

České učení technické v Praze
Fakulta strojní
Ústav přístrojové a řídicí techniky



Rozpoznávání unikátního označení vývalků pomocí strojového vidění

Diplomová práce

Karel Riedl

Magisterský program: Automatizační a přístrojová technika
Magisterský obor: Automatizace a průmyslová informatika
Vedoucí práce: Ing. Matouš Cejnek, Ph.D.

Praha, červen 2022

Zadání

Vedoucí práce:

Ing. Matouš Cejnek, Ph.D.
Ústav přístrojové a řídicí techniky
Fakulta strojní
České vysoké učení technické v Praze
Technická 2
160 00 Praha 6
Česká republika

červen 2022 Karel Riedl

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s tím, že její výsledky mohou být dále použity podle uvážení vedoucího diplomové práce jako jejího spoluautora. Souhlasím také s případnou publikací výsledků diplomové práce nebo její podstatné části, pokud bude uveden jako její spoluautor.

V Praze 2022

.....
Karel Riedl

Abstrakt

Abstrakt

Předmětem této diplomové práce je návrh a vývoj stanoviště automatického kamerového rozpoznání unikátního identifikátoru hutního polotovaru - sochoru. V rámci této diplomové práce je navrženo optické pracoviště a návrh a vývoj algoritmů pro automatické rozpoznávání unikátních identifikátorů na čelech železných sochorů po dalším tepelném zpracování. Identifikace se provádí za studena. Unikátní identifikátory jsou textové řetězce vyražené metodou "dot matrix" (důlkování podle definovaných vzorů).

Klíčová slova: Optické rozpoznávání znaků, Python, OpenCV

Abstract

The subject of this diploma thesis is the design and development of an automatic camera recognition station for a unique identifier of a metallurgical semi-finished product - a billet. Within this diploma thesis, an optical workplace is designed and an algorithm is developed for automatic recognition of unique identifiers on the faces of iron billets after further heat treatment. Identification is performed on a cold billet. Unique identifiers are text strings imprinted using the "dot matrix" method (to dimple according to defined patterns).

Keywords: Optical character recognition, Python, OpenCV

Poděkování

Rád bych touto cestou vyjádřil poděkování panu Ing. Matouši Cejnkovi, Ph.D. za jeho cenné rady a trpělivost při vedení mé bakalářské. Dále bych rád poděkoval panu Ing. Jiřímu Čápovi, Ph.D. za pomoc s výběrem osvětlení, kamery, objektivu a konstrukcí optického stanoviště. Rád bych poděkoval také své rodině, přítelkyni a kamarádům, kteří mi při vytváření této práce pomohli a podpořili mě. Bez jejich pomoci by nebylo možné práci dokončit.

Seznam obrázků

2.1	Grafické znázornění konvoluce [8]	3
2.2	Originální obraz(vlevo) a jeho histogram (vpravo)	4
2.3	Obraz se zvýšeným kontrastem(vlevo) a ekvalizovaný histogram (vpravo)	5
2.4	Obraz s Gaussovským šumem (vlevo) a šumem salt-and-pepper (vpravo) [10]	6
2.5	Porovnání originálního obrázku (vlevo) a filtrovaného (vpravo)	8
2.6	Segmentace obrazu a jeho převedení do grafu souvislostí regionu [10]	9
2.7	Porovnání originálního obrázku (vlevo) a ideálně prahovaného (vpravo)	10
2.8	Porovnání nízkého prahu (vlevo) a vysokého prahu (vpravo)	10
2.9	Porovnání rovnice 2.2 (vlevo) a rovnice 2.3 (vpravo)	11
2.10	Jasně rozdělitelný histogram [7]	11
2.11	Histogramy aproximovány pomocí normálního rozdělení (a) a jejich skutečné histogramy se znázorněným běžným prahem [9].	12
2.12	Snímky MR. V levém sloupci jsou původní snímky, další jsou snímky segmentované. Vlevo: Částečně objemová mapa šedé hmoty. Uprostřed: Částečně objemová mapa bílé hmoty. Vpravo: Částečně objemová mapa mozkomíšního moku [9].	13
2.13	Histogram snímku MR. (a) Původní histogram. (b) Gaussova křivka přizpůsobená původními histogramu. (c) Gaussovské rozdělení odpovídající zleva: mozkomíšní mok, šedá hmota, bílá hmota [9].	13
2.14	Ukázka izotropické dilatace (a) a eroze (b) [9]	15
2.15	Nalezení obrysu pomocí eroze [9]	15
2.16	Otevření (originál vlevo) [9]	17
2.17	Uzavření (originál vlevo) [9]	17
2.18	Šablona písmena R je vyhledávána v levém obrazu. Vpravo je graficky znázorněna hodnota E (černá je pro E nulové, tedy nejlepší schodu). Je zřejmé, že při natočení, nebo změně velikosti je E větší [9].	18
2.19	Ukázka gradientu Ψ a směru Φ [9]	19
2.20	Typické tvary hran [9]	19
3.1	Fotky stanoviště v železárně. Pohled odkud sochor vyjíždí (vlevo) a kam zajíždí (vpravo)	22
3.2	Model stanoviště zakrytý plechovým krytem	22
3.3	Stanoviště v podobném úhlu jako obrázek 3.1 vpravo	23
3.4	Pohledy směrem od vývalku (vlevo) a směrem od kamery (vpravo)	23
3.5	Diagram posloupnosti stanoviště	24
3.6	Diagram struktury programu	25
3.7	Vzorek s čelním osvětlením	26

3.8	Úvodní pokusy osvětlení pomocí laseru ve tmě (vlevo) za silného okolního osvětlení (vpravo)	27
3.9	Úvodní pokus s osvětlením ze všech stran	28
3.10	Prototyp stanoviště	29
3.11	Jedna z iterací stanoviště	30
3.12	Jeden důlek	31
3.13	Snímek ze vzdálenosti 105 cm od čela, 150 cm kolmo od osy	32
3.14	Snímek ze vzdálenosti 180 cm od čela, 50 cm kolmo od osy	33
3.15	Objektiv	33
3.16	Kamera na stativu	34
3.17	Pohled na vývalek od kamery	35
3.18	Foto celého stanoviště	35
3.19	Původní snímky vývalků 4 (vlevo) a 5 (vpravo)	36
3.20	Zprůměrované snímky vývalků 4 (vlevo) a 5 (vpravo)	37
3.21	Porovnání kvality snímků	37
3.22	Původní fotka	38
3.23	Původní hodnoty jasu (vlevo) a zprůměrované hodnoty (vpravo)	38
3.24	Oříznutý snímek	39
3.25	Filtrovaný snímek (vlevo) a prahovaný snímek (vpravo)	39
3.26	Prahování původních snímků (Prahy byly vybrány individuálně a pro nejlepší výsledky)	40
3.27	Úprava fotky pomocí sobelova operátoru	40
3.28	Úprava pomocí hledání černé matice 4x4	41
3.29	Prahovaný obraz po hledání šablony	41
3.30	Prahovaný obraz po hledání šablony 3.12	42
3.31	Původní snímek (vlevo) a výsledek s koncovou sumou (vpravo)	43
3.32	Obrázky po konvoluci	43
3.33	Prahované původní snímky	44
3.34	Snímky s odstraněnými velkými konturami	44
3.35	Snímky po operaci otevírání	45
3.36	Snímek s uměle přidaným logem	46
3.37	Šablony pro vyhledávání v obraze	46
3.38	Nalezení těžišť obrazů	47

Seznam zkratk

HW Hardware. 21

LED Elektroluminiscenční dioda. 27, 29, 32

PC Osobní počítač. 1, 21, 24, 31, 48

PLC Programovatelný logický automat. 24

PNG Přenosná síťová grafika. 36

TIFF Souborový formát pro ukládání rastrové počítačové grafiky. 36

USB Univerzální sériová sběrnice. 24, 31, 48

Obsah

Zadání	ii
Prohlášení	iv
Abstract	v
Poděkování	vi
Seznam obrázků	vii
Seznam zkratk	ix
1 Úvod	1
1.1 Související práce	2
2 Pojmy ze strojového vidění	3
2.1 Konvoluce	3
2.2 Histogram	4
2.2.1 Ekvalizace histogramu	4
2.3 Šum	5
2.4 Vyhlazování obrazu	6
2.4.1 Průměrování	6
2.4.2 Mediánový filtr	7
2.5 Segmentace	8
2.5.1 Prahování	9
2.5.2 Základní výběr prahu	11
2.5.3 Optimální prahování	12
2.6 Morfologické operace	14
2.6.1 Dilatace a eroze	14
2.7 Vyhledávání v obraze	17
2.7.1 Vyhledávání šablony v obraze	18
2.7.2 Detekce hran	18
2.8 Python	20
2.8.1 Knihovna OpenCV	20
3 Praktická část	21
3.1 Stanoviště	21
3.1.1 Vývalky	23
3.2 Struktura úlohy	24
3.2.1 Struktura programu	25

3.3	Úvodní pokusy	25
3.3.1	Struktura programu	26
3.3.2	Osvětlení	26
3.3.3	Kamera	30
3.3.4	Úhel kamery	31
3.3.5	Objektiv	33
3.4	Testování v laboratoři	34
3.4.1	Finální rozložení	34
3.4.2	Úprava fotek	36
3.4.3	Nalezení vývalku	37
3.4.4	Zvýraznění důlků	39
3.5	Hodnocení zvýraznění důlků	45
3.6	Orientace vývalku	45
3.6.1	Separace číslic	47
4	Závěr	48
	Bibliografie	49

Kapitola 1

Úvod

Jako téma své diplomové práce jsem si vybral rozpoznávání unikátního označení vývalků pomocí strojového vidění. Strojové vidění je relativně mladá disciplína, která se poslední době velmi rychle rozšiřuje do všech oblastí průmyslu. Toto téma jsem si vybral nejen proto, že v dnešní době je snaha o úplnou autonomnost výrobních procesů, kde strojové vidění hraje velkou roli, ale i pro široký rozsah využití této technologie. Strojové vidění se nemusí využívat jen pro velké automatizační projekty, díky relativně nízké ceně se dá tato technologie využít i pro osobní účely.

V hutí Poldi, pro kterou se tento projekt zpracovává, jsou jednotlivé vývalky (sochory) značeny unikátním identifikátorem metodou tzv. důlkování arabských číslic za účelem sledování jednotlivých kusů. Tyto identifikátory jsou v současnosti čteny lidskou obsluhou. Celý proces výroby je z velké části automatizován. Lidská obsluha je oproti automatickým systémům obecně nákladnější a méně spolehlivá. Další výhodou strojové (počítačové) identifikace je i automatické odeslání informace do procesní databáze (integrace se zbytkem systému). A opět s vyšší spolehlivostí (stroj nebude dělat překlepy a nebude zapomínat) a samozřejmě to realizuje řádově rychleji. Jde současně o činnost v zásadě rutinní, stereotypní a monotónní. Cílem je tedy nahradit operaci čtení kódu operátorem strojovou identifikací za pomoci metod strojového vidění z důvodů všeobecné optimalizace.

V rámci této diplomové práce budu vymýšlet celé optické stanoviště pro rozpoznávání značení. Pod toto zadání spadá návrh modifikace místa v továrně pro ideální světelné podmínky, návrh osvětlení, vybrání hardware (kamera, objektiv, PC) a návrh samotného algoritmu rozpoznávání. Pro algoritmus bude využito programovacího jazyku Python, jehož nedílnou součástí bude knihovna OpenCV. Pro návrh stanoviště a výběr hardware bude využito kombinace experimentů a apriorních znalostí.

1.1 Související práce

Je mnoho přístupů k optickému rozpoznávání znaků, ve kterých jsou využity rozdílné způsoby, jak získat text z obrazu. Autoři [1] využívají k rozpoznání data spotřeby na plastových lahvích 7 kroků úpravy obrazu před tím, než jednotlivé znaky vyhodnocují pomocí konvoluční neuronové sítě. Zajímavé v této práci je, že k nalezení oblasti s textem dochází před jakoukoliv úpravou obrazu. V řízeném prostředí a s rozdílnými fonty dosahují až 90% přesnosti. Dalšího přístupu k rozpoznávání znaků je využito v článku [2]. Je uvedeno, že metoda propojování jednotlivých teček je stále velmi neefektivní, jelikož v některých případech tato metoda některé znaky ještě více degraduje a je mnohem obtížnější je rozpoznat. Navrhují řešení pomocí metody podpůrných vektorů, kde každý znak má svůj charakteristický 2D vektor, kde jeho velikost je určena množstvím teček a jejich posunem. Metoda podpůrných vektorů je uznávána za její robustnost, ale metody využívajících neuronových sítí jsou mnohem jednodušší na učení.

Autoři v [3] představují revoluční nástroj pro optické rozpoznávání znaků vyvinutý ve společnosti HP s názvem Tesseract-OCR. Tento volně dostupný software dokáže rozpoznat více než 100 jazyků a funguje dokonce v nevyhovujících podmínkách, jako jsou spojené znaky, neúplné znaky, nerovnoměrné mezery mezi znaky a sklon textu. Autoři [4] navrhli mobilní aplikaci využívající Tesseract-OCR a porovnávají ji s výsledky s jinými volně dostupnými nástroji pro optické rozpoznávání znaků: NHocr [5] a OCRopus [6].

Tesseract rozpoznává největší množství typů znaků a má nejlepší přesnost rozpoznávání, ale stále trpí problémy, které je třeba vyřešit předběžným zpracováním obrazu. Nevýhodou tohoto přístupu je, že "dot-matrix" znaky jsou špatně rozpoznávány, dokonce ani po předběžném zpracování a po rekonstrukci znaků není výsledek zcela uspokojivý. Protože "dot-matrix" znaky nemají společné rysy se znaky s žádným jiným typem znaků, přístup Tesseractu selhává při rozpoznávání tečkovaných textů z obrázku.

Kapitola 2

Pojmy ze strojového vidění

2.1 Konvoluce

Konvoluce je velmi užitečná a často využívaná ve strojovém vidění. Konvoluce, značená jako $*$, je definována pro dvě funkce $f(x)$ a $g(x)$ jako:

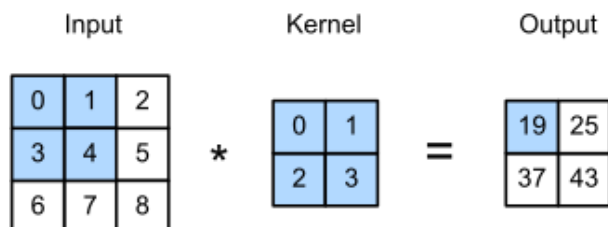
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(u)g(x - u) du$$

Při použití konvoluce na obrázek je potřeba tuto rovnici upravit. Jelikož mají obrázky dvě dimenze, je potřeba použít dvojný integrál. Data obrázku jsou diskrétní, takže se použije sumace místo integrálu. Tím získáme rovnici tuto:

$$F(x, y) = f(x, y) * g(x, y) = \sum_i \sum_j f(i, j)g(x - i, y - j)$$

Kde g je nyní označováno jako prostorová konvoluční maska.

Tato funkce je velmi užitečná například na hledání hran, nebo při vyhlazování obrázků [7].



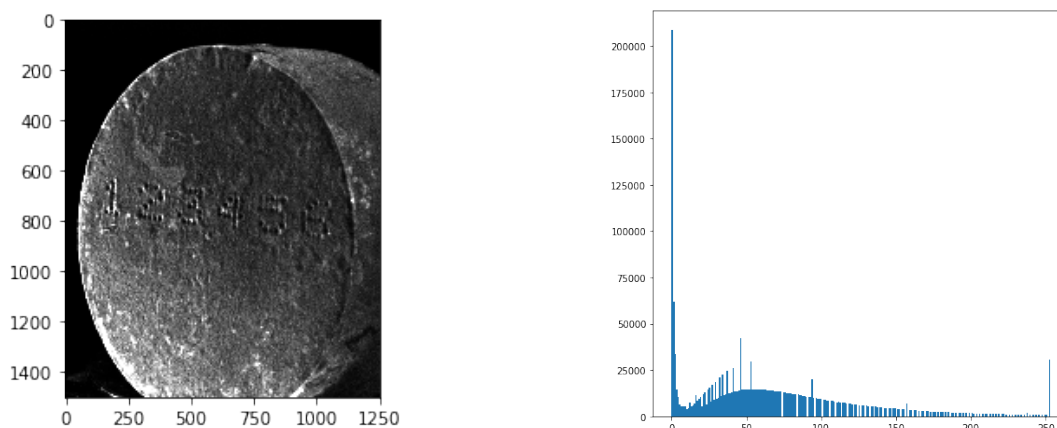
Obrázek 2.1: Grafické znázornění konvoluce [8]

Často se maska volí s lichým počtem sloupců a řádků, aby mohl reprezentativní bod ležet uprostřed.

2.2 Histogram

Histogram obrazu ve stupních šedi popisuje četnost jednotlivých hodnot jasu. Histogram obrazu je reprezentován 1-D polem, kde počet prvků je vázán k počtu úrovní šedi.

Histogram bývá jedinou dostupnou globální informací o obraze. Lze ho využít pro nalezení ideálních světelných podmínek nebo pro segmentaci. Jelikož pozice pixelů nemá na histogram žádný vliv, může několik obrazů mít stejný histogram. Při posunutí objektu po konstantním pozadí je získán identický histogram. Nejčastěji se vykresluje jako sloupcový graf.



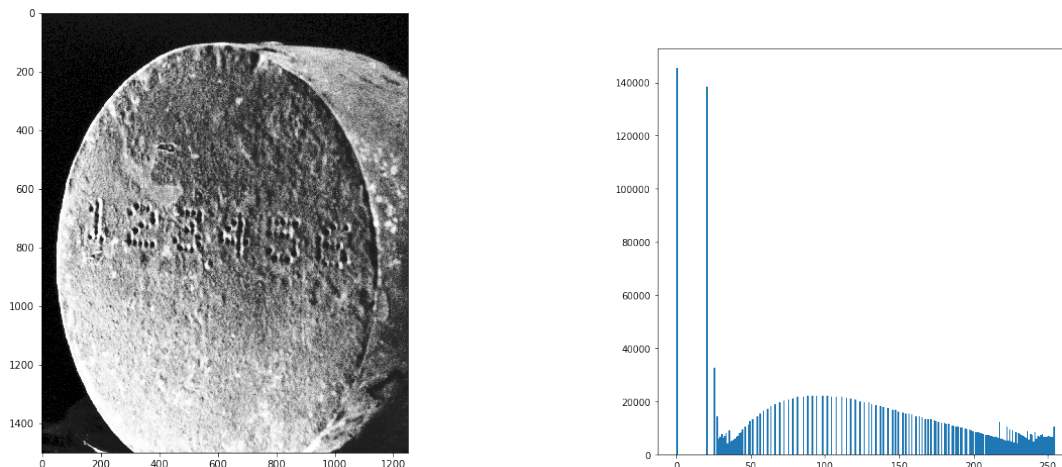
Obrázek 2.2: Originální obraz(vlevo) a jeho histogram (vpravo)

Z obrázku 2.3 je viditelné prominentní zastoupení černé (nulová hodnota) [9].

2.2.1 Ekvalizace histogramu

Ekvalizace histogramu patří mezi transformace v úrovních šedi. U těchto transformací nezávisí na pozici pixelu, ale pouze se transformují hodnoty jednotlivých pixelů podle jejich úrovně jasu. Díky tomu, že obrazy mají omezené množství úrovní šedi, se dá transformace provést pomocí výpočetně nenáročné vyhledávací tabulky.

Tyto transformace nejsou potřebné pro strojové vidění, ale využívají se hlavně pro lepší viditelnost lidským okem. Pro zvýšení kontrastu je prováděno většinou automaticky pomocí ekvalizace histogramu. Cílem je získat stejnou četnost odstínů šedé přes celou škálu. Ekvalizací histogramu je získán vyšší kontrast okolo maxim histogramu, ovšem nižší kontrast okolo minim histogramu.



Obrázek 2.3: Obraz se zvýšeným kontrastem(vlevo) a ekvalizovaný histogram (vpravo)

Z obrázku je viditelné, že spousta detailů, dříve pro lidské oko neviditelná, je teď mnohem zřetelnější. Histogram přišel o minima a hodnoty jsou rovnoměrněji rozprostřené [9].

2.3 Šum

Reálné obrazy jsou často degradovány některými náhodnými chybami – tato degradace se nazývá šum. Šum se může objevit během snímání, přenosu nebo zpracování obrazu a může být závislý nebo nezávislý na obsahu obrázku. Šum se obvykle popisuje pomocí jeho pravděpodobnostních charakteristik. Idealizovaný šum, tzv. bílý šum má konstantní výkonové spektrum, což znamená, že jsou přítomny všechny frekvence šumu a mají stejnou intenzitu. Například intenzita bílého šumu neklesá s rostoucí frekvencí, jako je typické pro signály v reálném světě. K modelování nejhorších případů degradace se často používá bílý šum, díky tomu se výpočty zjednoduší.

Speciálním případem bílého šumu je šum Gaussovský. Tento typ šumu následuje křivku normálního rozdělení a je velmi častý v reálném světě.

Když je obraz posílán ze senzoru dále nějakou cestou, vzniká šum nezávislý na našem obraze, tzv. aditivní šum popisován jako:

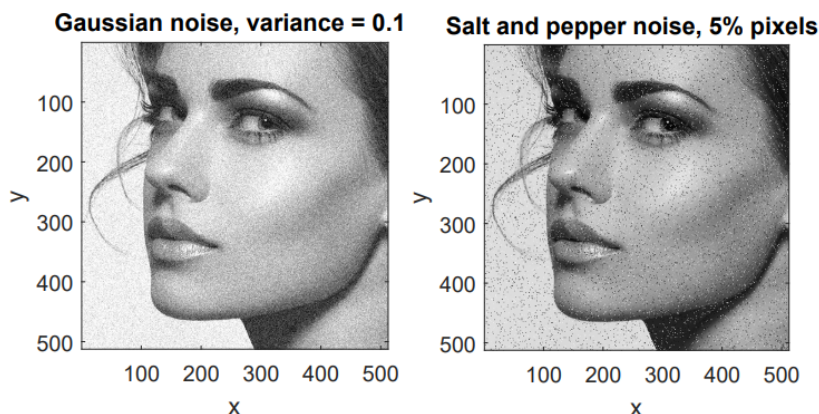
$$f(x, y) = g(x, y) + v(x, y),$$

kde šum v a obraz g jsou nezávislé proměnné.

Šum také může být závislý na obraze. Jedná se pak o multiplikatívni šum:

$$f = gv.$$

Mezi další typy šumu patří například kvantizační šum, který vzniká při nedostatečném množství odstínů barev, impulzní, který vzniká, když jednotlivé pixely mají diametrálně jinou hodnotu, než pixely sousedící. Při velké četnosti se nazývá "salt-and-pepper noise" (sůl a pepř, z obrázku je očividný) [9].



Obrázek 2.4: Obraz s Gaussovským šumem (vlevo) a šumem salt-and-pepper (vpravo) [10]

2.4 Vyhlazování obrazu

V obrazu se často vyskytuje šum (nežádané náhodné změny obrazu). Nejčastěji šum vzniká změnou osvětlení, šumem ze senzoru kamery nebo kompresí obrazu. Pomocí vyhlazování se tohoto šumu můžeme zbavit [9], [7].

Vyhlazování může být provedeno pomocí pořízení několika obrazů neměnné scény. Průměrují se hodnoty stejného pixelu v několika snímcích. Díky tomu se zbavíme šumu bez rozmazání obrazu. Nejmenší počet snímků bývá kolem 50.

Pokud ovšem máme snímek pouze jeden, musíme využít nadbytečnosti informací v obraze. Potlačení šumu vzniká využitím pixelů v okolí.

2.4.1 Průměrování

Nezákladnější verzí je průměrování. Vezmeme matici:

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

pomocí které zprůměrujeme hodnoty zvoleného obrázku. Vždy tedy vezmeme okno (v tomto případě 3x3), které "přiložíme" na oblast 3x3 zvoleném obraze a tím získáme jednu hodnotu v obraze novém. Můžeme si tedy představit, že námi zvolená matice h tedy

postupně, pixel po pixelu, putuje naším obrazem, jak horizontálně, tak vertikálně. Získáme tím tedy obraz nový, již vyhlazený. Matice jsou obdélníkového tvaru s lichým počtem sloupců a řádků.

Vezmeme tedy jednu 3x3 oblast z obrazu:

$$\begin{bmatrix} 90 & 100 & 85 \\ 76 & 90 & 40 \\ 102 & 102 & 98 \end{bmatrix}$$

a zprůměrováním získáme jednu novou hodnotu:

$$\begin{bmatrix} x & x & x \\ x & 87 & x \\ x & x & x \end{bmatrix}$$

Tento proces opakujeme, dokud nezískáme novou hodnotu pro každý pixel.

Matici h si můžeme také libovolně zvolit, například tak, abychom zvýšili důležitost středního pixelu:

$$h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

2.4.2 Mediánový filtr

Když uvažujeme konečnou množinu reálných čísel, která je seřazená buď vzestupně nebo sestupně, tak medián je hodnota uprostřed. Pro matici:

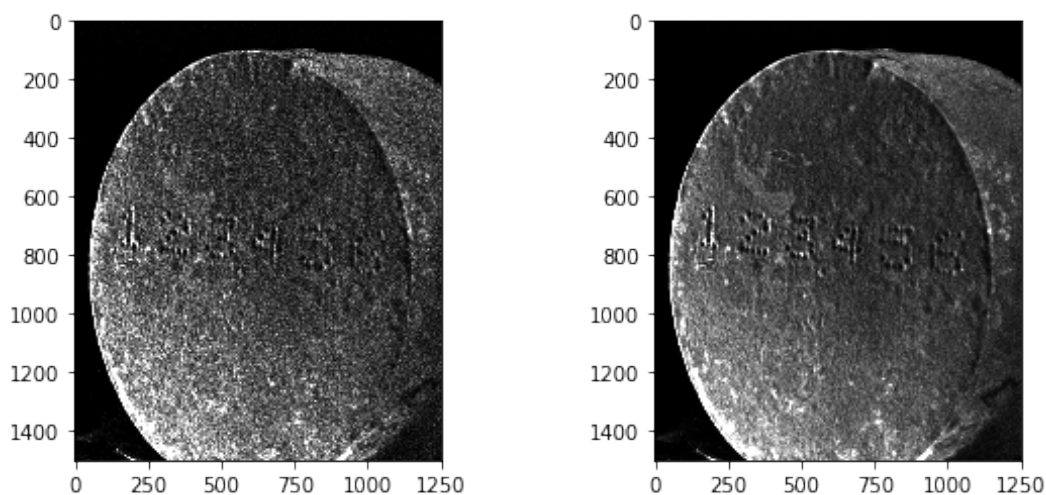
$$A = \begin{bmatrix} 90 & 100 & 85 \\ 76 & 90 & 40 \\ 102 & 102 & 98 \end{bmatrix}$$

je medián $M = 90$. Když to porovnáme s pouhým průměrováním, tak se na první pohled zdají tyto metody velmi podobné, ovšem tato metoda dokáže vyřadit extrémní, tedy velmi málo pravděpodobné veličiny. Pro matici:

$$B = \begin{bmatrix} 90 & 100 & 85 \\ 76 & 90 & 40 \\ 102 & 102 & 0 \end{bmatrix}$$

vidíme, že průměrná hodnota klesla na $\bar{x} = 76$. Medián zůstává stejný $M = 90$. Další výhodou je, že stačí částečné uspořádání posloupnosti. To znamená, že po spočítání mediánu a posunutí okna o jeden pixel není třeba řadit znovu všechny prvky. Nahradí se pouze změněné pixely a tím se sníží výpočetní náročnost [9].

Nevýhodou mediánového filtru je, že odstraňuje tenké čáry a ostré rohy. Proto je vhodné předem vědět, co od filtru potřebujeme a matici upravit podle toho. Díky tomu, že mediánový filtr nezpůsobuje takové rozmazání obrazu jako filtry jiné, patří mediánový filtr ve strojovém vidění k nejpoužívanějším.



Obrázek 2.5: Porovnání originálního obrázku (vlevo) a filtrovaného (vpravo)

Obrázek 2.5 je praktický příklad využití 3x3 mediánového filtru, je viditelné odstranění šumu, při zachování jasných hran.

2.5 Segmentace

Segmentace obrazu je jedním z nejdůležitějších přípravných kroků k analýze obrazu. Jejím hlavním cílem je rozdělit obraz na části, které mají silnou korelaci s předměty nebo oblastmi reálného světa vyskytující se v obrazu. Můžeme provést buď úplnou segmentaci, díky které získáme sadu disjunktních oblastí odpovídající objektům ve vstupním obrazu, nebo částečnou segmentaci, při které oblasti nekorrespondují přímo s objekty v obraze.

Úplná segmentace obrazu R je tvořena konečným množstvím regionů R_1, \dots, R_S ,

$$R = \bigcup_{i=1}^S R_i, \quad R_i \cap R_j = \emptyset, \quad i \neq j.$$

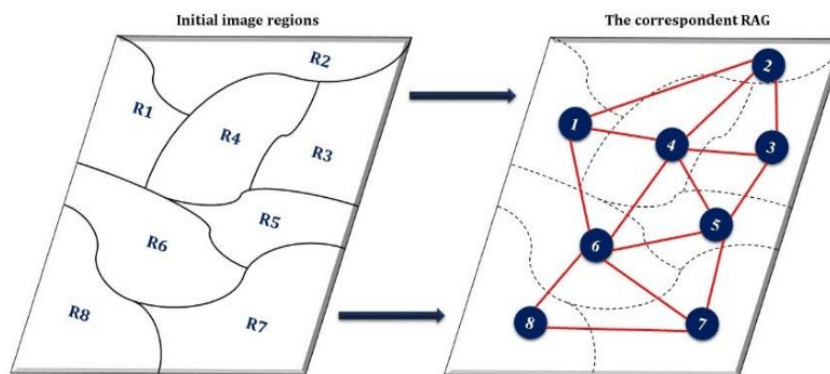
Opět je velmi výhodná apriorní znalost scény. V mém případě se jedná o silný kontrast objektu oproti čistě černému pozadí. Úplnou segmentací se tak dá získat rozdělení objektu

a pozadí.

Pokud chceme získat pouze oblast s čísly, je potřeba využít částečné segmentace pomocí například jasu, textury, nebo množství černé.

Často se provádí nejdříve částečná segmentace, která se využije jako vstup pro zpracování na vyšší úrovni, jelikož úplná segmentace by nebyla realizovatelná. Dvojznačnost obrazových dat spojená s informačním šumem je jedním z hlavních problémů segmentace.

Segmentace se dělí do tří hlavních skupin. Globální znalost obrazu, nebo jeho části, většinou reprezentovaná histogramem té známé části, segmentace založená na nalezení hran a segmentace podle regionů. Jelikož pomocí segmentace podle hran a podle regionů získáme uzavřené regiony, které nám dají jiné výsledky, můžeme využít jejich kombinaci pro získání tzv. "grafu souvislostí regionu", kde uzly znamenají regiony a hrany reprezentují souvislost vztahů podle hran regionů.



Obrázek 2.6: Segmentace obrazu a jeho převedení do grafu souvislostí regionu [10]

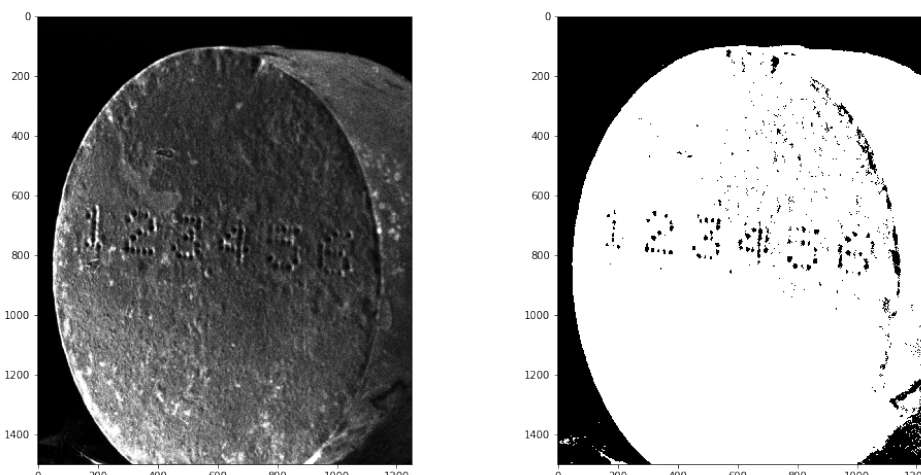
2.5.1 Prahování

Prahování (Thresholding) obrazu v úrovních šedé je nejjednodušší proces segmentace. Mnoho objektů nebo oblastí na obraze je charakterizováno konstantní odrazivostí, nebo absorpcí světla jejich povrchů. Díky tomu lze určit jasovou konstantu tzv. práh. Prahování je výpočetně nenáročné a rychlé – jde o nejstarší segmentaci a je stále široce používáno v jednoduchých aplikacích, jelikož se dá snadno používat i v reálném čase [9].

Prahování je transformace vstupního obrazu f do výstupního binárního obrazu g :

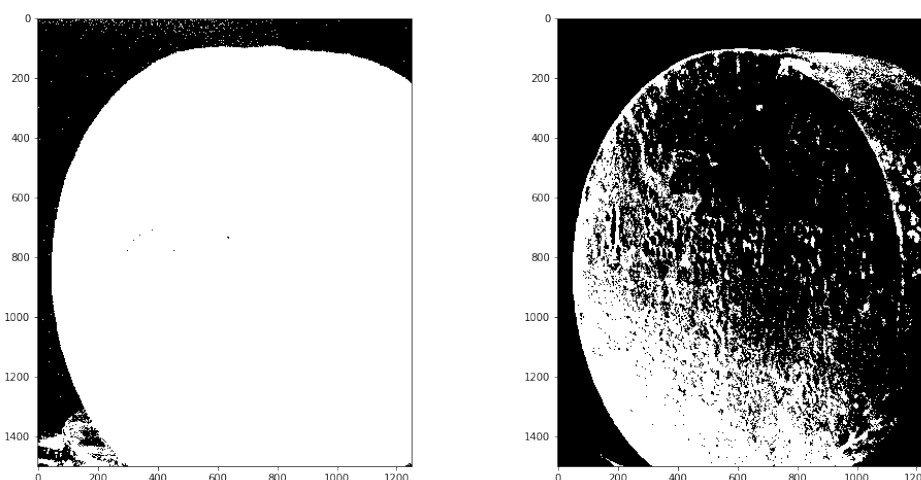
$$g(i, j) = \begin{cases} 1 & \text{když } f(i, j) \geq \text{Práh} \\ 0 & \text{když } f(i, j) < \text{Práh} \end{cases} \quad (2.1)$$

Získáme tedy obraz, kde oblast objektu bude mít hodnotu 1 a pozadí bude mít hodnotu 0 nebo i naopak. Na obrázku 2.7 můžeme vidět porovnání původního obrázku a jeho prahovanou verzi.



Obrázek 2.7: Porovnání originálního obrázku (vlevo) a ideálně prahovaného (vpravo)

Toto je ovšem ideální případ. Může se stát, že práh je nastavený moc vysoko nebo moc nízko, a tím mohou vzniknout obrazy takovéto:



Obrázek 2.8: Porovnání nízkého prahu (vlevo) a vysokého prahu (vpravo)

Z obrázku 2.8 je zřetelné, že správné nastavení prahu je esenciální.

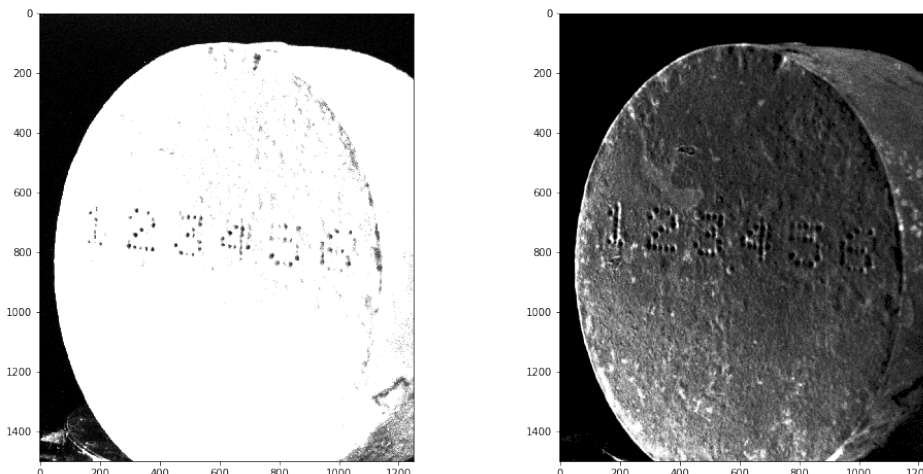
Prahování nemusí být pouze binární. Můžeme například využít těchto rovnic [11]:

$$g(i, j) = \begin{cases} \text{Práh} & \text{když } f(i, j) \geq \text{Práh} \\ f(i, j) & \text{když } f(i, j) < \text{Práh} \end{cases} \quad (2.2)$$

nebo:

$$g(i, j) = \begin{cases} f(i, j) & \text{když } f(i, j) \geq \text{Práh} \\ 0 & \text{když } f(i, j) < \text{Práh} \end{cases} \quad (2.3)$$

Čímž získáme takový obraz:



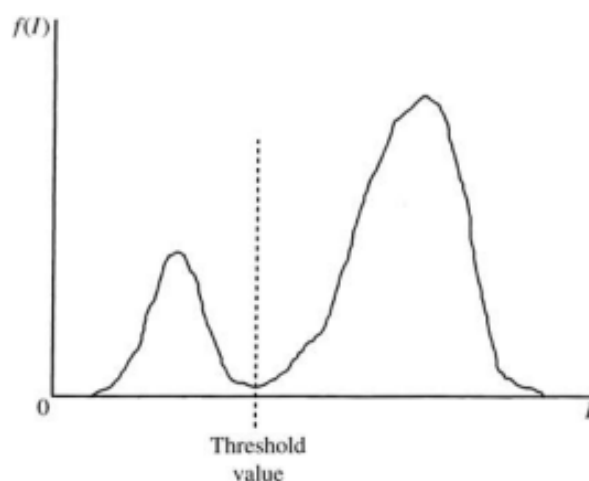
Obrázek 2.9: Porovnání rovnice 2.2 (vlevo) a rovnice 2.3 (vpravo)

Rovnice 2.3 je výhodná, když máme například moc úrovní černé nebo bílé, které nám omezují segmentaci. Sjednotí se všechny hodnoty nad/pod prahem.

2.5.2 Základní výběr prahu

Když je objekt správně osvětlen a hledáme objekt, který má jasnou siluetu oproti kontrastnímu pozadí, je prahování možné provést velmi jednoduše na jedné úrovni intenzity jasu.

Jedna z nejčastěji používaných technik využívá histogramu. Na obrázku 2.10 je vidět histogram se zřejmým minimem, které je využito jako hodnota prahu. Předpokládá se, že vrchol vlevo od prahu je objekt a vrchol vpravo od prahu je světlé pozadí.



Obrázek 2.10: Jasně rozdělitelný histogram [7]

Jelikož ne všechny obrazy mají takto jasný histogram, potýká se tato metoda s několika problémy [7].

- Minimum je velmi široké a není tedy jasné, kam přesně patří práh
- V histogramu se vyskytuje několik minim
- Šum v okolí minima zabraňuje přesné volbě prahu
- Pozadí nebo objekt mají mnohem větší vrchol a tím se ovlivní pozice prahu
- Histogram je multimodální (má několik vrcholů), je těžké určit relevantní práh

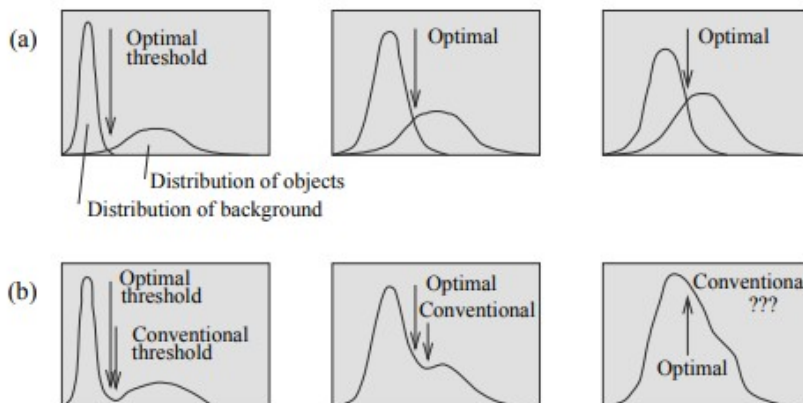
Poslední zmíněný problém je velmi náročný, jelikož je takřka nemožné vědět, jak moc důležité je lokální maximum. Obraz, který je z půlky černý a z půlky bílý, má stejný histogram jako obraz s náhodně rozmístěnými pixely černé a bílé v poměru 50/50.

Obecnější přístup využívá místo globálního histogramu lokální přístup s cílem získat histogram s lepším poměrem maxim a minim. Jedna možnost je přidat histogramu váhovou funkci za účelem omezení důležitosti pixelů v oblastech s vysokým gradientem. To znamená, že histogram se bude skládat z hodnot pozadí a objektu a s hodnotami na přechodu mezi nimi se nebude počítat. Tím získáme hlubší minimum a jednodušší určení prahu.

Opačný přístup je také možný. Využije se pouze pixelů s vysokým gradientem pro vytvoření histogramu. Tím je získán histogram s jedním vrcholem, který popisuje práh mezi objektem a pozadím [9].

2.5.3 Optimální prahování

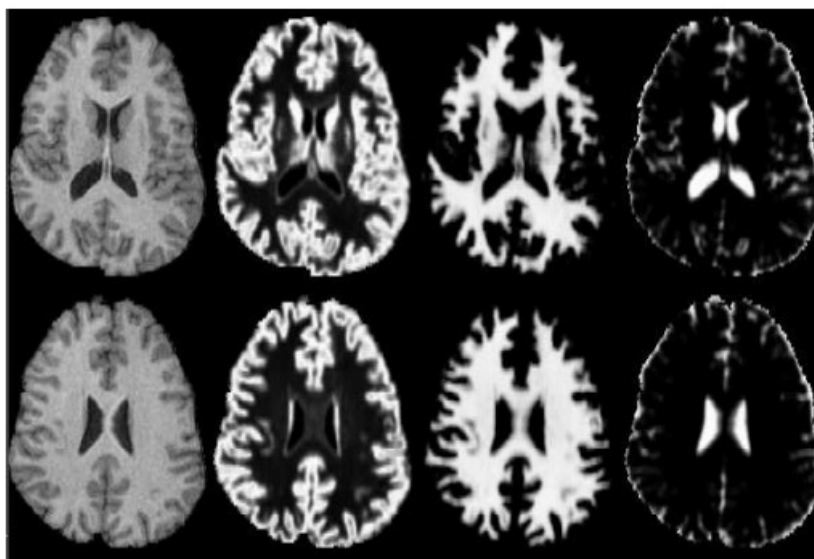
Další přístup k prahování je optimální prahování. Histogram obrazu chceme nahradit pomocí váženého součtu dvou nebo více hustot pravděpodobnosti s normálním rozdělením [9].



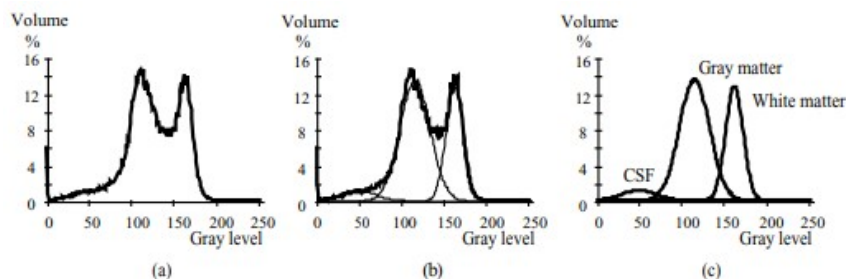
Obrázek 2.11: Histogramy aproximovány pomocí normálního rozdělení (a) a jejich skutečné histogramy se znázorněným běžným prahem [9].

Z obrázku 2.11 můžeme vidět využitý tento způsob prahování. Problém této metody je, že je náročné aproximovat parametry normálního rozdělení spojené s nejistotou, jelikož je obtížné zjistit, zda-li je toto rozdělení stále normální. Na tomto způsobu prahování je založený Otsuho algoritmus, je to jeden z nejčastěji využívaných algoritmů pro automatické prahování.

Příkladem praktického využití optimálního prahování jsou snímky magnetické rezonance (MR).



Obrázek 2.12: Snímky MR. V levém sloupci jsou původní snímky, další jsou snímky segmentované. Vlevo: Částečně objemová mapa šedé hmoty. Uprostřed: Částečně objemová mapa bílé hmoty. Vpravo: Částečně objemová mapa mozkomíšního moku [9].



Obrázek 2.13: Histogram snímku MR. (a) Původní histogram. (b) Gaussova křivka přizpůsobená původnímu histogramu. (c) Gaussovské rozdělení odpovídající zleva: mozkomíšní mok, šedá hmota, bílá hmota [9].

2.6 Morfologické operace

Zatímco dříve zmíněné operace vycházely z konvoluce, morfologické operace vychází z více objektově založené filozofie, která v mnoha ohledech nahrazuje lineární algebraický systém konvoluce. Zvládá mnoho operací, jako předzpracování obrazu, nebo segmentaci pomocí tvaru objektu a jeho kvantifikaci rychleji a lépe než běžné přístupy. Morfologie je často využívána v případech mikroskopických obrazů, kontroly v průmyslu, optického rozpoznávání znaků a analýzy dokumentů. [9]

2.6.1 Dilatace a eroze

Dilatace a eroze vychází z morfologických operací. Dilatace rozšiřuje objekty do pozadí a dokáže eliminovat "salt" šum z objektu. Na druhou stranu eroze zmenšuje objekt a odstraňuje "pepper" šum z objektu. Dilatace a eroze jsou velmi úzce spjaté, je ovšem velmi důležité si uvědomit, že platí:

$$\text{eroze}(\text{dilatace}(S)) \neq S,$$

to znamená, že i když jsou tyto operace praktickým opakem, není možné obnovit už ztracené detaily obrazu. Odstranění "salt" šumu z obrazu pomocí dilatace, znamená, že při erozi tyto vady již neexistují a eroze je nevrátí. Morfologické operace se dají provádět i na obraze v odstínech šedi, zde ale budou zmíněny pouze operace na binárním obraze [7].

Zobecněná morfologická dilatace

Opět za pomoci 3x3 masky, kde je předpokládána prostřední hodnota jako počátek, je možné provést v obraze posun.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Zkombinováním masek (levá zachovává původní obraz, pravá posouvá obraz o jeden pixel doleva), je získáno horizontální rozšíření objektů. Pomocí jedniček v rozdílných pozicích jsou získány jiné druhy rozšiřování. Pokud se maska skládá jen z jedniček, jedná se o izotropické rozšíření. Pro všechny druhy platí, že počátek je vždy 1, a je tedy zachován vždy původní pixel. Masky samozřejmě může nabývat i jiných rozměrů než 3x3, ovšem musí stále mít lichý počet sloupců a řádků. [9]

Zobecněná morfologická eroze

Na první pohled je eroze velmi podobná dilataci, ale eroze se provádí pomocí opačných posunů. Matice jsou identické s dilatací. Obraz se ovšem posouvá opačným směrem a provádí se průmět operací pro získání nového obrazu. Využívá se pro zjednodušení struktury objektu a jeho částí. Objekty s velikostí jedna zmizí a tím je možné získat více méně komplikovaných objektů.

Při využití masky z minulé podkapitoly by bylo tedy odebráno z levé strany objektů. Obraz je posunut doleva a je proveden booleovský operátor AND s původním obrazem.



Obrázek 2.14: Ukázka izotropické dilatace (a) a eroze (b) [9]

Základní morfologické operace lze využít pro nalezení obrysu objektu velmi efektivně a rychle.



Obrázek 2.15: Nalezení obrysu pomocí eroze [9]

Na obrázku 2.15 je vidět výsledek odečtení erodovaného obrázku od původního. [9]

Vlastnosti operace dilatace a eroze

Uvažujme obraz A a masky B , C a zároveň dilataci a erozi formálně zapsané jako $A \oplus B$ a $A \ominus B$.

Operátory dilatace a eroze mají několik důležitých a užitečných vlastností. Zaprvé, postupné dilatace jsou asociativní:

$$(A \oplus B) \oplus C = A \oplus (B \oplus C),$$

zatímco postupné eroze nejsou. Vztah pro postupné eroze je:

$$(A \ominus B) \ominus C = A \ominus (B \oplus C).$$

Dále nezáleží na pořadí, ve kterém jsou dilatace a eroze provedeny.

$$(A \oplus B) \oplus C = (A \oplus C) \oplus B,$$

$$(A \ominus B) \ominus C = (A \ominus C) \ominus B,$$

Navíc, při připojení matematické logiky k morfologickým operacím vzniká několik dalších vztahů. Tyto operace jsou velmi náchylné na chyby, jelikož jsou to jen specifické operace, pro které platí:

$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C)$$

$$A \ominus (B \cup C) = (A \ominus B) \cup (A \ominus C)$$

$$(A \cap B) \ominus C = (A \ominus C) \cup (B \ominus C)$$

Tyto rovnice ukazují, že velké a komplexní dilatace/eroze, lze implementovat efektivněji jako dvě postupné menší eroze/dilatace.

Můžeme toho například využít při provedení izometrické dilatace:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Tím, že je provedena nejdříve horizontální dilatace a poté vertikální dilatace, je získána dilatace izometrická. [9]

Otevření a uzavření

Dilatace a eroze jsou pouze základní operace, ze kterých lze odvodit mnoho dalších. V minulé podkapitole je zmínka o tom, že erozí nelze anulovat dilataci a naopak. Zavádějí se dva nové operátory, které popisují jejich vzájemnou anulaci. Jelikož jeden operátor typicky uzavírá mezery mezi objekty, je označován uzavření. Druhý otevírá mezery mezi objekty, a proto se nazývá otevření. Uzavření (\bullet) a otevření (\circ) jsou formálně definována:

$$A \bullet B = (A \oplus B) \ominus B$$

$$A \circ B = (A \ominus B) \oplus B$$

Uzavření eliminuje "salt" šum, praskliny a malé díry, zatímco otevření eliminuje "pepper" šum a malé výčnělky. Oboje tyto metody tedy odstraňují chyby v obraze, které jsou menší, než zvolená maska, a zároveň zachovávají celkový tvar objektu.



Obrázek 2.16: Otevření (originál vlevo) [9]



Obrázek 2.17: Uzavření (originál vlevo) [9]

Pomocí odečtení nového obrazu z původního jsou získány chyby odstraněné pomocí otevření nebo uzavření, což je velmi užitečné v defektoskopii. Mezi praktické aplikace těchto dvou operátorů patří lokalizace pájecích můstků a trhlin ve vodičích desek plošných spojů. [9]

2.7 Vyhledávání v obraze

Vyhledávání v obraze je základní přístup k segmentaci, pomocí kterého lze nalézt předem známé objekty nebo vzory v obraze. Tato metoda je velmi úzce spjatá s umělou inteligencí a strojovým viděním. Celkové vyhledávání je velmi obecné. Pro interpretaci textu lze využít vyhledávání pomocí šablony reprezentující jednotlivá písmena. Pokud ovšem text má neznámý font a velikost, je třeba hledání vzorů reprezentujících jednotlivá písmena. Mezi komplexnější případy patří například rozpoznávání obličeje. Vzor obličeje je potřeba nalézt v obraze 3D scény, kde je velké množství neznámých, jako velikost, póza, vousy, brýle, atd. Dá se vyhledávat například i chování. Z videa ulice jde zjistit, jestli chodec přechází ulici, pokračuje po chodníku nebo jen stojí.

2.7.1 Vyhledávání šablony v obraze

Tento přístup patří k nejjednodušším k nalezení objektu v obraze. Hledá se v obraze oblast, která odpovídá pixelovým hodnotám šablony. Objekt v obraze musí být naprosto identický, žádná změna ve velikosti, nebo natočení.

Při uvažování šablony T o velikosti $r_T \times c_T$ a obraz I , který je udržován v offsetu $x = (x_a, x_b)$ platí, že pokud bude šablona identická s oblastí obrazu je získáno:

$$E(x) = \sum_{i=1}^{r_T} \sum_{j=1}^{c_T} (T_{i,j} - I_{x_a+i, x_b+j})^2 = 0$$

,

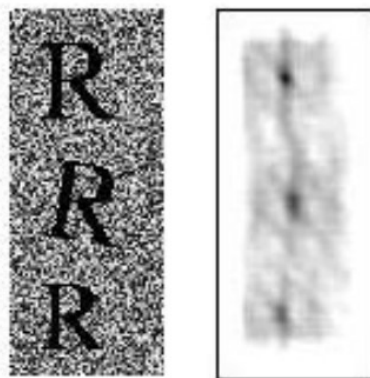
kde E popisuje chybu. Dalšími úpravami této rovnice lze zjistit, že hledání šablony lze provést pomocí maximalizace korelace:

$$Corr_T(x) = \sum_{i=1}^{r_T} \sum_{j=1}^{c_T} (T_{i,j} I_{x_a+i, x_b+j})$$

.

Důležité je zmínit, že sumace je citlivá na rozsah intenzity a velikost oblasti T . Je vhodné využít prostorové nebo intenzitní škálování.

V praktickém využití je to velmi omezené. I malé pootočení nebo změna velikosti způsobují rychlý nárůst chyby E . [9]



Obrázek 2.18: Šablona písmena R je vyhledávána v levém obraze. Vpravo je graficky znázorněna hodnota E (černá je pro E nulové, tedy nejlepší schodu). Je zřejmé, že při natočení, nebo změně velikosti je E větší [9].

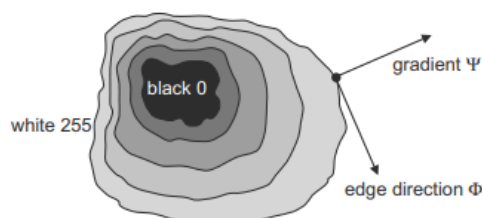
2.7.2 Detekce hran

Detektory hran jsou souborem velmi důležitých metod lokálního předzpracování obrazu sloužící k vyhledávání změn intenzity jasu. Hrany jsou pixely, kde se intenzita jasu náhle

změní.

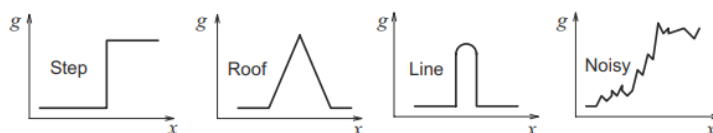
Neurologický a psychofyzický výzkum naznačuje, že části obrazu, kde se intenzita jasu náhle mění, jsou velmi důležité pro jeho vnímání. Hrany jsou téměř invariantní na změnu osvětlení a úhlu. Často pro pochopení obrazu stačí pouze pixely s velkou změnou intenzity tzv. "edgels". Tím se dá razantně snížit množství dat obrazu. Zároveň nedochází k výrazné ztrátě informací.

Jak moc jsou jednotlivé pixely hranou, se počítá pomocí funkce v okolí daného pixelu. Jedná se o vektorovou veličinu s velikostí a směrem. Velikost odpovídá gradientu a směr je pootočený o -90° . Gradient udává velikost největšího růstu funkce, v tomto případě tedy přechod z černé $h = (0)$ na bílou $h = (255)$.



Obrázek 2.19: Ukázka gradientu Ψ a směru Φ [9]

Hrany se často využívají pro nalezení hranic oblastí. Pokud oblast má stejný jas, je hranice oblasti v místě se skokovou změnou jasu [9].



Obrázek 2.20: Typické tvary hran [9]

Jednotlivé operátory gradientu se počítají pomocí konvoluce a dají se tedy vyjádřit pomocí konvoluční matice.

Sobelův operátor

Sobelův operátor je často využíván pro jednoduché a rychlé nalezení horizontálních a vertikálních hran. K tomu je využito dvou matic:

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

Pokud uvažujeme $h_1 = y$ a $h_3 = x$, získáme důležitost hrany pomocí

$$\sqrt{x^2 + y^2} \text{ nebo } |x| + |y|$$

a směr pomocí $\arctan(y/x)$ [12], [9].

2.8 Python

Python je programovací jazyk využívaný jak nováčky, tak zkušenými programy pro velmi širokou škálu projektů. Hlavním důvodem, proč je využívaný pro strojové vidění, je integrace strojového učení.

Do tohoto projektu byl vybrán hlavně kvůli jeho jednoduchosti, rychlosti prototypování a apriorní znalosti. Rychlost prototypování přichází na úkor optimalizace výkonu. Python stále využívá originálních knihoven psaných v C/C++ a není tedy optimální pro maximální rychlost programu. [12]

2.8.1 Knihovna OpenCV

Open Source Computer Vision Library (otevřený software pro strojové vidění) je knihovna často využívaná pro strojové učení a vidění. Knihovna OpenCV byla vytvořena pro poskytnutí infrastruktury pro aplikace strojového vidění a zrychlení vývoje strojového vnímání v komerčních produktech. Tato knihovna má přes 2500 optimalizovaných algoritmů, které se dají zdarma využít pro osobní i komerční využití. Všechny upravované snímky dále v této práci byly získány za pomoci Pythonu a OpenCV. [11]

Kapitola 3

Praktická část

Předmětem praktické části diplomové práce je návrh a vývoj stanoviště automatického kamerové rozpoznání unikátního identifikátoru hutního polotovaru - sochoru. Projekt sestává z návrhu optického pracoviště a z návrhu a vývoje algoritmů pro automatické rozpoznávání unikátních identifikátorů na čelech železných sochorů po dalším tepelném zpracování. Identifikace se provádí za studena. Unikátní identifikátory jsou textové řetězce vyražené metodou "dot matrix" (důlkování podle definovaných vzorů). V současnosti jsou sochory čísla ze sochorů čtené obsluhou. Tento způsob je nejen nákladnější, ale zároveň i víc náchylný k chybám. Proto bylo rozhodnuto tuto část výrobního procesu automatizovat. Mezi další výhody automatického snímání patří i možnost implementace do dalších databázových systémů a jeho rychlost. Aby byl systém zpětně kompatibilní a zároveň byla usnadněna práce obsluhy při dalším zpracování sochorů, bylo rozhodnuto, že vyražená čísla na sochory musí být stále čitelná obsluhou. Vyrážený jsou tedy arabské číslice ve formě "dot matrix", které jsou čitelné jak obsluhou, tak automaticky programem.

3.1 Stanoviště

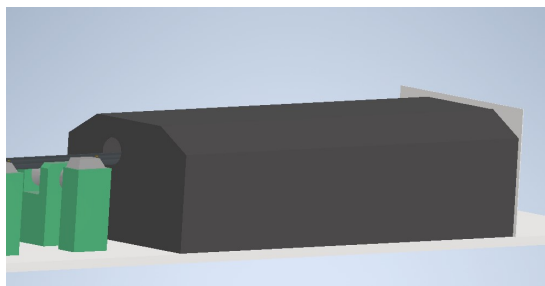
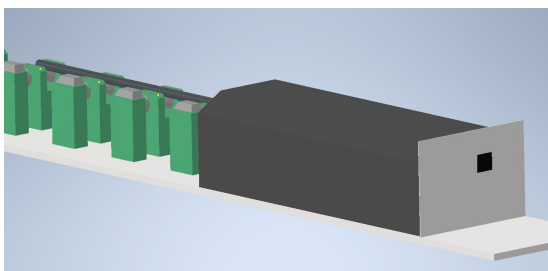
V rámci toho projektu se věnuji návrhu celého stanoviště. To znamená návrh zakrytí části stanoviště kvůli lepšímu osvětlení a výběr HW (kamery, osvětlení, Osobní počítač (PC)). V rámci železárně se jedná o část, kde se ze sochoru ofukují okuje, která se posune před rozpoznávání.



Obrázek 3.1: Fotky stanoviště v železárně. Pohled odkud sochor vyjíždí (vlevo) a kam zajíždí (vpravo)

Na obrázku 3.1 vlevo v popředí je vidět, kde se sochor ofukuje. Místo této části bude umístěno osvětlení a bude zde zastavený sochor pro focení. Na obrázku vpravo je bílá zeď, u které bude umístěna kamera. Část této oblasti až po bílou zeď bude zakrytá plechem, aby osvětlení železářny neinterferovalo s osvětlením sochoru.

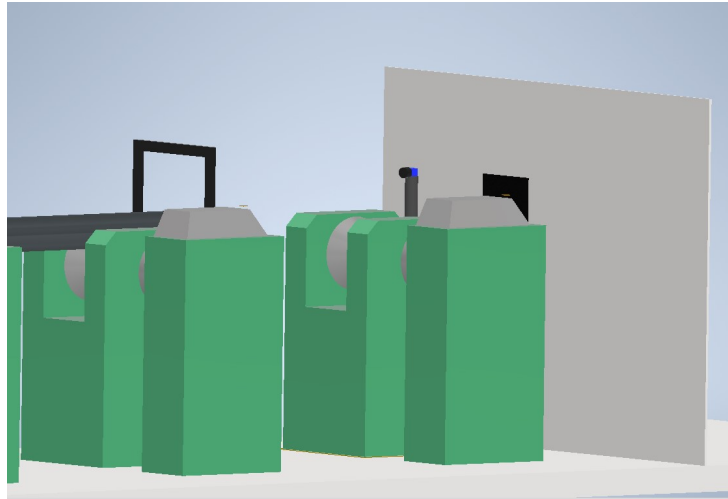
Finálně by stanoviště mělo vypadat takto:



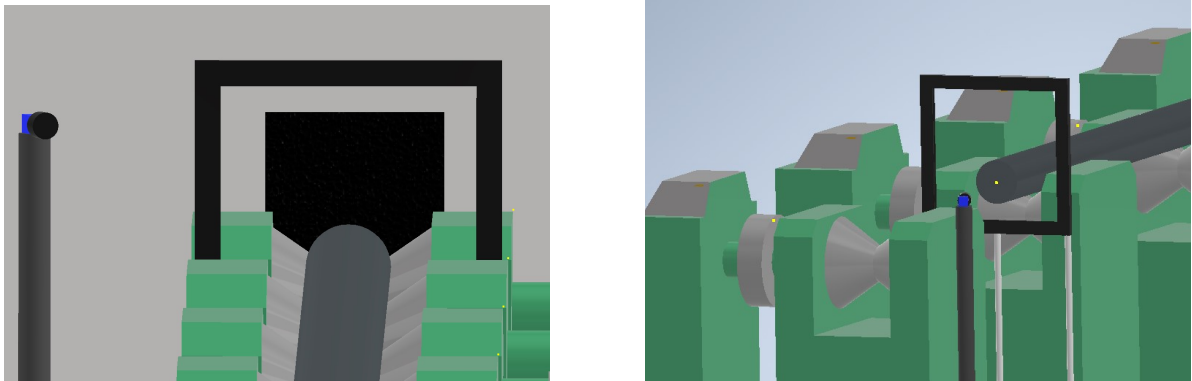
Obrázek 3.2: Model stanoviště zakrytý plechovým krytem

Na vstupu je pouze malý otvor, který je uzavřený pryžovým krytem podobně jako na fotce 3.1 vpravo.

Bez krytu pak stanoviště bude vypadat takto:



Obrázek 3.3: Stanoviště v podobném úhlu jako obrázek 3.1 vpravo

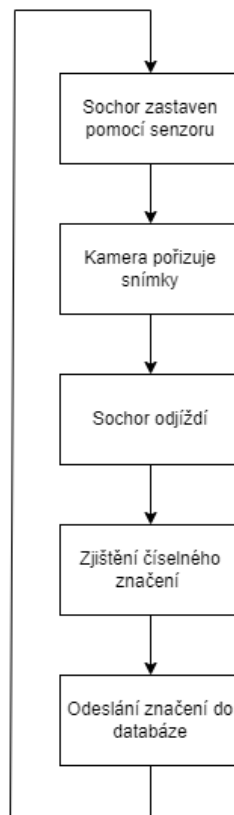


Obrázek 3.4: Pohledy směrem od vývalku (vlevo) a směrem od kamery (vpravo)

3.1.1 Vývalky

Na toto stanoviště přijíždějí vývalky o průměru od 12 cm do 35 cm. Jejich profil může být kruhový nebo čtvercový. Vývalky mohou mít jakoukoliv orientaci, vývalky se čtvercovým profilem mají vždy jednu z hran dole. Vyražené číslo "dot matrix" je vždy stejné velikosti. Vývalky přijíždějí na stanoviště za pokojové teploty.

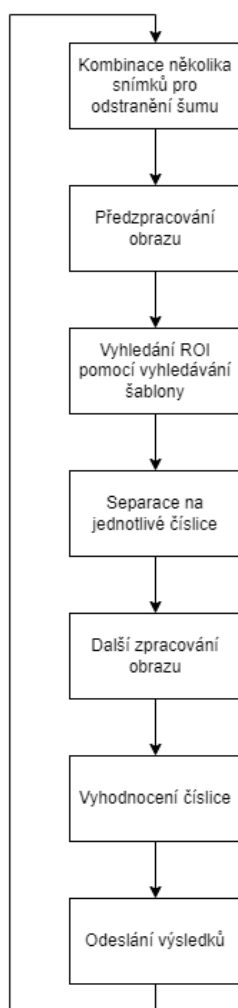
3.2 Struktura úlohy



Obrázek 3.5: Diagram posloupnosti stanoviště

Stanoviště se bude skládat z PLC se senzorem, které bude komunikovat s PC a informovat ho o přítomnosti sochoru. PC spustí softwarovou spoušť kamery přes USB. Po pořízení snímků bude PC informovat PLC, že sochor může opustit stanoviště. Poté dojde ke zpracování snímků a odeslání číselného hodnoty na kontrolu do databáze.

3.2.1 Struktura programu

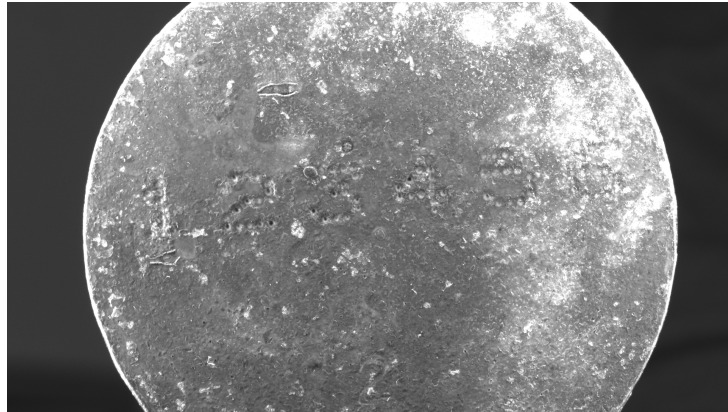


Obrázek 3.6: Diagram struktury programu

3.3 Úvodní pokusy

Jelikož vývalek, když je osvětlen zepředu vypadá takto:

3.3.1 Struktura programu



Obrázek 3.7: Vzorek s čelním osvětlením

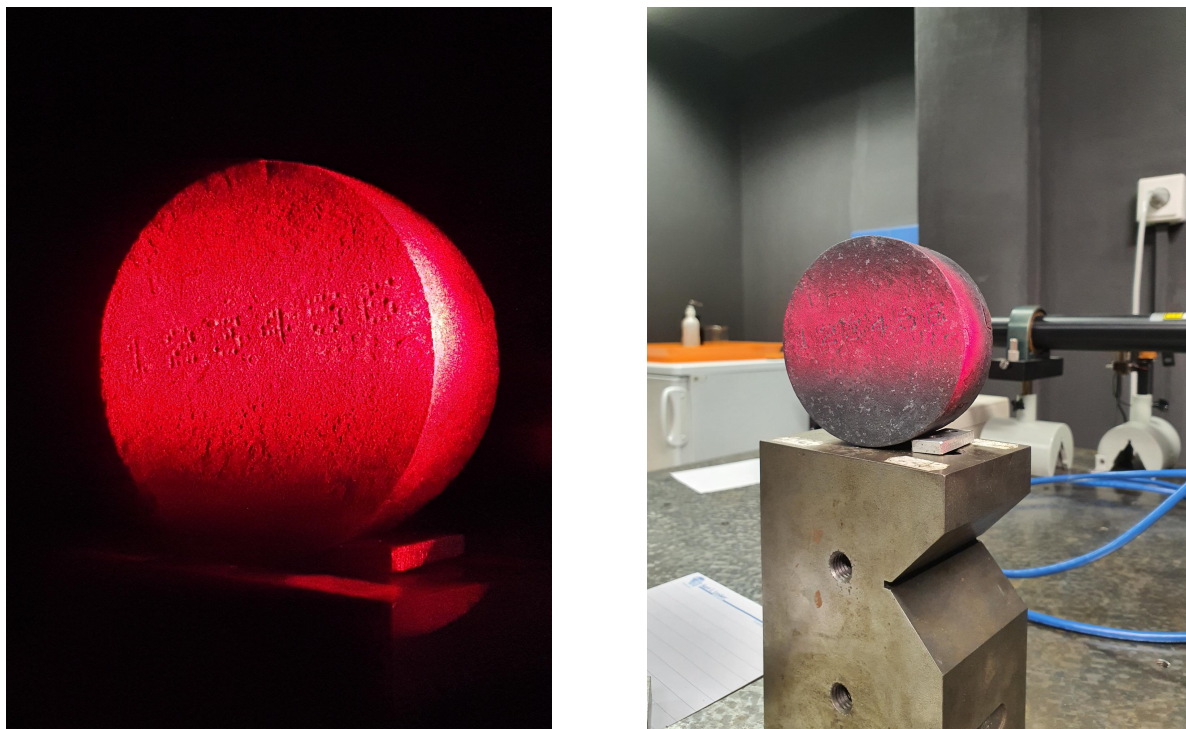
bylo hned jasné, že bude potřeba speciální přípravy, aby značení bylo čitelné.

3.3.2 Osvětlení

U strojového vidění je osvětlení nesmírně důležité. Dobré osvětlení vám může velmi usnadnit práci, zatímco špatné osvětlení může úlohu znemožnit.

Z fotky 3.7 bylo jasné, že čelní osvětlení nebude vhodné použít. Ve svém testování jsem tedy přistoupil k osvětlení ze strany.

Nejdříve jsem zkoušel osvětlení pomocí laseru:



Obrázek 3.8: Úvodní pokusy osvětlení pomocí laseru ve tmě (vlevo) za silného okolního osvětlení (vpravo)

Výsledky vypadaly velmi slibně, ovšem kvůli velké variaci ve velikosti vývalků by bylo potřeba pokrýt moc velké plochy. Zároveň jsem kvůli lepšímu rozlišení zvolil černobílou kameru, takže červená barva laseru by byla nevyužitá. Bylo tedy odstoupeno od tohoto návrhu.

Další možností bylo infračervené osvětlení. Největší výhodou tohoto osvětlení by bylo, že by nebylo třeba zakrývání stanoviště. Bohužel infračervené kamery většinou provádějí vlastní osvětlení, které by dopadalo na vývalek čelně, takže by byl nečitelný. Jinak by se musela vyrábět speciální konstrukce na infračervené osvětlení, která by byla dražší než využití běžného osvětlení, tak také bylo od tohoto nápadu odstoupeno.

Zvolil jsem tedy osvětlení pomocí LED. Původní návrh byl mít osvětlení z několika stran a postupně složit fotky s osvětlením z různých stran. Při původních pokusech jsem ovšem zjistil, že při osvětlení ze všech stran najednou jsou výsledky ještě lepší. Eliminují se všechny nečistoty povrchu, které vystupují ven z vývalku, jelikož díky osvětlení ze všech stran nevrhají stín, a tím zůstanou ve stínu pouze důlky a nečistoty vstupující dovnitř.



Obrázek 3.9: Úvodní pokus s osvětlením ze všech stran

Toto vedlo k vytvoření prototypu stanoviště.



Obrázek 3.10: Prototyp stanoviště

Tento prototyp prošel několik iteracemi. Nejdříve se jednalo jen o rám držící LED. Zjistil jsem, že bez stínidla mezi kamerou a LED vznikají na obraze artefakty a zároveň snižují rozsah jasu. Proto byly přidány stínidla (na obrázku 3.10 černé pruhy papíru). Jedná se tedy o U profil, kde jeho strany musí být dostatečně dlouhé z obou stran, aby nedošlo ke svícení LED do kamery a zároveň nesvítily přímo na sochor a zhoršily tím rozeznatelnost důlků.

Dále byl černý kryt rozšířen přes celý rám, aby se ve snímcích nevyskytovaly žádné objekty s vysokým jasem.

Jelikož LED produkují příliš mnoho ostrého světla do všech stran, které se pak dále odráží i ze stínítek, byl přes LED umístěn difuzor, díky kterému se z LED stal zdroj více plošného světla.

Vývalek byl pak umístěný na stojánek, který poskytoval možnost posunu vývalku skrz rám a bylo zjištěno, že vývalek je čitelný na přibližném rozsahu 1,5 cm hloubky.



Obrázek 3.11: Jedna z iterací stanoviště

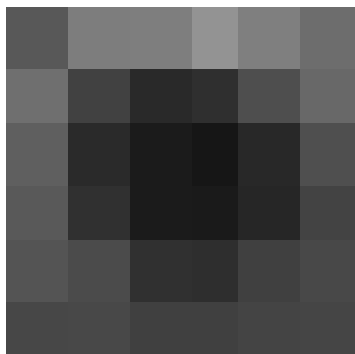
Z obrázku 3.11 je lépe viditelné, jak je celé stanoviště sestavené. Jedná se o jednu z dřívějších iterací, kde ještě není zakrytý celý rám a je využit špatný difuzor. Dále je také focení prováděno barevnou kamerou, která se nakonec nepoužila. Fotky novějších iterací v použití nejsou moc užitečné, jelikož je vše černé, kromě vývalku. Fotky prezentované v dalších kapitolách jsou již pořízené na poslední iteraci stanoviště.

3.3.3 Kamera

Nejdříve jsem pro focení využíval barevné 5 MP kamery. Tato kamera byla nevhodná, jelikož k šumu jasu se přidal šum chromatický. Vznikal tedy nedostatečně kvalitní obrázek. Dále také barevné kanály nebylo možné využít pro vyhodnocení obrázku. Přistoupil jsem tedy k monochromatické kameře s rozlišením 6,3 MP.

Mezi další možnosti patřila infračervená kamera. Jak už je v předchozí kapitole zmíněno, bylo by nutné stejně komplexního řešení osvětlení a zároveň jsou infračervené kamery dražší při menším rozlišení.

S rozlišením 6,3 MP při pokrytí oblasti 35x35 cm vychází, že každý důlek, který má průměr přibližně 0.8 mm, má přibližně 4 px průměr.



Obrázek 3.12: Jeden důlek

Fotky jsou ukládány ve formátu .TIFF, aby docházelo k co nejmenší kompresi.

Kamera má možnosti snímat i při menších rozlišeních, a díky tomu bylo potvrzeno, že menší rozlišení je neslučitelné se získáním kvalitních výsledků.

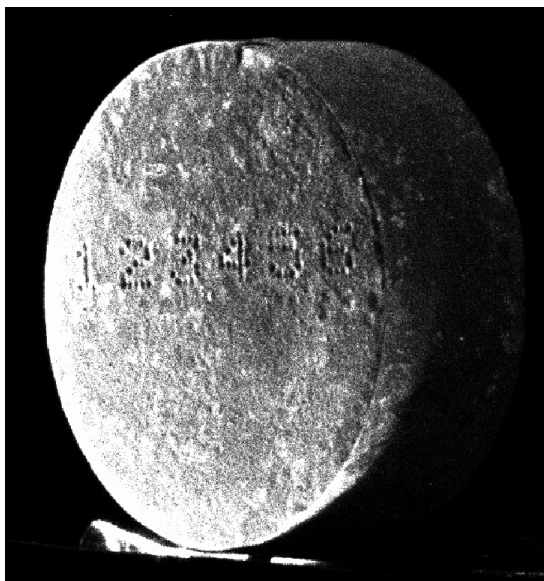
Ze zadání je vyžadováno, aby kamera komunikovala s PC přes USB 3.0, a tedy třeba zvažovat jiné způsoby komunikace.

V laboratoři jsem využíval kameru "DMK 33UX178 USB 3.0 monochrome industrial camera" prodávaná firmou The imaging source.

3.3.4 Úhel kamery

Dalším důležitým rozhodnutím bylo umístění kamery. V ose vývalku je pouze 1.8 m maximální vzdálenosti, zatímco kolmo na osu je prostoru prakticky neomezeně. Bylo tedy potřeba rozhodnout, jestli kameru umístit 1.8 m od čela vývalku a 0.5 metru od osy, nebo kameru umístit blíže k čelu, ale dále od osy.

Mezi hlavní výhody umístění kamery patří efektivnější využití rozlišení kamery. Při pořizování snímků čtvercového tvaru využíváte pouze plné rozlišení pro kratší stranu senzoru, zatímco delší strana má spoustu nevyužitého rozlišení. Při umístění pod velkým úhlem vypadá čtvercový tvar jako obdélník a jde tedy senzor využít plně.



Obrázek 3.13: Snímek ze vzdálenosti 105 cm od čela, 150 cm kolmo od osy

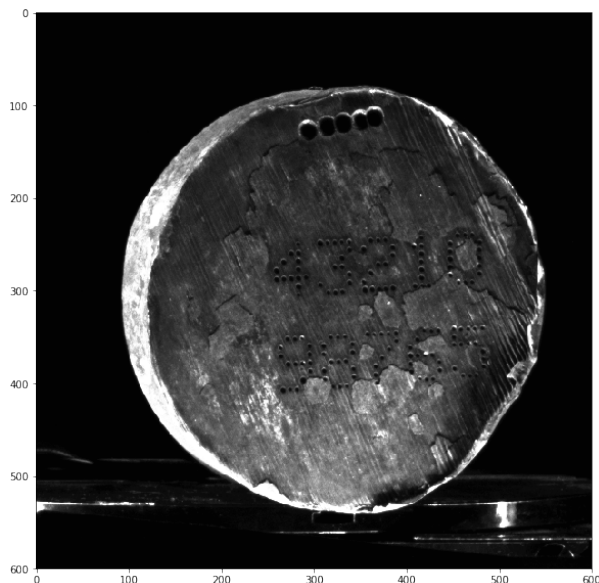
Bylo možné tedy místo symetrického důlku 4x4 získat asymetrický důlek průměrně s rozlišením 4x8.

Větší úhel ovšem způsoboval to, že důlky blíže kamery mají větší rozlišení než důlky vzdálenější. Rozdíl mezi nejlepším a nejhorším případem mohl být až 4 px.

Dalším problémem snímání pod velkým úhlem je hloubka ostrosti. Jelikož značení se může libovolně objevit téměř na celém snímku, musí být celé čelo vývalku ostré. To by znamenalo zvětšit clonové číslo a tím pádem mít větší osvětlení a delší expoziční čas. Jelikož v okolí stanoviště jsou lehké vibrace, mohl by delší expoziční čas znamenat rozmazaný obraz.

Největší problém ovšem představuje osvětlení. Jelikož je kamera pod velkým úhlem, snímá přímo i LED osvětlující vývalek. To znamená, že ve snímku se objevují i extrémně jasná místa, která pak zhoršují kvalitu snímku. Zakrytí LED tak, aby na ně kamera neviděla, by znamenalo i omezení osvětlení vývalku.

Kvůli těmto důvodům jsem se rozhodl kameru umístit do vzdálenosti 1,8 m v ose vývalku a 50 cm kolmo na osu vývalku.



Obrázek 3.14: Snímek ze vzdálenosti 180 cm od čela, 50 cm kolmo od osy

Snímek vývalku není třeba dále transformovat, osvětlení je zakryté a důlky mají pravidelný tvar o průměru 4 px.

3.3.5 Objektiv

Pro pořizování snímků v laboratoři jsem používal "Carl Zeiss Jena DDR Tevidon 1,4/25".



Obrázek 3.15: Objektiv

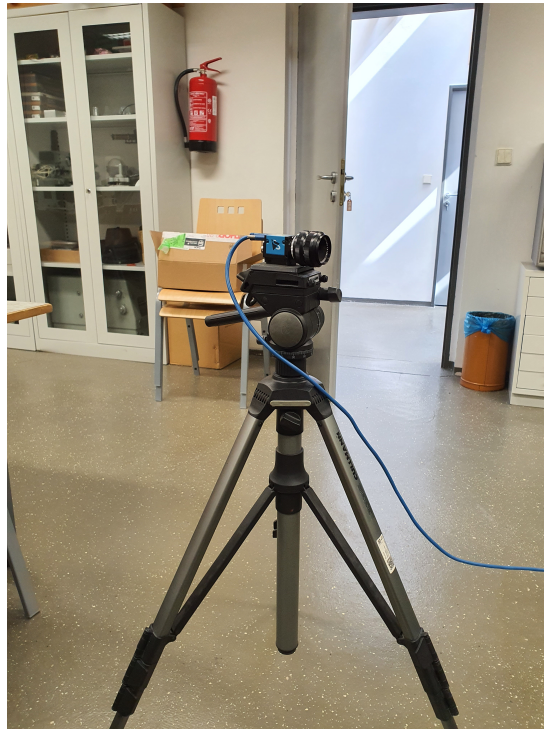
Pro stanoviště je adekvátní náhradou jakýkoliv objektiv s maximální clonou $f/1.4$ a ohniskovou vzdáleností 25 mm. Tento objektiv byl vybrán pomocí apriorní znalosti a

testování v laboratoři.

3.4 Testování v laboratoři

3.4.1 Finální rozložení

Pro testování v laboratoři jsem využil kameru zmíněnou v kapitole 3.3.3 a objektiv zmíněn v kapitole 3.3.5. Tato kamera byla postavena 1.8 m daleko od čela vývalku a 50 cm od osy vývalku na stativ. Pro osvětlení jsem využil prototyp znázorněný na obrázku 3.10. Vývalek byl umístěný na posuvný stoleček. Za osvětlovací prototyp byla umístěná černá látka, aby bylo pozadí fotek čistě černé.



Obrázek 3.16: Kamera na stativu



Obrázek 3.17: Pohled na vývalek od kamery



Obrázek 3.18: Foto celého stanoviště

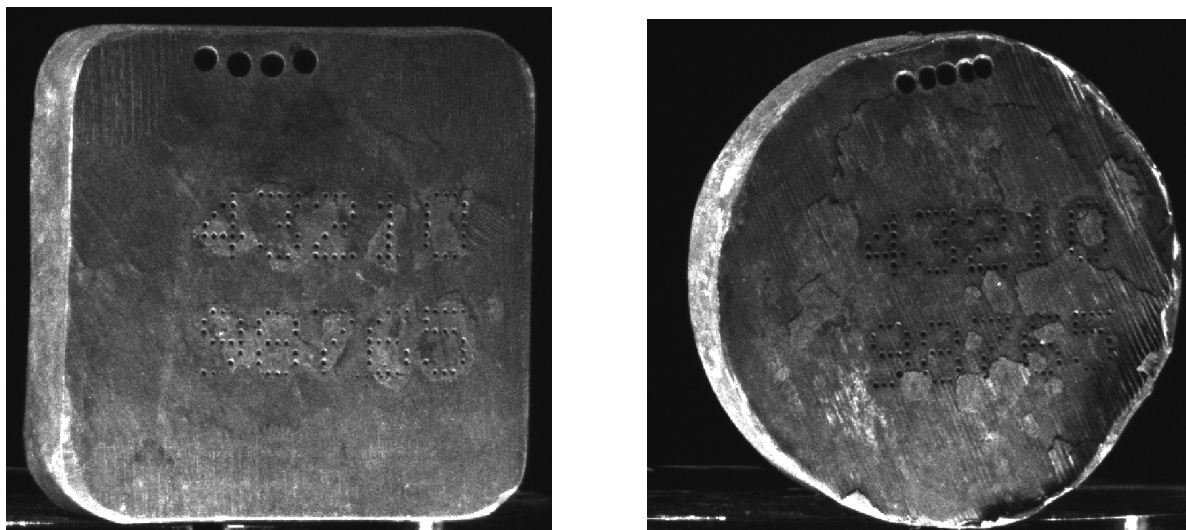
Na obrázcích 3.16 - 3.18 je znázorněn prototyp stanoviště v laboratoři. Laboratoř má začerněná okna, takže při zhasnutí je možné docílit tmy. Snímky představené v další kapitole jsou pořízeny na tomto prototypu.

3.4.2 Úprava fotek

V této sekci se věnuji tomu, jak rozdílné efekty mají vliv na čitelnost značení v obraze. Je důležité si připomenout, že úprava obrazu pro strojové vidění a pro lidské oko jsou velmi rozdílné úlohy, některé obrazy se tedy mohou zdát horší na pohled, i když jsou ve výsledku lepší.

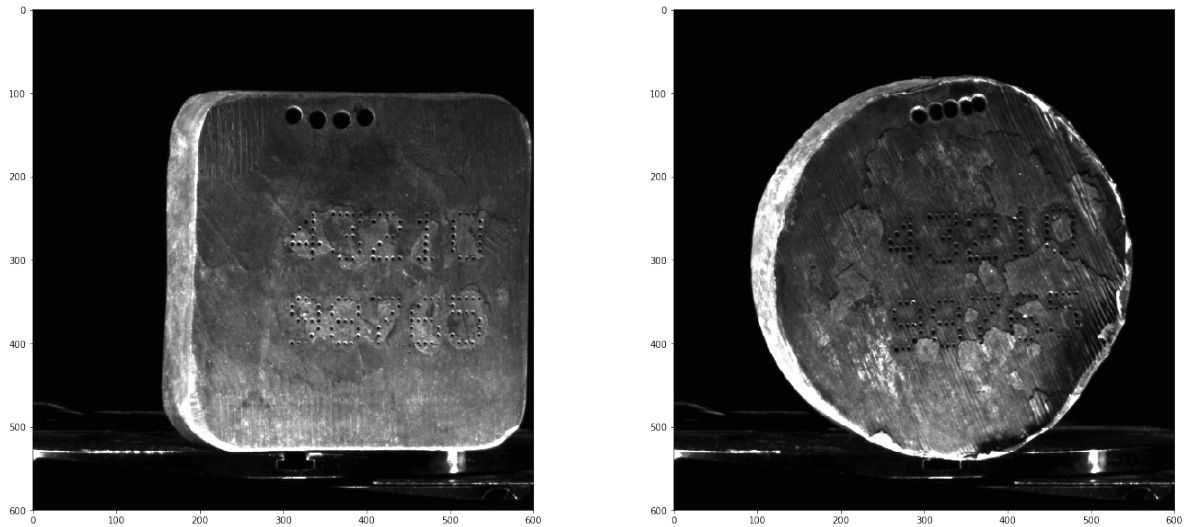
Jsou tu porovnány dva obrázky s rozdílnými povrchy pro bližší představení rozdílů mezi jednotlivými vývalky. Zároveň je z celého obrazu vyříznuta pouze oblast s vývalkem. Vývalky tedy mají správné rozlišení, ale zároveň je vývalek viditelnější. Také je důležité podotknout, že obrázky jsou zde zobrazeny ve formátu .PNG oproti původnímu formátu .TIFF, mají tedy horší kvalitu.

Zde můžeme vidět původní obrázky vývalků označených 4 a 5 (podle děr v horní části). Vývalky jsou rozdílné kvalitou povrchu a zároveň hloubkou důlků.



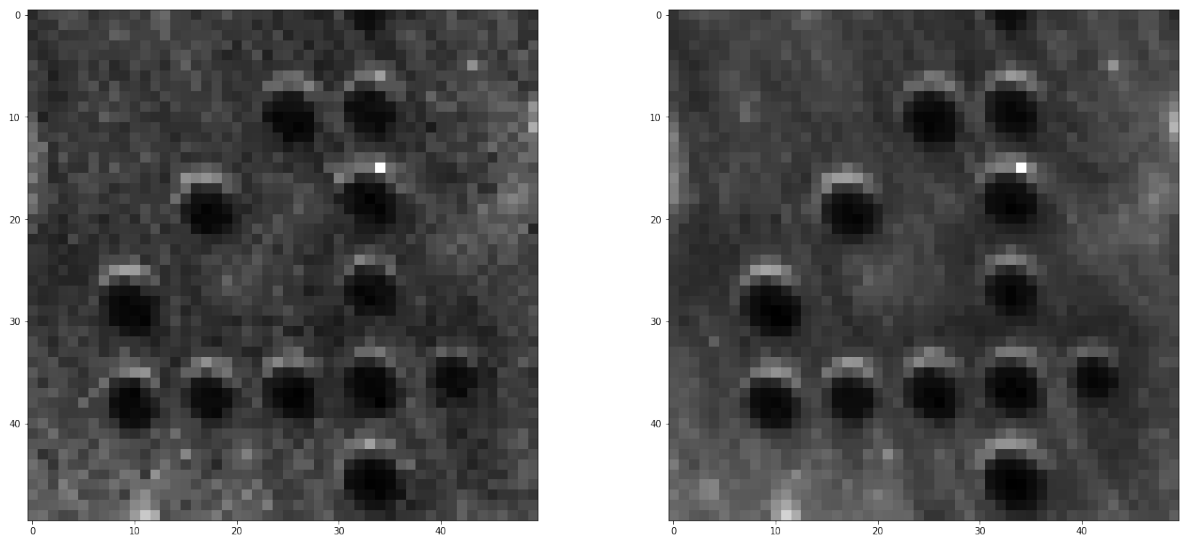
Obrázek 3.19: Původní snímky vývalků 4 (vlevo) a 5 (vpravo)

Takových snímků je pořízeno 10 a poté je vytvořen snímek nový pomocí mediánů jednotlivých pixelů.



Obrázek 3.20: Zprůměrované snímky vývalků 4 (vlevo) a 5 (vpravo)

Pro zvýraznění užitečnosti tohoto kroku jsem vybral menší oblast z obou obrázků pro porovnání.

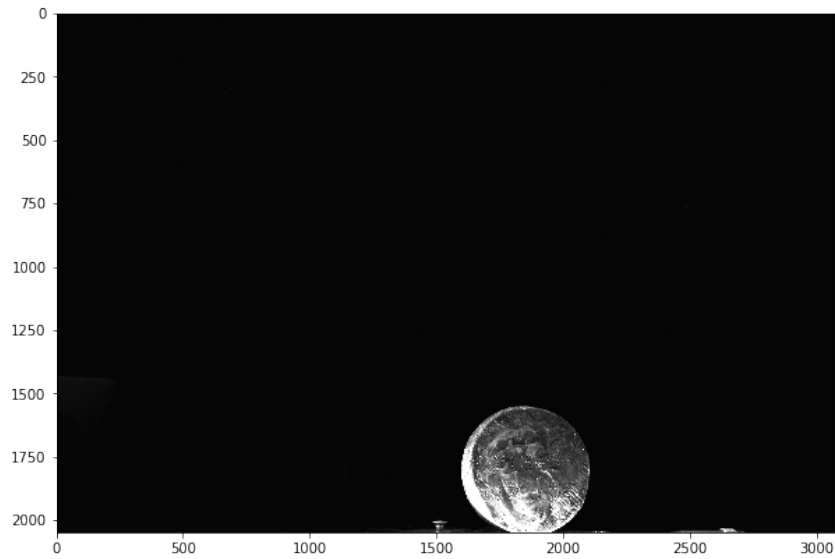


Obrázek 3.21: Porovnání kvality snímků

Dojde tedy k odstranění šumu ze senzoru kamery při zachování stejného rozlišení.

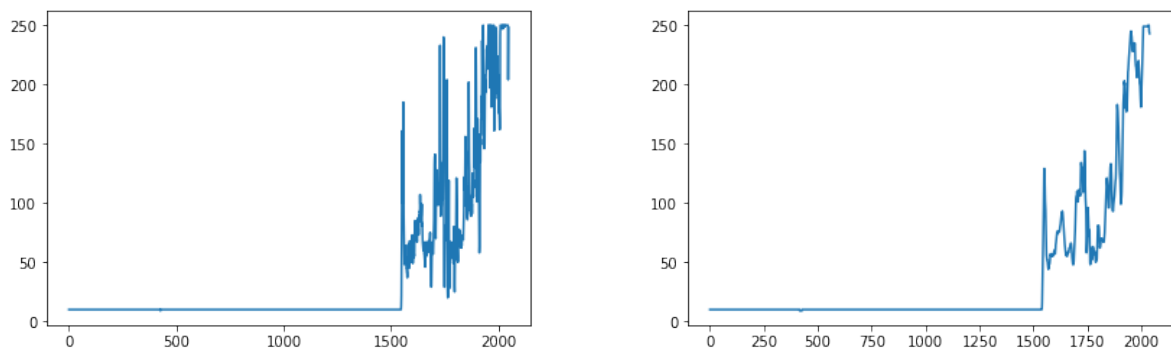
3.4.3 Nalezení vývalku

Jelikož vím, kde se nachází osa vývalku (vývalek se pohybuje po kuželových válcích), je relativně snadné zjistit jeho velikost.



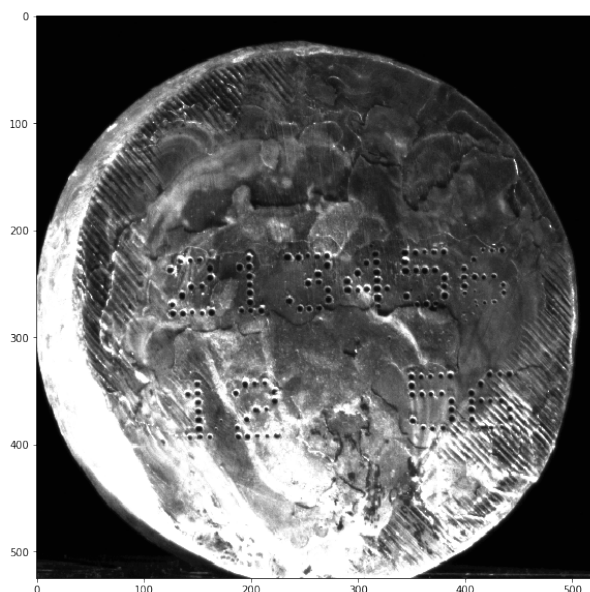
Obrázek 3.22: Původní fotka

Z původního obrázku 3.22 vezmu jas pixelů ve svislé ose vývalku a získám graf, ze kterého je jasně zřejmé, kde se vývalek nachází. Pomocí klouzavého průměru odstraním šum, aby neměl vliv na detekci.



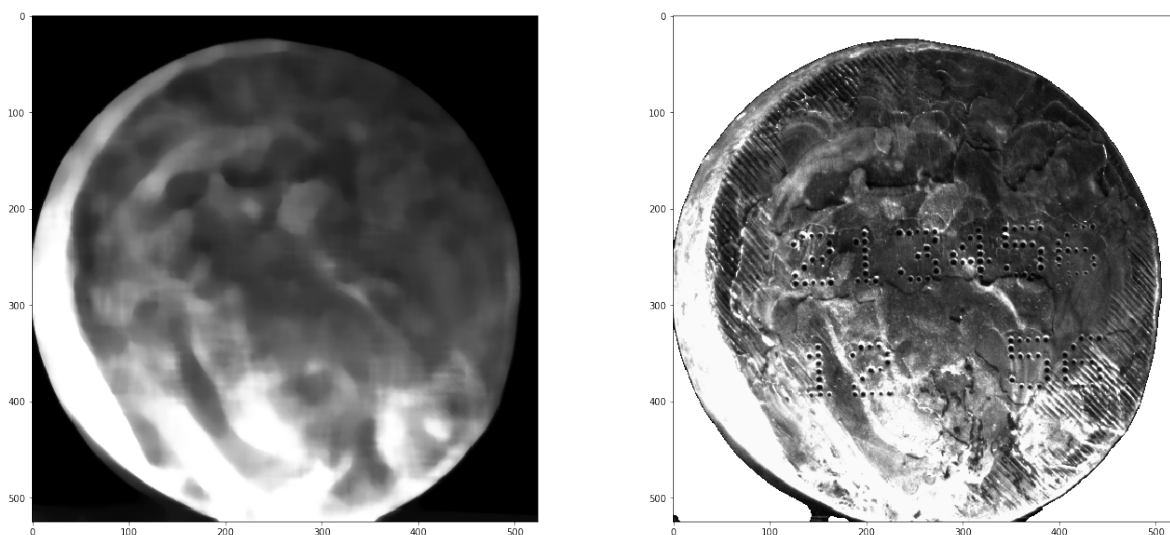
Obrázek 3.23: Původní hodnoty jasu (vlevo) a zprůměrované hodnoty (vpravo)

Z grafu je jasné, že vývalek začíná přibližně na pixelu 1550 ve svislé ose. Je tím získán výřez pouze vývalku a jeho okolí.



Obrázek 3.24: Oříznutý snímek

Dále je na snímek použit velmi silný mediánový filtr, pomocí kterého je možné segmentovat vývalek od pozadí. Mediánový filtr je použit právě proto, že zachovává ostré hrany. Poté pomocí prahování je pozadí vývalku převedeno na bílé.



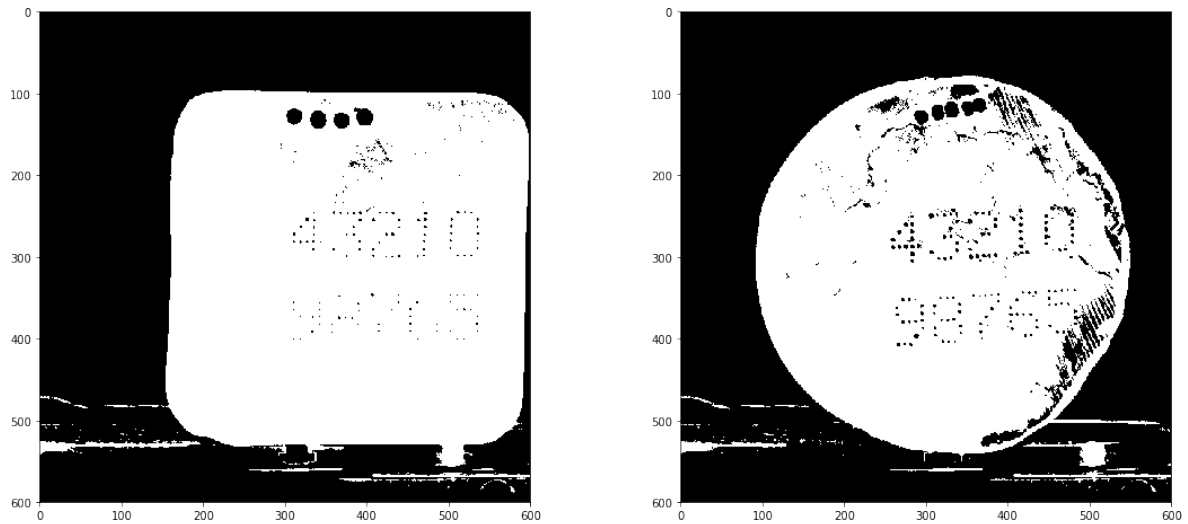
Obrázek 3.25: Filtrováný snímek (vlevo) a prahovaný snímek (vpravo)

3.4.4 Zvýraznění důlků

Další kroky se už budou vyvíjet podle fotek na skutečném stanovišti, je tedy připraveno několik možností.

Prahování původního snímku

Jedna z možností je například rovnou prahovat původní snímek.

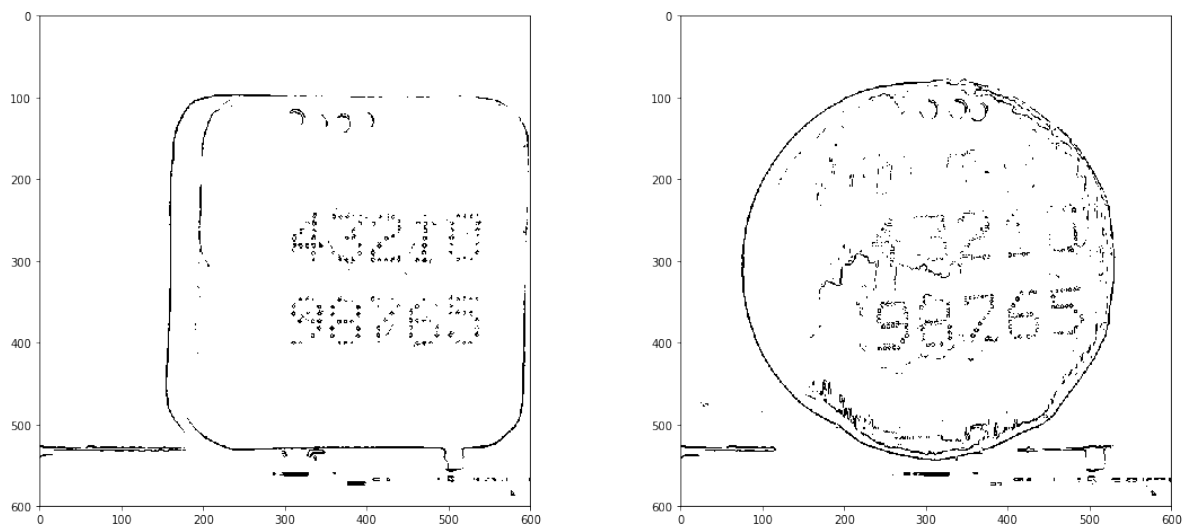


Obrázek 3.26: Prahování původních snímků (Prahy byly vybrány individuálně a pro nejlepší výsledky)

Z obrázku je viditelné, že u vývalku s mělkými důlky (vlevo) jsou čísla viditelná jen v části, jelikož spodní část byla blíže světlu a stíny nebyly tak znatelné. z druhého vývalku vystupují číslice velmi prominentně. Stále je znatelná blízkost světla v druhé řadě.

Sobelův operátor

Další možností je sobelův operátor, který je velmi často užívaný na detekci hran.

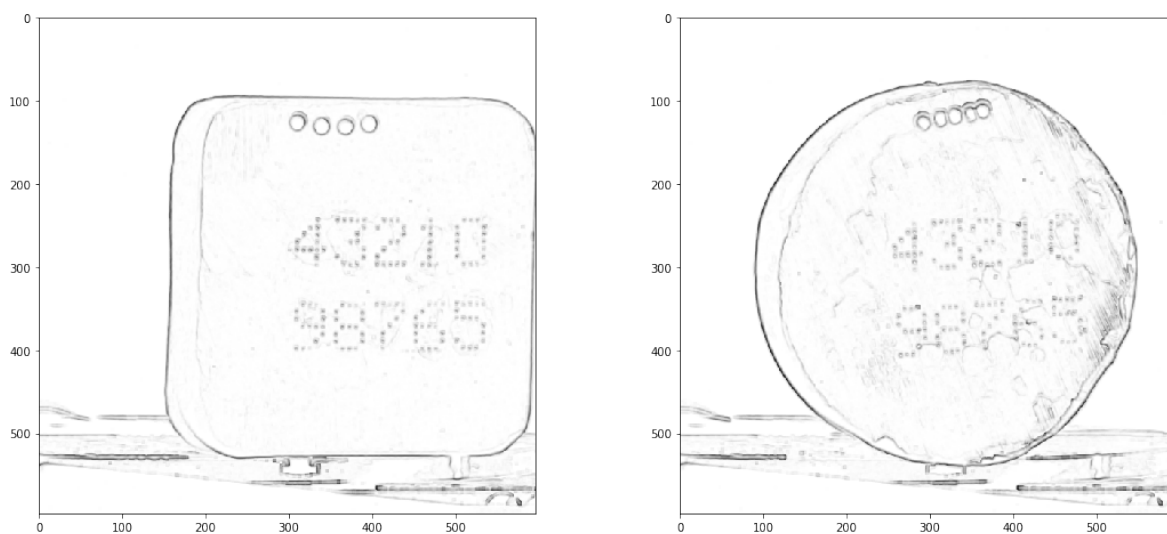


Obrázek 3.27: Úprava fotky pomocí sobelova operátoru

Z obrázku je zřejmé, že vývalek 4 s méně povrchovými vadami je mnohem čitelnější než vývalek 5, kde povrchové vady vytvářejí oblasti, které by se daly interpretovat jako čísla. Byl vyzkoušen i Laplacův operátor, ale výsledky byly ještě o poznání horší.

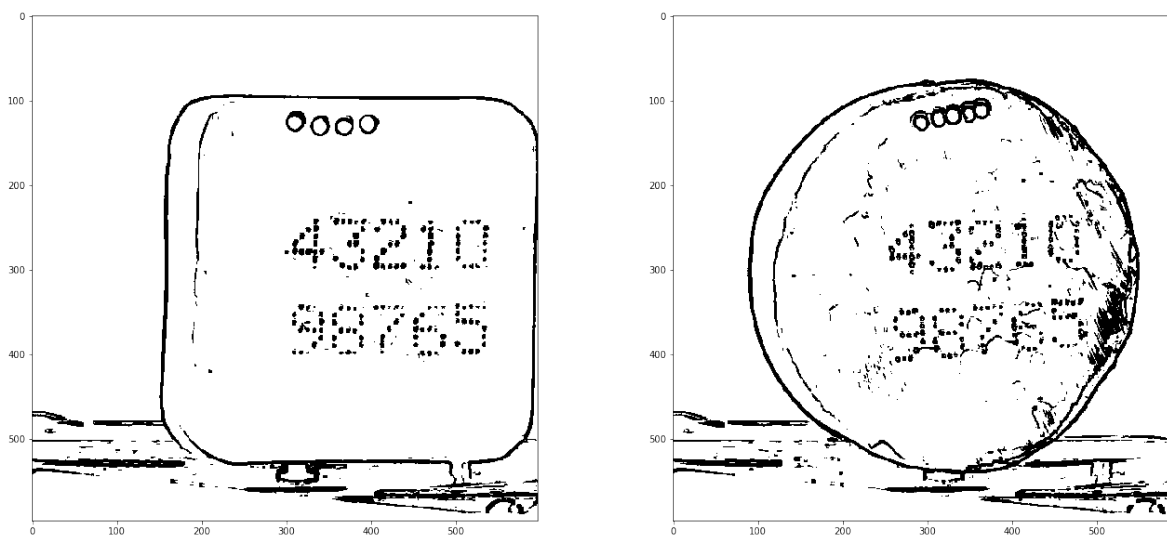
Vyhledávání šablony v obraze

Další možností je vyhledávání šablony v obraze. Nejdříve jsem pro šablonu využil čistě černou matici 4x4 a dále pak přímo snímek důlku 3.12, který má rozměry 6x6.



Obrázek 3.28: Úprava pomocí hledání černé matice 4x4

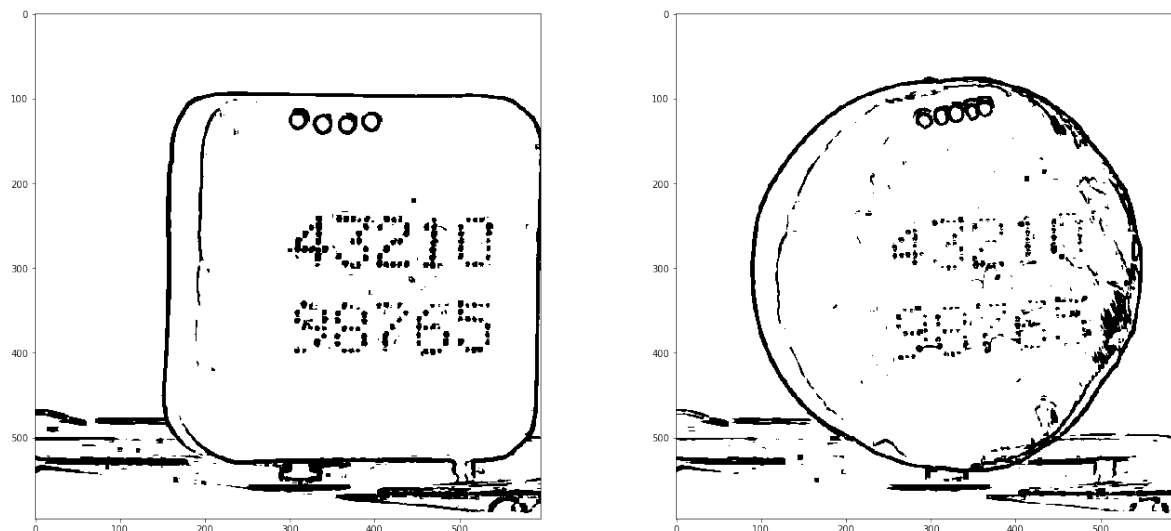
Pro názornější zobrazení jsem fotky použil prahování na prahu 0.96. Hodnoty získané z hledání šablony jsou škálované na rozsah $\langle 0;1 \rangle$, kde 1 znamená, že obraz je identický se šablonou.



Obrázek 3.29: Prahovaný obraz po hledání šablony

Z obrázku 3.29 je vidět, že výsledky jsou velmi podobné sobelovu operátoru, ale celkově jsou důlky výraznější. Vývalek 4, který nemá povrchové vady je téměř perfektní.

Při využití šablony 3.12 a prahování, získám takovéto snímky:

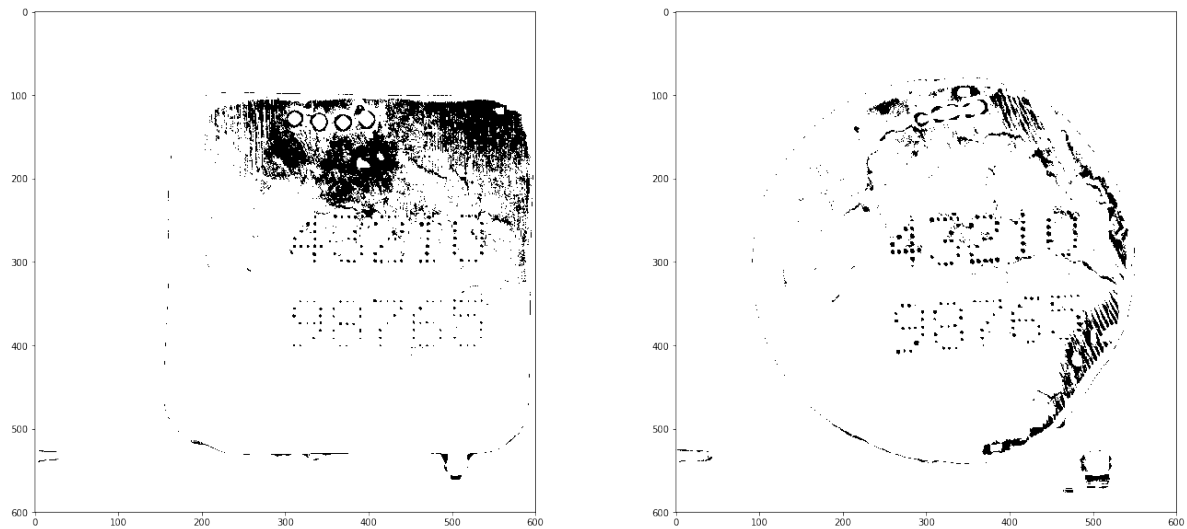


Obrázek 3.30: Prahovaný obraz po hledání šablony 3.12

Je vidět, že použití specifické šablony eliminovalo šum způsobený nečistotami, ale zároveň zmenšilo i důlky z čísel. [12], [13]

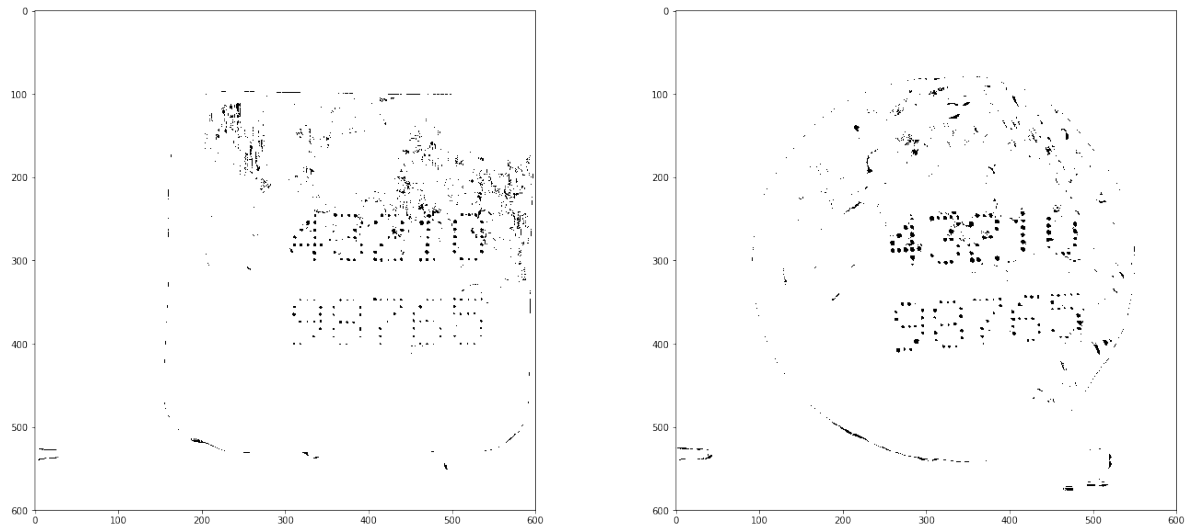
Konvoluce

Vymyslel jsem konvoluční masku, která z binárního obrazu propouští pouze uzavřené tečky. Vezme se tedy původní snímek reprezentovaný hodnotami 1 a 2. Původně binární obraz má hodnoty 0 a 1, ale s 0 se u konvoluce těžko pracuje. Pomocí konvoluční masky je každý pixel vynásoben hodnotou, a poté je konečná suma umístěna na počátek, odkud se část snímku vzala. Pro oblasti, které chci černé (oblast tečky), je rovnice jednoduchá. Hodnotu pixelu vynásobím konstantou, a to znamená, že pokud je pixel bílý (reprezentován hodnotou 2), je získáno číslo dvojnásobné. Pro oblasti, kde požaduji hodnotu 2, vezmu inverzní hodnotu pixelu a tu poté vynásobím konstantou. To znamená, že černý pixel má dvojnásobnou hodnotu pixelu bílého. Jelikož pixelů bílého pozadí je 3x více, byly jeho váhy sníženy. V okolí tečky je také "nerozhodná oblast", která je hodnocena mírněji, jelikož tyto pixely klidně mohou být černé i bílé.



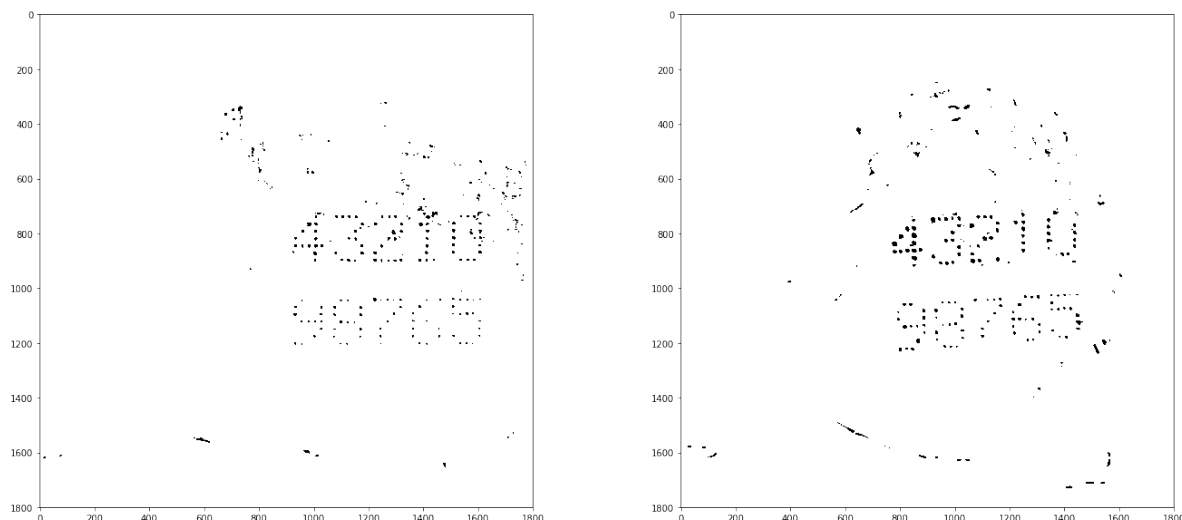
Obrázek 3.33: Prahované původní snímky

Velkou výhodou už od začátku je možnost posazení prahu výše, jelikož v následujícím korku se odstraní všechny oblasti s velkým obsahem. Pomocí hledání kontur (pixely stejného jasu tvořící souvislou oblast) a spočítání jejich obsahu můžu odstranit všechny kontury, které mají obsah větší než jedna tečka.



Obrázek 3.34: Snímky s odstraněnými velkými konturami

Poté jsou vyčištěny malé nečistoty pomocí morfologické operace otevírání s maskou ve tvaru tečky.



Obrázek 3.35: Snímky po operaci otevírání

3.5 Hodnocení zvýraznění důlků

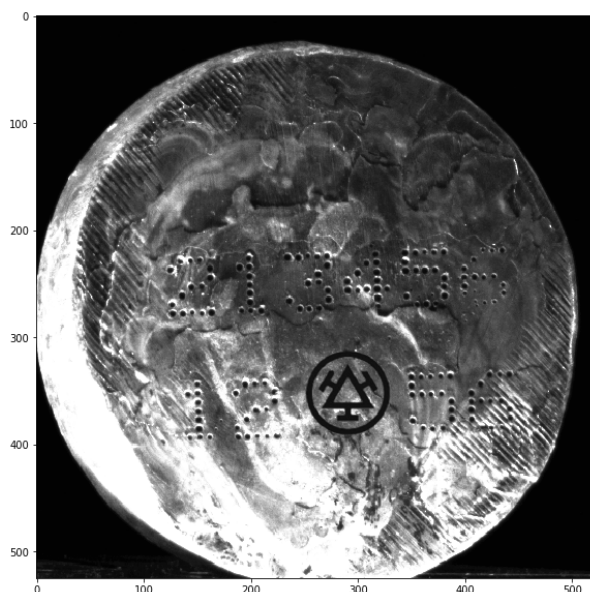
Nejvíce spokojený jsem s výsledky použití konvoluce. Tato metoda je ovšem velmi výpočetně náročná a určitě nebude využita v konečném produktu.

Metoda hledání šablon v obraze produkuje solidní výsledky, je ovšem velmi citlivá na prahování. Když je vývalek trošku hůře osvětlen, jsou číslice pohlceny stínem.

Metoda morfologických operací a hledání kontur produkuje výsledky, které jsou kvalitativně o trochu horší než výsledky konvoluce a hledání šablon, ale její silnou stránkou je možnost počítání kontur. Jelikož víme, že každá číslice má průměrně 15 důlků (kontur), je možné určit, jestli je potřeba použít silnější morfologické operace, nebo více iterací, pokud je kontur na obrázku příliš mnoho. Na obrázku 3.35 vpravo je například 256 kontur.

3.6 Orientace vývalku

Na finálních vývalcích se má mezi čísly vyrazit i logo firmy.

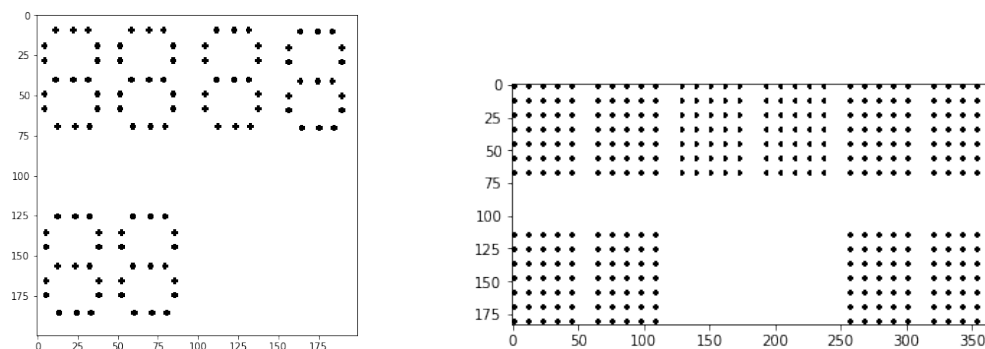


Obrázek 3.36: Snímek s uměle přidaným logem

Jelikož logo bude jednotné a je relativně symetrické, mělo by jít nalézt pomocí hledání šablony v obraze. Dále pomocí kombinace velikosti vývalku a polohy loga by mělo být možné spočítat orientaci vývalku.

Tyto vývalky bohužel ještě nebyly vyrobeny a je tedy prozatím nemožné tuto teorii ověřit.

Proto byly vyzkoušeny i jiné metody pomocí vyhledávání čísel. Například byly využity tyto dvě šablony:

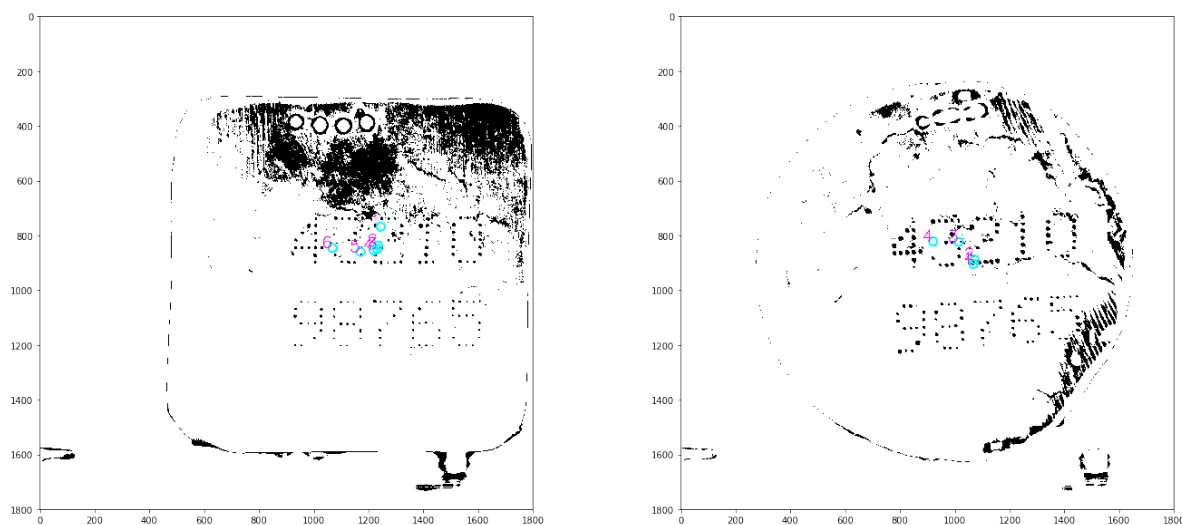


Obrázek 3.37: Šablony pro vyhledávání v obraze

kde šablona vlevo hledá levý horní roh číslcového značení pomocí podobnosti čísel 2,3,5,6,8,9. Šablona vpravo měla hledat celé značení pomocí rastru, podle kterého se všechny čísla vyrážejí. Tato metoda ovšem měla problémy se šumem v obraze a nebyla dostatečně přesná.

Další vyzkoušenou metodou je výpočet těžiště obrazu. Postupnou iterací morfologických operací je odstraňováno postupně více a více teček z obrazu a při každé iteraci je

vypočítané těžiště.



Obrázek 3.38: Nalezení těžišť obrazů

Jelikož zatím nebyly vyrobeny vývalky s větším průměrem, není zřejmé jestli těžiště míří k těžišti číslic, nebo celého vývalku.

3.6.1 Separace číslic

Pokud bude možné vývalek správně orientovat, měla by pak být možná i separace jednotlivých číslic. Díky téměř nulovému úhlu pohledu na vývalek nejsou číslice zkreslené. Dále jsou číslice raženy neproporcionálním písmem (každá číslice má pevně danou šířku) a lze pomocí oken konstantní šířky a výšky jednotlivé číslice rozdělit.

Kapitola 4

Závěr

Cílem této práce bylo rozpoznání unikátního značení vývalků pomocí strojového vidění. Teoretická část se věnuje metodám běžně využívaných ve strojovém vidění.

Už od prvního pohledu na vývalek bylo jasné, že se nebude jednat o snadný úkol. Velmi mělké důlky v kombinaci s hrubým povrchem vývalku znamenaly, že běžně prodávané metody osvětlení nebyly dostačující a musel se navrhnout speciální způsob osvětlení. U strojového vidění si správným osvětlením lze ušetřit spoustu práce, a proto byly nesmírně důležité úvodní pokusy s různými typy osvětlení.

Výběr kamery a objektivu byl proveden pomocí měření a pokusů. Jelikož byla dána maximální vzdálenost od čela vývalku a velikost jednoho důlku, byl výběr kamery a objektivu jednoznačný.

Vývoj algoritmu pro rozpoznání jednotlivých číslic byl omezen nedostatkem vývalků a programováním hlavice na vyrážení důlků, která stále ještě nemá finální verzi značení. Bylo tedy připraveno několik způsobů, jak důlky zvýraznit a lokalizovat jejich přibližnou polohu.

Jelikož ještě není hotový finální algoritmus, je nemožné vybrat postačující PC pro tuto úlohu. Operace, které bude provádět PC, by neměly být nijak náročné a nejsou tedy vysoké požadavky na výkon. Nutné je, aby v perifériích byl port USB 3.0 nebo lepší.

Do budoucna je třeba finalizovat značení vývalků a vyrobit více vývalků rozdílných velikostí nebo stanoviště zkonstruovat pro získání mnohem více snímků. Díky tomu bude možné lépe zkalibrovat připravené algoritmy a navrhnout neuronovou síť, která bude už rozpoznávat jednotlivé číslice.

Bibliografie

- [1] MURESAN, Mircea Paul; SZABO, Paul Andrei; NEDEVSCHI, Sergiu. Dot Matrix OCR for Bottle Validity Inspection. In: *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*. 2019, s. 395–401. Dostupné z DOI: 10.1109/ICCP48234.2019.8959762.
- [2] PATKI, Siddharth Nitin; JOSHI, Madhuri; KULKARNI, Abhishek Ninad. Dot matrix text recognition for industrial carton classification. In: *2015 International Conference on Industrial Instrumentation and Control (IIC)*. 2015, s. 777–782. Dostupné z DOI: 10.1109/IIC.2015.7150847.
- [3] SMITH, R. An Overview of the Tesseract OCR Engine. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. 2007, sv. 2, s. 629–633. Dostupné z DOI: 10.1109/ICDAR.2007.4376991.
- [4] HOSOZAWA, Kento; WIJAYA, Ricky Hendra; LINH, Tran Duy; SEYA, Hiroaki; ARAI, Masayuki; MAEKAWA, Tsukasa; MIZUTANI, Kozo. Recognition of Expiration Dates Written on Food Packages with Open Source OCR. *International Journal of Computer Theory and Engineering*. 2018, roč. 10, č. 5, s. 170–174. Dostupné z DOI: 10.7763/ijcte.2018.v10.1220.
- [5] CODE.GOOGLE.COM. “OCR engine for Japanese language. Dostupné také z: <https://code.google.com/archive/p/nhocr/>.
- [6] BREUEL, Thomas. The OCRopus open source OCR system. In: 2008, sv. 6815, s. 68150. Dostupné z DOI: 10.1117/12.783598.
- [7] DAVIES, E. R. *Computer Vision: Principles, Algorithms, Applications, Learning*. 5th ed. Academic Press, 2017. ISBN 9780128095751.
- [8] Convolutions for Images [online] [cit. 2022-05-13]. Dostupné z: https://d2l.ai/chapter_convolutional-neural-networks/conv-layer.html.
- [9] SONKA, Milan; HLAVAC, Vaclav; BOYLE, Roger. *Image processing, analysis and machine vision*. 1993. vyd. New York, NY: Springer, 2013.
- [10] ALLIOUI, Hanane; SADGAL, Mohammed; FAZZIKI, Aziz. A robust multi-agent Negotiation for advanced Image Segmentation: Design and Implementation. *Inteligencia Artificial*. 2019, roč. 22, s. 102–122. Dostupné z DOI: 10.4114/intartif.vol22iss64pp102-122.
- [11] The OpenCV Reference Manual. 4.5.5. vyd. 2021.
- [12] MINICHINO, Joe. *Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning, 3rd Edition*. 3rd ed. Packt Publishing Limited, 2020. ISBN 1789531616.
- [13] BRADSKI, G. The OpenCV Library. Dr. Dobb’s Journal of Software Tools. 2000.