480 pixelů, snímaných na tmavém pozadí. Při snímání jsou využívány červené LED diody, které poskytují nejlepší výsledky pro osvětlení kovových kontaktů. Systém pro kontrolu vznikl v programovacím jazyce C++.

### 2.8.2 Detekce globálních a lokálních vad pružných tištěných obvodů

Pružné tištěné spoje jsou systémy tištěných obvodů na tenké elastické fólii vyrobené ze silikonu. Klíčové vlastnosti těchto prvků je vysoká hustota spojů, nízká váha,

zanedbatelná šířka a možnost ohýbání a překládání (36). Studie čínských autorů popisuje systém na detekci výrobních vad založený na klasickém modelu zpracování obrazu. Pro zachycení obrazu je využita CCD kamera pro snímání povrchu desek, následuje zpracování obrazu pro výpočet charakteristik a nalezení informací o možných výrobních vadách. Pro segmentaci povrchových vad je využita Gray Gradient Co-occurrence Matrix (GGCM). Pro identifikaci a klasifikaci jsou aplikovány neuronové sítě, díky nimž celý systém dosahuje vysoké přesnosti zpracování.

## 2.9 Knihovna OpenCV

OpenCV je otevřená multiplatformní knihovna pro manipulaci s obrazem zaměřená především na počítačové vidění a zpracování obrazu v reálném čase, původně vyvíjená společností Intel (37). V současné době ji lze využít v prostředí mnoha programovacích jazyků, C, C++, Python a od verze 2.4.4 také v prostředí Java. Knihovnu OpenCV lze snadno použít ve vývojovém prostředí Eclipse, jak popisuje autor v (38). Knihovna má modulární strukturu a je rozdělena do několika sdílených knihoven. K nejdůležitějším patří:

- `core` – kompaktní modul definující základní datové struktury včetně vícerozměrného pole `Mat` a další základní funkce používané všemi ostatními moduly
- `imgproc` – modul pro zpracování obrazu, který zahrnuje lineární i nelineární filtrování obrazu, geometrické transformace, konverze barevného prostoru, histogramy apod.
- `video` – modul pro analýzu videa, který zahrnuje odhady pohybu, odečítání pozadí a algoritmy sledování objektu
- `objdetect` – detekce objektů a instancí předdefinovaných tříd (např. tváře, oči, lidé, auta, atd.)
- `highgui` – snadno použitelné rozhraní pro zachytávání videa, obsahující kodeky pro obraz a video a také jednoduché uživatelské rozhraní

Knihovna také obsahuje řadu algoritmů pro strojové učení.

### 3 NAVRŽENÉ ŘEŠENÍ

Praktická část bakalářské práce je věnována vývoji systému pro kontrolu tištěných topných těles během výroby. V kapitole 3.1 je vysvětleno, co jsou tištěná topná tělesa a jaké je jejich praktické využití. Kapitola 3.2 krátce popisuje výrobu tištěných topných těles, zaměřuje se na kontrolu kvality a popis sledovaných výrobních vad. Ve finální fázi výroby, kdy je pro zvyšování kvality výroby pořizována obrazová dokumentace, jak je uvedeno v kapitole 3.3, by bylo vhodné nasadit optickou kontrolu do většího množství výrobních operací.

#### 3.1 Tištěná topná tělesa

Plochá tištěná topná tělesa jsou vhodná pro kontaktní ohřev rovinných ploch, případně tekutin přes rovinnou stěnu. Oproti jiným způsobům ohřevu mají tištěná topná tělesa řadu výhod:

- těleso na plechu zaručuje významně lepší přestup tepla do rovné stěny než klasické trubkové těleso
- rychlý náběh teploty – úspora energie
- snadná montáž a demontáž – úspora nákladů
- možnost vysokého povrchového zatížení – desítky  $W/cm^2$
- vnitřek ohřívané nádoby zůstává hladký, nečlenitý a snadno čistitelný
- velmi vhodné pro ohřev agresivních tekutin
- při údržbě odpadá nutnost vypouštět obsah nádoby
- ohřevem přes dostatečně velkou plochu lze účinně zabránit napalování obsahu na povrch nádoby

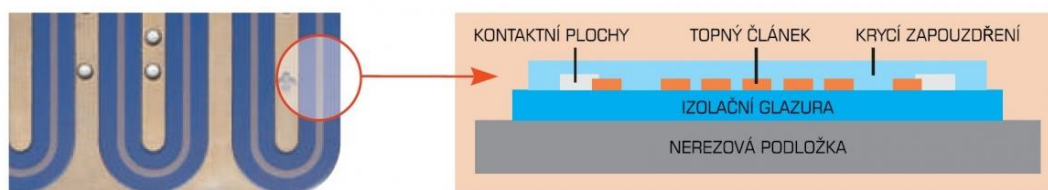
Tělesa mohou pracovat při běžných napětích v síti (až 400 V). Technologie umožňuje vyrábět tělesa s velmi vysokým plošným zatížením – až desítky  $W/cm^2$ . Jejich použití je ale limitováno aplikací u zákazníka – aplikace musí zajišťovat takový odvod tepla, aby povrchová teplota tělesa nepřesáhla cca 300 °C.

Tištěná topná tělesa mají znatelný tzv. PTC efekt (s rostoucí teplotou tělesa vzrůstá jeho odpor, tzn., klesá jejich příkon), který je pro řadu aplikací výhodný. Vzhledem k této vlastnosti je výstupním parametrem těchto těles odpor za studena a napětí, na které je těleso navrženo.

Tištěná topná tělesa lze využít v mnoha oblastech, kde je potřebný ohřev kapalin. Vyráběná tělesa se aktuálně využívají v automobilovém průmyslu, elektrických spotřebičích pro domácnost, zařízeních v gastronomii, pro ohřev průmyslových nádrží a zásobníků, či varen v minipivovarech.

### 3.2 Výroba a kontrola kvality

Těleso se skládá z nerezové destičky, na kterou se nejprve nanese potřebná tloušťka izolační glazury (splňuje podmínku elektrické pevnosti), na ni pak vlastní topný článek, následně kontaktní a propojovací vrstva a všechny tyto vrstvy se nakonec překryjí ještě vrchní vrstvou glazury (viz Obr. 10), která slouží pouze proti mechanickému poškození, nesplňuje podmínku elektrické pevnosti. Jednotlivé vrstvy se nanášejí sítotiskem a každá vrstva se vysouší a vypaluje. V průběhu procesu je důležitá maximální ochrana před nečistotami. Výroba se provádí v klimatizovaných prostorách s vysokou třídou čistoty.



Obr. 10 – Schéma tištěného topného tělesa

Takto precizně vyrobená topná tělesa se vyznačují rychlým náběhem teplot, minimální tepelnou setrvačností a minimálním kolísáním teploty. Vysoká účinnost 70 – 95 % je odvislá od způsobu užití těles pro přímý, nebo nepřímý ohřev.

Kvalita výrobků je nyní kontrolována manuálně po každé nanesené vrstvě. Seznam sledovaných vad je uveden v Tabulce 3. Pokud není zjištěna žádná závada, výrobek postupuje do dalších fází výroby. Pokud je zjištěno nesprávné natištění nebo nečistota, není možné nátisk jednoduše odstranit a těleso musí být z výroby vyřazeno. Stroje, na kterých probíhá tisk, jsou opatřeny kamerami, které by mohly kontrolovat kvalitu natištění, prozatím nejsou využívány.

Tabulka 3 - Soupis sledovaných vad

Kód vady	Popis vady
T010	škrábance na plechu
T011	špatný tvar plechu
T012	ostré hrany (otřepy)
T013	koroze plechu
T014	deformace plechu
T015	nelze založit do přípravku – nesoustřednost
T020	poškozený tisk
T021	zašpinění před výpalem
T022	přerušený spoj (tisk)
T023	nečistoty před výpalem
T024	posunutá poloha tisku
T025	neúplný tisk
T026	nestejný barevný odstín
T027	přetisk vrstvy přes pájecí plošku
T028	deska znečištěná pastou
T030	zapečené nečistoty z pece
T031	škrábance na tisku
T032	stopy po smirkování
T033	Poškozeno při manipulaci před výpalem
T040	průraz vysokým napětím
T041	nízký odpor
T042	vysoký odpor
T043	přerušený elektrický obvod
T044	poškozená izolace vodiče
T050	křivý konektor
T051	křivý termistor
T052	chybějící konektor
T053	chybějící termistor
T054	chybějící ERNI konektor
T055	nesmáčivé pájecí plošky
T056	tmavé konektory
T057	konektory nepřiléhající k ploše
T058	prasklý termistor
T059	Poškozený ERNI konektor
T060	zničeno při testování
T061	špatné značení
T062	chybějící značení

### 3.3 Fotodokumentace

Veškeré výrobky jsou nyní před expedicí fotografovány. Bohužel tato kvanta fotografií jsou vhodná pouze pro zdokumentování osazení dvojice konektorů, dvou–



pinového ERNI konektoru a termistoru a průkazu, že dodávka byla v době expedice kompletní.

## 4 PRŮBĚH A VÝSLEDKY VÝZKUMU

Při vlastním výzkumu jsem se zaměřil na zjišťování výrobních vad v izolační vrstvě, kde se nejčastěji objevují různé nečistoty, škrábance či výrazné barevné odlišnosti, viz Obr. 11. Důležité bylo pořídit dostatečné množství snímků, na kterých se některá výrobní vada vyskytuje, viz kapitola 4.1. Toto nakonec nebylo tak obtížné, protože vadné výrobky jsou skladovány v kontejnerech, kde nyní bohužel končí tisíce kusů s výrobní vadou. Snímky jsem zpracovával ve vlastní aplikaci, kde jsem vytvořil několik datových sad pro testování v algoritmech strojového učení, jak je popsáno v kapitole 4.2.



Obr. 11 – Ukázka vad v izolační vrstvě

Testování probíhalo v prostředí programu RapidMiner Studio, kde jsem zjišťoval využitelnost vybraných algoritmů. Testy shrnuje kapitola 4.3. Na základě slibných výsledků jsem doplnil vlastní aplikaci o možnost trénování vlastní neuronové sítě a testoval fungování kontroly povrchu tištěných topných těles, jak vysvětluje kapitola 4.4.

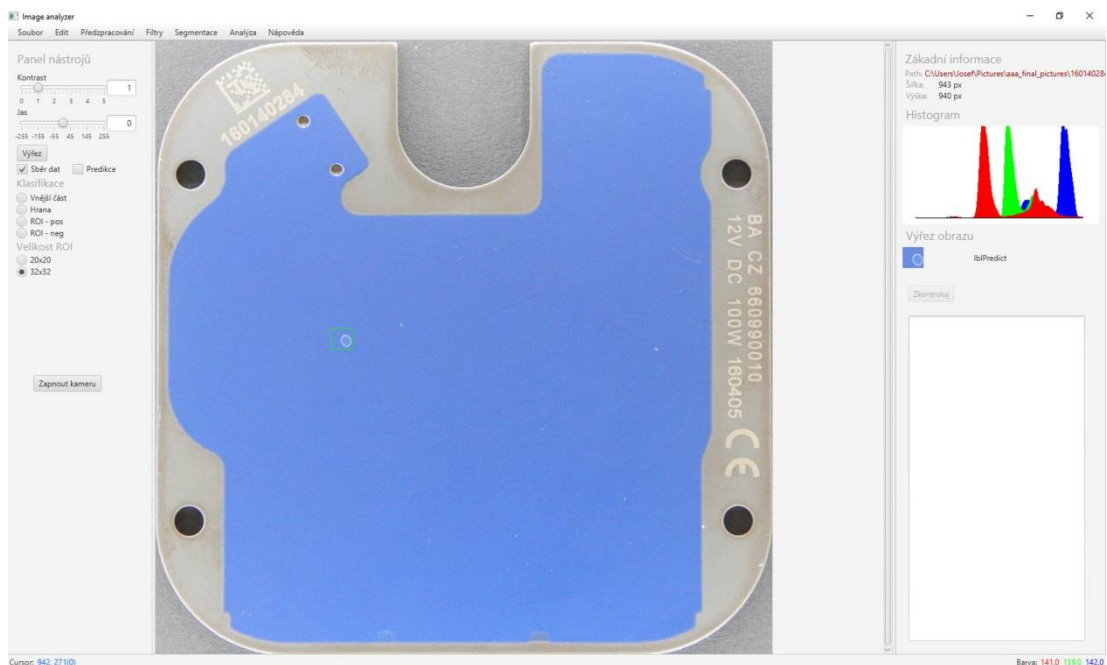
### 4.1 Sběr informací

Základním předpokladem pro další šetření je dostatečné množství testovacích vzorků. Snímky byly pořizovány za pomoci digitálního fotoaparátu umístěného na stativu a testovací vzorky byly umísťovány kolmo pod objektiv. Již při vlastním snímání docházelo k drobným komplikacím, protože nerezové podložky z oceli i samotné natištěné vrstvy odrážejí okolní objekty. Osvětlení snímaných desek hraje významnou roli a místo blesku je nutné využít konstantní zdroj světla. Nejlepších výsledků snímání bylo nakonec dosaženo po vytvoření osvětlovacího rámu. Do vnitřní strany rámu se nalepí dnes běžně dostupné LED pásy v několika vrstvách, čímž je dosaženo rovnoměrného osvětlení snímaného objektu ze všech stran. V našem případě byla využita páska s bílými LED diodami, využití jiných barev osvětlení vyzkoušeno nebylo. Snímání bylo testováno na bílém i černém pozadí,

přičemž snímky na tmavém pozadí lépe vykreslovaly výrobní vady, které byly na snímcích lépe rozeznatelné.

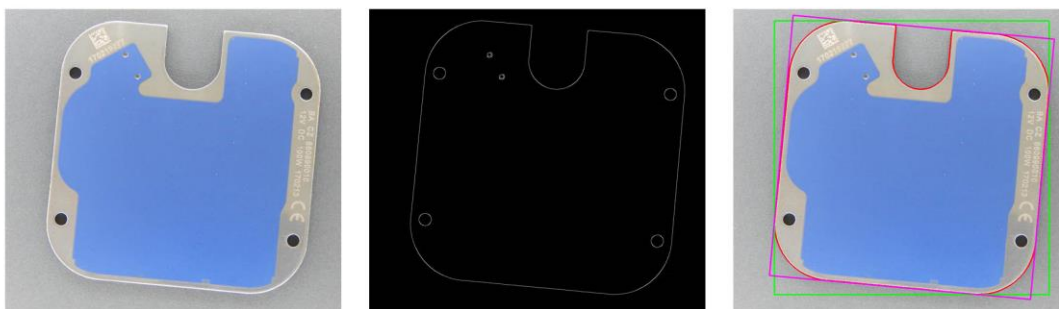
## 4.2 Aplikace pro zpracování obrazu

Za použití programovacího jazyka Java a knihovny OpenCV byla vytvořena aplikace pro další zpracování nasnímaných obrazů. Aplikace umožňuje provádět základní operace předzpracování obrazu, které jsou popsány v teoretické části této práce, tj. jasové modifikace obrazu, využití konvolučních filtrů, segmentační metody apod. Hlavním účelem aplikace je možnost vytvořit datové sady pro algoritmy strojového učení.



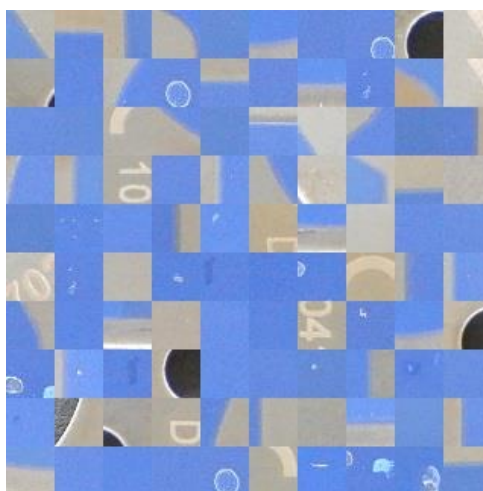
Obr. 12 – Hlavní okno aplikace

Nasnímané obrázky je potřeba před dalším zpracováním upravit a z obrazu je třeba vyjmout pouze oblast našeho zájmu. Pro extrakci objektu z obrazu je využito několika dílčích kroků. Nejprve je obraz rozostřen pomocí filtru s Gaussovým rozložením a poté proveden převod do odstínů šedi. Následně je na obraz aplikován Cannyho hranový detektor a použita funkce na vyhledání kontur v obraze. Ze znalosti tvaru objektu lze předpokládat, že největší nalezená kontura bude odpovídat právě hledanému tělesu. Při snímání se neubráníme drobnému natočení jednotlivých těles, proto je v poslední fázi prováděno otáčení do rovnoběžné polohy s okraji snímků a následně provedeno ořezání, viz Obr. 13.



**Obr. 13 – Postup vyhledání oblasti zájmu**

Protože v izolační vrstvě jsou sledované vady ve vzorcích většinou velmi malé, zvolil jsem metodu rozdělení obrazu na menší oblasti. Z jednoho obrazu je tedy možné vybrat stovky podoblastí, které rozdělují do čtyř základních tříd – vnější oblast (out), hrana (edge), oblast zájmu s pozitivním výskytem vady (pos) a oblast zájmu s negativním výskytem vady (neg). Jednotlivé podoblasti se dají pořizovat ve velikosti 32 x 32 nebo 20 x 20 pixelů. Ač se jedná o velmi malé oblasti, poskytují nám obrovské množství dat. Pro obraz o velikosti 32 x 32 pixelů dostáváme 3072 hodnot jasové funkce v barevném obraze, pro menší obraz 20 x 20 pixelů se jedná o 1200 hodnot. Tento počet atributů je pro jeden příklad datové sady celkem vysoký a fáze učení je v případě některých algoritmů časově velice náročná. Zjednodušíme datové sady s využitím nasnímaných obrazů pouze v odstínech šedi nebo průměrováním čtyř nebo šestnácti pixelů, případně využitím obou metod. Tímto způsobem získáme datové sady čítající stovky a tisíce testovacích dat. Data v obrazové podobě zachycuje Obr. 14.



**Obr. 14 – Testovací data v obrazové podobě**

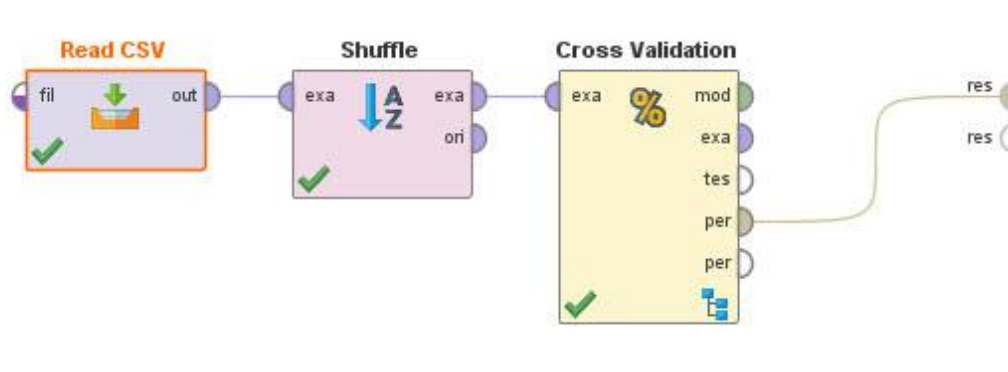
### 4.3 Testování datových sad

Pro vytváření datových modelů a jejich následného testování existuje řada aplikací, mezi které patří projekt Weka University of Waikato na Novém Zélandu (39) nebo uživatelsky přívětivější RapidMiner Studio (40). V obou aplikacích lze otestovat výkonnost a přesnost modelů naučených z nasbíraných datových sad za použití mnoha algoritmů strojového učení. Při testování byly využity následující modely: rozhodovací strom, náhodný les, gradient boosted trees a neuronová síť. Pro analýzu odporové vrstvy, tedy první fáze výroby bylo pořízeno 100 snímků, ze kterých byly vytvořeny 3 série malých obrázků a celkem 11 datových sad. Vlastnosti jednotlivých testovacích sad shrnuje Tabulka 4.

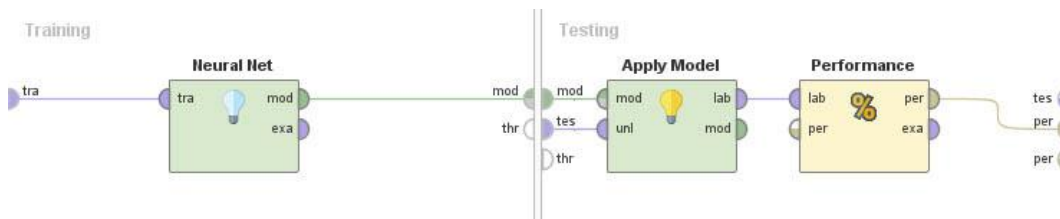
Tabulka 4 - Přehled vlastností datových sad

Datová sada	Počet příkladů	Velikost snímků	Počet atributů	Barevný model	Rozložení tříd (out/edge/neg/pos)
1	1200	32x32	64	šedotónový	300/300/300/300
2	1200	32x32	192	RGB	300/300/300/300
3	1200	32x32	265	šedotónový	300/300/300/300
4	6000	32x32	64	šedotónový	1815/1550/2070/565
5	6000	32x32	192	RGB	1815/1550/2070/565
6	6000	32x32	256	šedotónový	1815/1550/2070/565
7	10000	20x20	25	šedotónový	3500/2270/3500/730
8	10000	20x20	75	RGB	3500/2270/3500/730
9	10000	20x20	100	šedotónový	3500/2270/3500/730
10	10000	20x20	300	RGB	3500/2270/3500/730
11	10000	20x20	400	šedotónový	3500/2270/3500/730

V programu RapidMiner Studio jsem vytvořil model pro analýzu jednotlivých datových sad. V modelu načteme datovou sadu uloženou ve formátu csv a označíme, která část dat označuje atributy, neboli vstupní hodnoty, a která část tvoří výstupní hodnotu neboli label. Datová sada následně prochází procesem křížové validace, ve které máme nadefinovaný testovaný učicí algoritmus a měříme jeho výkonnost a přesnost, viz Obr. 15 a 16. Křížovou validaci provádíme na deseti skupinách, kdy je datový set rozdělen na 10 stejně velkých segmentů se stejným poměrným zastoupením jednotlivých tříd, 9 segmentů je použito pro trénování a jeden pro testování. Výsledná přesnost modelu je spočtena jako průměr z jednotlivých měření. Empiricky bylo zjištěno, že rozdělení do deseti skupin se jeví jako dostačující.



Obr. 15 – Diagram křížové validace



Obr. 16 – Diagram trénování a testování neuronové sítě

Celkem bylo provedeno 44 testů, čtyři testování pro každou datovou sadu. Samotné testování je kromě časové náročnosti také hardwarově celkem náročné, zejména na využití operační paměti. Trénování nakonec probíhalo na dvouprocesorovém serveru s 32GB operační paměti a i v této konfiguraci trénink nejnáročnějších sad trval i více než pět hodin. Zde se plně projeví nevýhoda RapidMiner Studia, kdy volně dostupná edice pracuje pouze s jedním logickým jádrem procesoru. Křížovou validaci s více než 400 atributy jsem neprováděl. Tabulky 5 až 15 shrnují naměřené hodnoty a výkonnost jednotlivých datových sad. Červeně jsou zvýrazněné nejpřesnější modely pro každou sadu.

Tabulka 5 - 1200 příkladů, 64 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
1	Rozhodovací strom	71.50% +/- 5.67%	4s	maximální hloubka 25
2	Náhodný les	84.00% +/- 2.00%	11s	50 stromů, maximální hloubka 25
3	Gradient boosted trees	92.50% +/- 2.39%	3m	50 stromů, maximální hloubka 25
4	Neuronová síť	92.25% +/- 1.97%	7m 53s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 6 - 1200 příkladů, 192 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
5	Rozhodovací strom	79.42% +/- 11.52%	23s	maximální hloubka 25
6	Náhodný les	84.25% +/- 4.12%	17s	50 stromů, maximální hloubka 25
7	Gradient boosted trees	93.67% +/- 1.91%	7m 49s	50 stromů, maximální hloubka 25
8	Neuronová síť	83.67% +/- 4.49%	25m 3s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 7 - 1200 příkladů, 256 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
9	Rozhodovací strom	67.25% +/- 10.89%	7s	maximální hloubka 25
10	Náhodný les	83.25% +/- 3.88%	14s	50 stromů, maximální hloubka 25
11	Gradient boosted trees	91.17% +/- 2.01%	8m 15s	50 stromů, maximální hloubka 25
12	Neuronová síť	91.08% +/- 2.39%	22m 2s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 8 - 6000 příkladů, 64 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
13	Rozhodovací strom	78.62% +/- 1.78%	20s	maximální hloubka 25
14	Náhodný les	78.37% +/- 1.37%	1m 56s	60 stromů, maximální hloubka 25
15	Gradient boosted trees	94.28% +/- 0.68%	10m 3s	60 stromů, maximální hloubka 25
16	Neuronová síť	93.12% +/- 0.98%	20m 1s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 9 - 6000 příkladů, 192 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
17	Rozhodovací strom	87.63% +/- 6.14%	52s	maximální hloubka 25
18	Náhodný les	87.47% +/- 0.38%	1m 30s	60 stromů, maximální hloubka 25
19	Gradient boosted trees	97.10% +/- 0.46%	25m 40s	60 stromů, maximální hloubka 25
20	Neuronová síť	76.95% +/- 14.60%	1h 28m 7s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 10 - 6000 příkladů, 256 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
21	Rozhodovací strom	78.55% +/- 1.69%	37s	maximální hloubka 25
22	Náhodný les	78.97% +/- 1.59%	2m 16s	60 stromů, maximální hloubka 25
23	Gradient boosted trees	93.73% +/- 0.74%	36m 29s	60 stromů, maximální hloubka 25
24	Neuronová síť	92.60% +/- 0.93%	1h 34m 42s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 11 - 10000 příkladů, 25 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
25	Rozhodovací strom	87.64% +/- 0.64%	11s	maximální hloubka 25
26	Náhodný les	84.51% +/- 0.64%	1m 41s	100 stromů, maximální hloubka 25
27	Gradient boosted trees	97.18% +/- 0.70%	10m 12s	100 stromů, maximální hloubka 25
28	Neuronová síť	94.77% +/- 0.98%	15m 31s	500 cyklů, 1 skrytá vrstva (15)



Tabulka 12 - 10000 příkladů, 75 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
29	Rozhodovací strom	87.80% +/- 2.75%	23s	maximální hloubka 25
30	Náhodný les	84.74% +/- 0.72%	3m 9s	100 stromů, maximální hloubka 25
31	Gradient boosted trees	94.37% +/- 0.76%	13m 11s	100 stromů, maximální hloubka 25
32	Neuronová síť	98.70% +/- 0.35%	1h 1m 37s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 13 - 10000 příkladů, 100 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
33	Rozhodovací strom	85.76% +/- 0.89%	20s	maximální hloubka 25
34	Náhodný les	84.59% +/- 0.93%	2m 47s	100 stromů, maximální hloubka 25
35	Gradient boosted trees	97.01% +/- 0.49%	26m 36s	100 stromů, maximální hloubka 25
36	Neuronová síť	96.14% +/- 0.71%	1h 25m 20s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 14 - 10000 příkladů, 300 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
37	Rozhodovací strom	74.54% +/- 5.96%	35s	maximální hloubka 50
38	Náhodný les	89.51% +/- 0.43%	5m 33s	100 stromů, maximální hloubka 25
39	Gradient boosted trees	98.46% +/- 0.37%	1h 11m 46s	100 stromů, maximální hloubka 25
40	Neuronová síť	81.29% +/- 13.59%	4h 7m 32s	500 cyklů, 1 skrytá vrstva (25)

Tabulka 15 - 10000 příkladů, 400 vstupních atributů, 4 třídy

Test	Model	Přesnost	Čas křížové validace	Parametry
41	Rozhodovací strom	84.64% +/- 1.30%	47s	maximální hloubka 25
42	Náhodný les	84.13% +/- 0.85%	6m 43s	100 stromů, maximální hloubka 25
43	Gradient boosted trees	96.65% +/- 0.43%	1h 29m 32s	100 stromů, maximální hloubka 25
44	Neuronová síť	95.60% +/- 0.64%	5h 51m 16s	500 cyklů, 1 skrytá vrstva (25)

#### 4.4 Shrnutí výsledků a další rozvoj aplikace

Při testování bylo zjištěno, že nejlepších výsledků pro predikci vad dosahují modely za použití algoritmů gradient boosted trees a neuronových sítí. Rozhodovací strom i náhodný les není pro predikci vad použitelný. Celková přesnost mezi 70 – 90% může být zajímavá, ale pokud přidáme matici záměn, jsou výsledky více než špatné. Tabulky 16 a 17 ukazují matici záměn nejlepších výsledků při použití rozhodovacího stromu a náhodného lesa.

Tabulka 16 - Matice záměn, 29. test

29	true edge	true neg	true out	true pos	class precision
pred. edge	1733	1	10	7	98.97%
pred. neg	473	3491	0	657	75.55%
pred. out	54	0	3490	0	98.48%
pred. pos	10	8	0	66	78.57%
class recall	76.34%	99.74%	99.71%	9.04%	

Tabulka 17 - Matice záměn, 38. test

38	true edge	true neg	true out	true pos	class precision
pred. edge	1938	7	1	25	98.33%
pred. neg	49	3493	0	684	82.65%
pred. out	283	0	3499	0	92.52%
pred. pos	0	0	0	21	100.00%
class recall	85.37%	99.80%	99.97%	2.88%	

V obou případech se celková přesnost modelu pohybuje na hranici 90%, ale klasifikace je dobrá pouze v případě označení vnější a vnitřní části. V predikci

klíčové třídy, tedy pozitivního výskytu vady, je přesnost zanedbatelná. Naopak, zbývající dva algoritmy jsou jednak mnohem přesnější a navíc algoritmus gradient boosted trees je při všech měření nejstabilnější. Neuronová síť dosahuje výsledků srovnatelných s gradient boosted trees, ale v několika případech je celková přesnost horší, než u rozhodovacího stromu nebo náhodného lesa. Tyto výkyvy mohou být způsobeny nevhodnou strukturou neuronové sítě pro daný počet atributů. Potvrzuje se, že s rostoucím počtem atributů se většinou nezlepšuje predikce, mnohem důležitější je počet příkladů, a proto datové sady obsahující 10000 příkladů mají lepší výsledky, než datové sady s 1200 a 6000 příklady. Nejlepšího výsledku ze všech 44 testovacích modelů bylo získáno za použití neuronové sítě. Jedná se o test č. 32. Tento model obsahuje 10000 příkladů, každý příklad má 75 atributů získaných z barevných obrazových segmentů 20 x 20 pixelů. Atributy byly vypočteny jako průměr z šestnácti pixelů ze čtverců o hraně 4 pixely. Výsledná přesnost modelu je 98.70% +/- 0.35%. Matici záměn znázorňuje Tabulka 18.

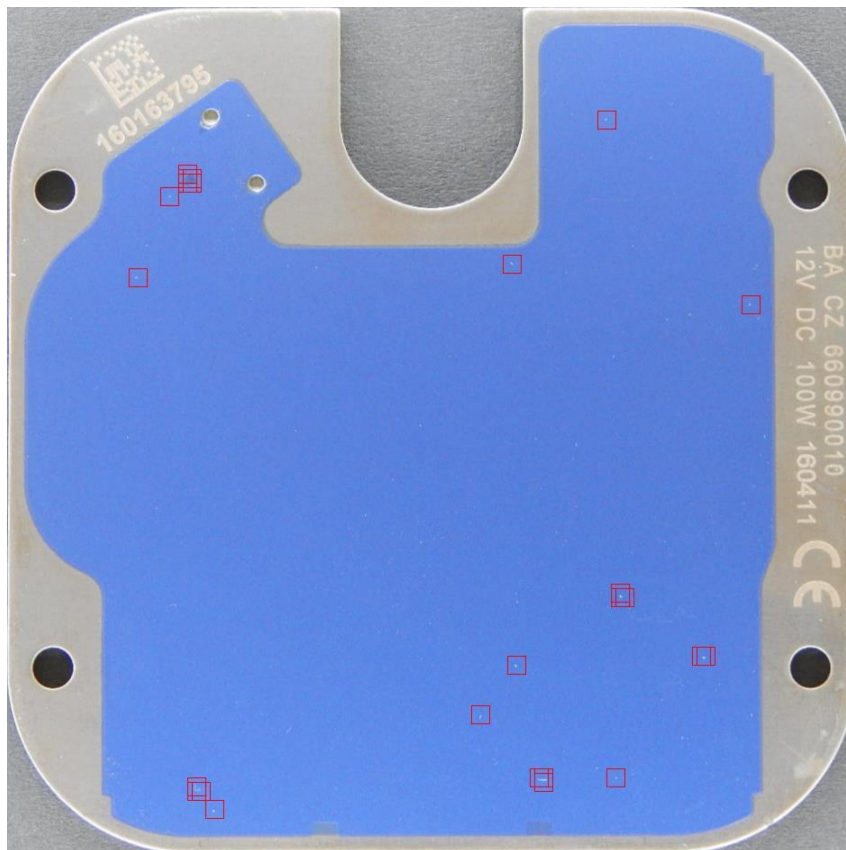
Tabulka 18 - Matice záměn, 32. test

32	true edge	true neg	true out	true pos	class precision
pred. edge	2254	2	3	3	99.65%
pred. neg	1	3489	0	97	97.27%
pred. out	12	0	3497	0	99.66%
pred. pos	3	9	0	630	98.13%
class recall	99.30%	99.69%	99.91%	86.30%	

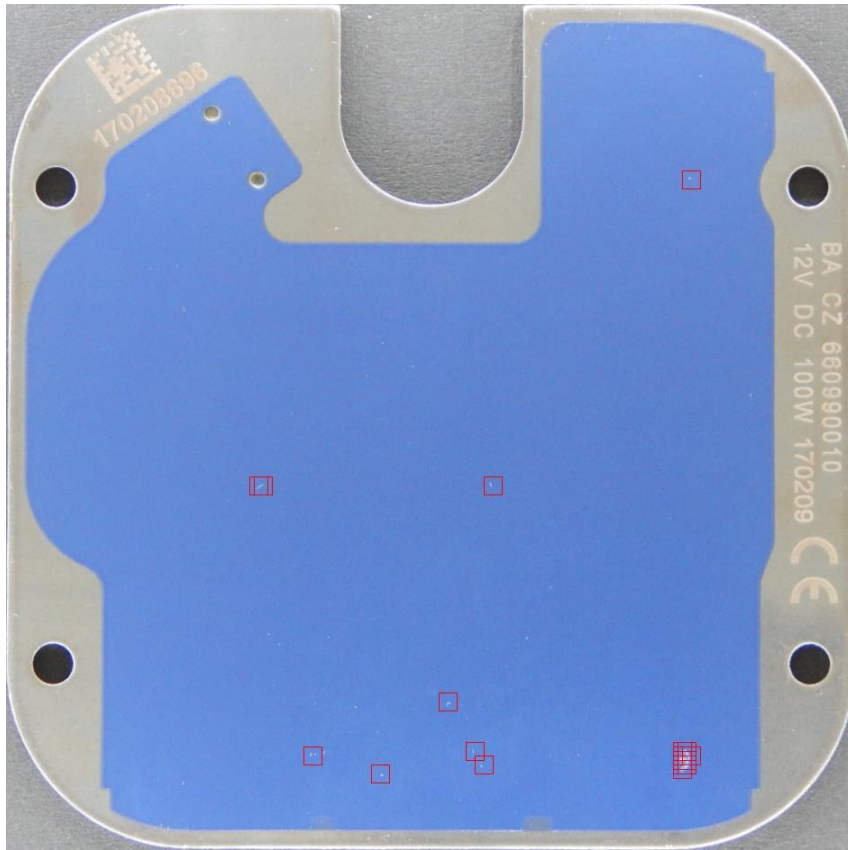
Model vykazuje pouze horší senzitivitu při predikci pozitivních segmentů, v predikci ostatních tříd je téměř bezchybný.

Na základě získaných výsledků z různých datových modelů byla do aplikace pro tvorbu datových sad přidána analýza pomocí neuronové sítě a následná možnost kontroly jednotlivých snímků. Základní možnosti nastavení neuronové sítě jsou totožné s nastavením sítě v prostředí RapidMiner Studia, jednotlivé parametry jako strukturu sítě, koeficient učení, počet iterací apod. lze libovolně měnit. Po naučení neuronové sítě se kontrola snímků provádí jednoduchým způsobem, kdy je po obraze posouváno okénko o velikosti naučených segmentů a každý segment obrazu je klasifikován. Velikost posunu okénka je nastavena na 5 pixelů. Velikost jednotlivých snímků je po ořezání cca 900 x 900 pixelů a v jednom obraze je tedy provedeno přibližně 35000 kontrol. Segmenty, které jsou klasifikovány jako pozitivní, jsou

ohraničeny červeným rámem, po kontrole je zobrazena informace o celkovém počtu zkontrolovaných segmentů a počet pozitivních nálezů. Tato funkce je pro rozvoj systému naprosto klíčová, v kontrolovaných snímcích okamžitě vidíme, v jakých případech naučená síť predikuje správně a v jakých se mýlí. Následně lze provádět korekce, chybně označené segmenty správně klasifikovat a přidat do nové datové sady. Následně neuronovou síť znovu natrénovat a sledovat, jak se mění klasifikace na testovaných snímcích. Lze tak vysledovat oblasti, kde má klasifikátor s predikcí problémy. Obecně to jsou oblasti kolem hran, kde se skokově mění hodnota jasové funkce a pak také specifické oblasti samotného topného tělesa, zejména oblasti pro napájení ERNI konektoru. Ukázky klasifikace zobrazují Obr. 17 a Obr. 18, kde lze považovat predikci vad za velmi dobrou.



Obr. 17 – Ukázka kontroly topného tělesa č. 1



Obr. 18 – Ukázka kontroly topného tělesa č. 2

## 5 NÁVRHY A DOPORUČENÍ

Pro další rozvoj systému na rozpoznávání vad by bylo vhodné zdokonalit systém samotného snímání a sběru dat. Kvalita nasnímaných dat je zásadní pro dobré fungování celého systému a v tomto směru má celý systém největší rezervy. Pro nasazení v reálném provozu by se pro snímání dala použít průmyslová kamera, kterých je v současné době na trhu nepřehledné množství. Zpracována je pouze malá oblast sledování kvality výrobku, tudíž je spousta možností dalšího rozvoje celého systému, ať se bude jednat o kontrolu vodivé a odporové vrstvy, správného osazení jednotlivých elektrotechnických prvků nebo kontrolu ostatních vyráběných topných těles. Kontrola izolační vrstvy, ač byla testována pouze na výrobcích jednoho typu, by se dala bez významnějších úprav použít i na dalších typech, neboť kontrola není zaměřena na zjišťování konkrétního tvaru tělesa, ale testuje nezávisle jednotlivé segmenty obrazu.

Otázkou zůstává, zda je dobré aplikaci nadále vyvíjet v prostředí Java. Oba systémy pro strojové učení a dolování dat, Weka i RapidMiner, jsou psány v jazyce Java, ale za nejpoužívanější jazyky pro vývoj strojového učení jsou pokládány jazyky R a Python.

## 6 ZÁVĚR

Cílem práce bylo zjistit, zda se počítačové vidění může uplatnit v průmyslové výrobě i pro detekci velice malých výrobních vad.

V práci je nejprve vysvětlen princip počítačového vidění a popsány jednotlivé fáze procesu zachycení a zpracování obrazu. Zároveň jsou rozvedeny základní principy strojového učení a popsány některé z algoritmů strojového učení, neboť tyto vědní oblasti se v současné době úzce prolínají. V práci je také uvedeno typické využití počítačového vidění v průmyslu, včetně dvou konkrétních příkladů fungujících systémů.

Práce se dále zabývá návrhem systému pro detekci drobných výrobních vad při výrobě tištěných topných těles v izolační vrstvě. Byla implementována aplikace pro zpracování nasnímaných obrazů a vytvoření datových sad pro testování v modelech strojového učení. Testování bylo prováděno v programu RapidMiner, kde je možné vyzkoušet řadu algoritmů strojového učení. Testováno bylo několik algoritmů vytvořených na základě rozhodovacích stromů a také, v současné době velice populární, neuronové sítě. Aplikace pro zpracování obrazu byla doplněna o trénování neuronové sítě a následné kontrolování topných těles.

Výsledné hodnoty jsou velice příznivé, při správné metodice je možné strojové vidění použít i při kontrole kvality výroby. Zpracována byla pouze jedna z mnoha operací, které jsou při kontrole kvality prováděny, a nezbývá než pokračovat v testování možností, které se uplatní v dalších fázích výroby. Momentálně jsou nejčastější problémy s kvalitou napájecích konektorů, kde se objevuje špatná rozteč a nerovnoběžnost osazení, další výzkum by se mohl věnovat této problematice.

## 7 SEZNAM POUŽITÉ LITERATURY

1. Papert S. The Summer Vision Project. 1. červenec 1966 [citován 31. červenec 2016]; Dostupné z: <http://dspace.mit.edu/handle/1721.1/6125>
2. Cifu, David X. Braddom's physical medicine & rehabilitation. Fifth edition. ISBN 978-0-323-28046-4.
3. Bunke, H. a Wang, P.S.P. Handbook of character recognition and document image analysis. Reprint. Singapore: World Scientific, 1997. ISBN 9789810222703.
4. Lowe DG. Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. 1999. s. 1150–7 roč.2.
5. Fei-Fei L, Perona P. A Bayesian hierarchical model for learning natural scene categories. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 2005. s. 524–31 roč. 2.
6. Suykens J a. K, Vandewalle J. Least Squares Support Vector Machine Classifiers. Neural Process Lett. 1. červen 1999;9(3):293–300.
7. Challenges in Machine Learning [Internet]. Challenges in Machine Learning. [citován 17. srpen 2017]. Dostupné z: <http://www.chalearn.org/>
8. Bradski, Gary and Kaehler, Adrian. Learning OpenCV Computer Vision with the OpenCV Library. Sebastopol: O'Reilly Media, 2008. ISBN 9780596554040.
9. Horák, Karel, et al. Počítačové vidění. Brno: Vysoké učení technické v Brně, 2008.
10. Šonka, Milan, Hlaváč, Vaclav a Boyle, Roger. Image processing, analysis, and machine vision. 3rd ed., International student edition. Toronto [u.a.]: Thomson, 2008. ISBN 0495244384.
11. Security U. Security camera - CCD vs CMOS image sensor [Internet]. Unifore - Poineer HD IP Camera/GSM Wi-Fi Alarm System Brand. [citován 17. srpen 2017]. Dostupné z: <https://www.unifore.net/analog-surveillance/security-camera-ccd-vs-cmos-image-sensor.html>
12. Hájovský, Radovan, Pustková, Radka, Kutálek, František. Zpracování obrazu v měřicí a řídicí technice: učební text : studijní materiály pro studijní obor Měřicí a řídicí technika, Elektronika Fakulty elektrotechniky a informatiky. Ostrava: Vysoká škola báňská - Technická univerzita, 2012. ISBN 978-80-248-2596-0.
13. Hlaváč, Vaclav a Sedláček, Miloš. Zpracování signálů a obrazů. Vyd. 2., přeprac. Praha: Vydavatelství ČVUT, 2005. ISBN 8001031101.
14. Histogram. In: Wikipedie [Internet]. 2016 [citován 15. srpen 2016]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Histogram&oldid=13591751>



15. Davies, E. R. Computer and machine vision: theory, algorithms, practicalities. 4th ed. Boston: Elsevier, 2012. ISBN 978-0-12-386908-1.
16. Segmentace obrazu. In: Wikipedie [Internet]. 2013 [citován 8. srpen 2016]. Dostupné z: [https://cs.wikipedia.org/w/index.php?title=Segmentace\\_obrazu&oldid=10988879](https://cs.wikipedia.org/w/index.php?title=Segmentace_obrazu&oldid=10988879)
17. Otsu N. A Threshold Selection Method from Gray-Level Histograms. IEEE Trans Syst Man Cybern. leden 1979;9(1):62–6.
18. Canny J. A Computational Approach to Edge Detection. IEEE Trans Pattern Anal Mach Intell. listopad 1986;PAMI-8(6):679–98.
19. Adams R, Bischof L. Seeded region growing. IEEE Trans Pattern Anal Mach Intell. červen 1994;16(6):641–7.
20. Pradhan, Ratika, et al. Contour line tracing algorithm for digital topographic maps. International Journal of Image Processing (IJIP), 2010, 4.2: 156-163.
21. Pavlidis, Theo. Algorithms for Graphics and Image Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982. ISBN 9783642932083.
22. Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. Int J Comput Vis. 1. leden 1988;1(4):321–31.
23. Chanussot J, Lambert P, Information-signal-image-vision CGI. Watershed approaches for color image segmentation. In: in: Proceedings of the IEEE Workshop on nonlinear signal and image processing. 1999.
24. Freeman H. On the Encoding of Arbitrary Geometric Configurations. IRE Trans Electron Comput. červen 1961;EC-10(2):260–8.
25. Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn. 1. květen 1992;8(3–4):229–56.
26. Quinlan JR. Induction of decision trees. Mach Learn. 1. březen 1986;1(1):81–106.
27. Random forests - classification description [Internet]. [citován 6. srpen 2017]. Dostupné z: [https://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm)
28. Freund, Yoav; Schapire, Robert; Abe, Naoki. A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence, 1999, 14.771-780: 1612.
29. Freund Y, Schapire RE. A desicion-theoretic generalization of on-line learning and an application to boosting. In: Computational Learning Theory [Internet]. Springer, Berlin, Heidelberg; 1995 [citován 27. červenec 2017]. s. 23–37. (Lecture Notes in Computer Science). Dostupné z: [https://link.springer.com/chapter/10.1007/3-540-59119-2\\_166](https://link.springer.com/chapter/10.1007/3-540-59119-2_166)

30. Frank, Eibe. Pruning decision trees and lists. 2000. PhD Thesis. University of Waikato.
31. Kačenka P. Matematický korespondenční seminář [Internet]. [citován 18. srpen 2017]. Dostupné z: <https://mks.mff.cuni.cz/library/library.php?categ=60&supcats=10>
32. Volná, Eva. Neuronové sítě 1. Ostravská univerzita v Ostravě, 2008.
33. Fawcett T. An introduction to ROC analysis. Pattern Recognit Lett. 1. červen 2006;27(8):861–74.
34. Automa: odborný časopis pro automatizační techniku. Praha 8: FCC Public, 1995-. ISSN 1210-9592.
35. Sun T-H, Tseng C-C, Chen M-S. Electric contacts inspection using machine vision. Image Vis Comput. červen 2010;28(6):890–901.
36. Wang L, Zhao Y, Zhou Y, Hao J. Calculation of flexible printed circuit boards (FPC) global and local defect detection based on computer vision. Circuit World. květen 2016;42(2):49–54.
37. OpenCV library [Internet]. [citován 9. srpen 2017]. Dostupné z: <http://opencv.org/>
38. Baggio, Daniel Lelis. OpenCV 3.0 computer vision with Java. Birmingham, UK: Packt Publishing Limited, 2015. ISBN 978-1-78328-397-2.
39. Weka 3 - Data Mining with Open Source Machine Learning Software in Java [Internet]. [citován 9. srpen 2017]. Dostupné z: <http://www.cs.waikato.ac.nz/ml/weka/>
40. Data Science Platform [Internet]. RapidMiner. [citován 9. srpen 2017]. Dostupné z: <https://rapidminer.com/>

## 8 SEZNAM OBRÁZKŮ

Obr. 1 – Obraz jako matice čísel, převzato z (8).....	4
Obr. 2 – Sekvence úkonů zpracování obrazu, převzato z (9) .....	5
Obr. 3 – CCD vs. CMOS, převzato z (11).....	6
Obr. 4 – Modifikace jasové stupnice, převzato z (9) .....	9
Obr. 5 – Konvoluční filtry, převzato z (13) .....	11
Obr. 6 – Underfitting a overfitting, převzato z (8).....	19
Obr. 7 – Struktura neuronu s logistickou aktivační funkcí.....	24
Obr. 8 – Schéma neuronové sítě s jednou skrytou vrstvou .....	25
Obr. 9 – Povrchové vady elektrických kontaktů, převzato z (35).....	30
Obr. 10 – Schéma tištěného topného tělesa .....	33
Obr. 11 – Ukázka vad v izolační vrstvě .....	36
Obr. 12 – Hlavní okno aplikace .....	37
Obr. 13 – Postup vyhledání oblasti zájmu .....	38
Obr. 14 – Testovací data v obrazové podobě.....	38
Obr. 15 – Diagram křížové validace .....	40
Obr. 16 – Diagram trénování a testování neuronové sítě .....	40
Obr. 17 – Ukázka kontroly topného tělesa č. 1 .....	46
Obr. 18 – Ukázka kontroly topného tělesa č. 2.....	47

## 9 SEZNAM TABULEK

Tabulka 1 - Chyby v modelech a možná řešení .....	19
Tabulka 2 - Matice záměn .....	26
Tabulka 3 - Soupis sledovaných vad.....	34
Tabulka 4 - Přehled vlastností datových sad .....	39
Tabulka 5 - 1200 příkladů, 64 vstupních atributů, 4 třídy .....	40
Tabulka 6 - 1200 příkladů, 192 vstupních atributů, 4 třídy .....	41
Tabulka 7 - 1200 příkladů, 256 vstupních atributů, 4 třídy .....	41
Tabulka 8 - 6000 příkladů, 64 vstupních atributů, 4 třídy .....	41
Tabulka 9 - 6000 příkladů, 192 vstupních atributů, 4 třídy .....	42
Tabulka 10 - 6000 příkladů, 256 vstupních atributů, 4 třídy .....	42
Tabulka 11 - 10000 příkladů, 25 vstupních atributů, 4 třídy .....	42
Tabulka 12 - 10000 příkladů, 75 vstupních atributů, 4 třídy .....	43
Tabulka 13 - 10000 příkladů, 100 vstupních atributů, 4 třídy .....	43
Tabulka 14 - 10000 příkladů, 300 vstupních atributů, 4 třídy .....	43
Tabulka 15 - 10000 příkladů, 400 vstupních atributů, 4 třídy .....	44
Tabulka 16 - Matice záměn, 29. test .....	44
Tabulka 17 - Matice záměn, 38. test .....	44
Tabulka 18 - Matice záměn, 32. test .....	45

## 10 PŘÍLOHY

Přiložené DVD obsahující:

- Zdrojové kódy aplikace pro analýzu obrázků
- Samostatně spustitelný program, manuál k ovládání programu
- Datové sady v obrazové i tabulkové podobě
- Testovací snímky
- Kompletní výsledky jednotlivých cross-validačních testů

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Málek Josef	Topolská 751, Chrudim - Chrudim II	I14523

**TÉMA ČESKY:**

Počítačové vidění v průmyslové výrobě

**TÉMA ANGLICKY:**

Computer Vision in Industrial Production

**VEDOUcí PRÁCE:**

Mgr. Jan Vaněk, Ph.D. - KIKM

**ZÁSADY PRO VYPRACOVÁNÍ:**

Cíl práce:

Prostudovat metody zpracování obrazu, navrhnout systém pro rozpoznávání vad na tištěných tompných tělesech, systém implementovat a otestovat.

Postup prací:

Rešerše literatury  
Návrh systému  
Implementace  
Testování

**SEZNAM DOPORUČENÉ LITERATURY:**

DAVIES, E. R. Computer and machine vision: theory, algorithms, practicalities. 4th ed. Waltham, Mass.: Elsevier, 2012. ISBN 978-0123869081.

BRADSKI, Gary R. Learning OpenCV. Sebastopol: O'Reilly, 2008. ISBN 978-0-596-51613-0.


Podpis studenta:



Datum:

27.10.2016

Podpis vedoucího práce:



Datum:

27.10.2016