

Odečítání hodnot z analogových přístrojů

Bc. Luboš Toman

Diplomová práce
2024



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

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

Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Luboš Toman**
Osobní číslo: **A22409**
Studijní program: **N0714A150006 Automatické řízení a informatika v průmyslu 4.0**
Forma studia: **Kombinovaná**
Téma práce: **Odečítání hodnot z analogových přístrojů**
Téma práce anglicky: **Reading consumption data from analog meters**

Zásady pro vypracování

1. Zpracujte rešerši s problematikou strojového vidění a optického rozpoznání znaků.
2. Navrhněte systém pro osvětlení měřiče a snímání obrazu založené na mikrokontroléru.
3. Navrhněte systém pro agregaci dat z mikrokontrolérů postavený na vestavěném zařízení Raspberry Pi.
4. Vytvořte aplikaci pro rozpoznávání znaků z obrázků měřičů, ukládání hodnot do databáze a vizualizaci dat.
5. Otestujte zařízení a aplikaci na několika měřičích.

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

Seznam doporučené literatury:

1. SZELISKI, Richard. Computer vision: algorithms and applications. Texts in computer science (Springer). London: Springer, 2011. ISBN 978-184-8829-343.
2. LUTZ, Mark. Learning Python. 4th ed. Beijing: O'Reilly, 2009. ISBN 978-0-596-15806-4.
3. HALFACREE, Gareth, 2019. The Official Raspberry Pi Beginner's Guide. 4th edition. Raspberry Pi Trading. ISBN 978-1-912047-62-8.

Vedoucí diplomové práce: **Ing. Jakub Novák, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání diplomové práce: **8. prosince 2023**

Termín odevzdání diplomové práce: **27. května 2024**

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



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

Ve Zlíně dne 8. prosince 2023

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně, dne

.....
podpis studenta

ABSTRAKT

Tato práce se zabývá návrhem a implementací systému sloužícím ke čtení a ukládání dat z měřících přístrojů energií, tedy vodoměrů, plynůměrů a elektroměrů na principu strojového vidění. Součástí řešení je pořízení obrazu pomocí periferního zařízení, jeho odeslání do centrální jednotky, která provede extrakci dat a jejich uložení do souboru pro potřeby uživatele.

Klíčová slova: Strojové vidění, OpenCV, optické rozpoznání znaků, Python, Arduino

ABSTRACT

This work deals with a draft and implementation of machine vision system used for reading and storing data from the energy measuring instruments, namely water, gas and electricity meters. The solution contains image acquisition with peripheral device, sending the image to the central unit, which will then execute extraction of data and saving those to the file for users' needs.

Keywords: Machine vision, OpenCV, optical character recognition, Python, Arduino

Tímto děkuji panu Ing. Jakubu Novákovi, Ph.D. za odborné rady, připomínky a samozřejmě také vedení této diplomové práce.

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

OBSAH

ÚVOD	6
I TEORETICKÁ ČÁST	7
1 STROJOVÉ VIDĚNÍ	8
1.1 ZÁKLADNÍ KOMPONENTY STROJOVÉHO VIDĚNÍ	9
1.1.1 Osvětlení scény	9
1.1.1.1 Způsoby osvětlení scény	10
1.1.2 Objektiv	13
1.1.3 Obrazový senzor.....	13
1.1.4 Zpracování obrazu.....	14
1.2 OPTICKÉ ROZPOZNÁNÍ ZNAKŮ	15
1.2.1 Typy programů OCR.....	15
1.2.2 Průběh OCR	16
1.3 STROJOVÉ UČENÍ	17
1.3.1 k – Nearest Neighbors	19
1.4 BINÁRNÍ OBRAZ.....	20
1.5 KNIHOVNA OPENCV	21
1.5.1 Odstranění šumu.....	21
1.5.2 Prahování obrazu.....	22
1.5.3 Kontury	24
II PRAKTICKÁ ČÁST	25
2 HARDWARE	26
2.1 ESP-32 CAM.....	26
2.2 OV2640	27
2.3 RASPBERRY PI 4 MODEL B	28
3 INTEGRACE OCR	29
3.1 ČTENÍ OBRAZOVEK.....	29
3.2 ZPRACOVÁNÍ OBRAZU	30
3.2.1 Úprava fotografie a nalezení číslic.....	30
3.2.2 Rozpoznání číslic	34
3.2.3 Shromáždění dat.....	37
3.3 FIXACE ZAŘÍZENÍ.....	38
3.3.1 Elektroměr	38
3.3.2 Vodoměr.....	39
3.3.3 Plynoměr	42
3.4 NAPÁJENÍ	43
4 VYHODNOCENÍ	45
4.1 ROZPOZNÁNÍ ČÍSLIC VODOMĚRU	46
4.2 ROZPOZNÁNÍ ČÍSLIC ELEKTROMĚRU	49
4.3 ROZPOZNÁNÍ ČÍSLIC PLYNOMĚRU	52
4.4 KOMENTÁŘ	55
ZÁVĚR	57
SEZNAM POUŽITÉ LITERATURY	58

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	61
SEZNAM OBRÁZKŮ	62
SEZNAM PŘÍLOH.....	64

ÚVOD

Téma této práce je problematika strojového vidění. Šíře záběru této disciplíny je značně rozsáhlá, protože schopnost zraku je pro stroje velmi žádaná meta takřkajíc v jakémkoliv odvětví. Nicméně s různými typy aplikací přicházejí různé typy úrovně složitostí. Komplikace spojené se systémem nuceným korektně v požadovaném časovém horizontu reagovat na dynamické prostředí s velkým množstvím změn nastávajících v krátkých intervalech, je velmi vzdálená jednodušším aplikacím, jako je identifikace předmětů na statických obrazech.

Opakované rozpoznávání čísel z periodicky pořizovaných fotografií normovaných měřících přístrojů implementované v této práci spadá do druhé kategorie, čemuž napomáhá uniformita číslic ať už na digitálních obrazovkách elektroměrů, či analogových zobrazovacích plochách, přičemž v tomto případě se naráží na určité množství problémů, jako nestejná poloha číslic, číslice v mezipoloze, a podobně.

Cílem práce je vytvořit systém pro čtení a ukládání hodnot z měřících přístrojů energií, kdy periferní jednotky vybavené kamerou pořídí obraz daného přístroje a tento odešlou do centrální vyhodnocovací jednotky, která obraz zpracuje a hodnoty přečtené z obrazovky přístroje uloží k dalšímu zpracování uživatelem.

V podstatě úloha této aplikace patří do oblasti domácí automatizace, jejímž účelem je úspora času, který by jinak člověk strávil obíháním měřících přístrojů a odečtem jimi zobrazovaných hodnot za účelem kontroly vlastní spotřeby. S tímto řešením bude mít aktuální i historické výsledky vzdálené pouze na pár kliknutí. Navíc je tímto způsobem možné docílit vyšší hustoty odečtů, v závislosti na periodě pořizovaných vzorků, a tím získat i přehled o změnách spotřeby v jednotlivých částech dne, případně je možné získaná data statisticky vyhodnocovat.

Součástí práce je seznámení s platformou Arduino, respektive klonu od firmy Espressif, jejím programováním, a následným návrhem a vytvořením modulu schopného snímat obrazy číselníků analogových měřících přístrojů a tyto poté zasílat k dalšímu zpracování.

Dalším krokem je identifikace efektivního a spolehlivého přístupu rozpoznání číselných znaků z obrazového souboru, jejich převodu na text a implementace tohoto přístupu do zařízení, jehož účelem bude přijímání dat z periférií, jejich zpracování a poskytnutí uživateli.

I. TEORETICKÁ ČÁST

1 STROJOVÉ VIDĚNÍ

Problematika zpracování obrazu je rozsáhlou a přitom důležitou součástí automatizace, jelikož kamerové systémy dokážou neúnavně bez ztráty pozornosti sledovat vymezený prostor a je tedy nasnadě jejich použití v aplikacích neefektivních, případně nevhodných pro nasazení člověka, ať už z důvodu nedostatečné rychlosti lidského „hardwaru“ například při okamžitém čtení a včasné vyhodnocování čárových kódů, nebo v podmínkách rutinní činnosti, jako je kontrola kvality, kdy hrozí ztráta pozornosti, což může mít za následek přehlédnutí vadného výrobku mezi záplavou bezchybných, popřípadě z čistě praktického hlediska v aplikacích, kde by použití člověka na danou činnost bylo neekonomické, či využití v procesech, kde lze pomocí technologie dosáhnout úspory času a tím i větší efektivity ať už v pracovním nasazení, nebo v osobním životě.

Zatímco v případě člověka, v rovině oko-mozek, dochází k přečtení obrazu a vyhodnocení jeho obsahu na takřka instinktivní úrovni, tak u strojového vidění je tento problém o několik řádů složitější. Především už jen proto, že počítač na rozdíl od člověka nedokáže extrapolovat na základě zkušenosti, a ačkoliv v tomto případě bylo v nedávné době dosaženo určitého pokroku s rozvojem strojového učení, neuronových sítí, či umělé inteligence obecně, stále dochází k narážení na určité technologické limity, které jsou zřejmější, čím více má být daná aplikace obecnější. Zjednodušeně je možné říct, že složitost programu se odvíjí od složitosti aplikace, ve které má k rozpoznání docházet a čím variabilnější a nepředvídatelnější je prostředí, ve kterém musí systém spolehlivě fungovat a čím více objektů musí brát v potaz a správně vyhodnotit jak jejich status, tak svou reakci na ně, tím vyšší je nespolehlivost a chybovost celé aplikace.

Nicméně, jak už bylo naznačeno, při jednodušších, či jednoúčelových implementacích práce s obrazem není tento problém spolehlivosti vyhodnocování zdaleka tak markantní. Příkladem může být právě výstupní kontrola kvality výrobků, kdy ve většině případů jsou dané díly snímány vždy ve stejném úhlu ve stabilních podmínkách, kdy pak kontrolní program nemusí řešit nic jiného než kontrolu velmi často předdefinovaných pozic v obraze.

Další taková aplikace by byla rozpoznání textu, či znaků, kdy vstupem do procesu je digitální obraz často papírového dokumentu a požadovaným výstupem je textový soubor v dále počítačově zpracovatelném formátu. Složitost se zde odvíjí od typu písma, kdy tištěný jednolitý font je podstatně méně náročný na identifikaci, než rukopis.

Strojové vidění je tedy automatizovaná schopnost počítače rozpoznat, identifikovat a rozřadit (klasifikovat) specifické objekty ve zpracovávaném obraze, ať už jde o předměty,

zvířata, lidi, či pouze text. Výstupem tohoto procesu pak není další obraz, ale data extrahovaná či odvozená z daného obrazu ve formátu požadovaném pro další práci s nimi, většinou v zájmu integrace nových řešení pro průmyslovou nebo domácí automatizaci.

1.1 Základní komponenty strojového vidění

Celý systém strojového vidění je složen z čtyř základních komponent, jejichž vzájemná integrace a spolupráce je klíčová pro korektní funkci a použitelný výstup. Tyto komponenty jsou: osvětlení, objektiv, kamera, nebo také obrazový senzor a zpracování obrazu. [1]

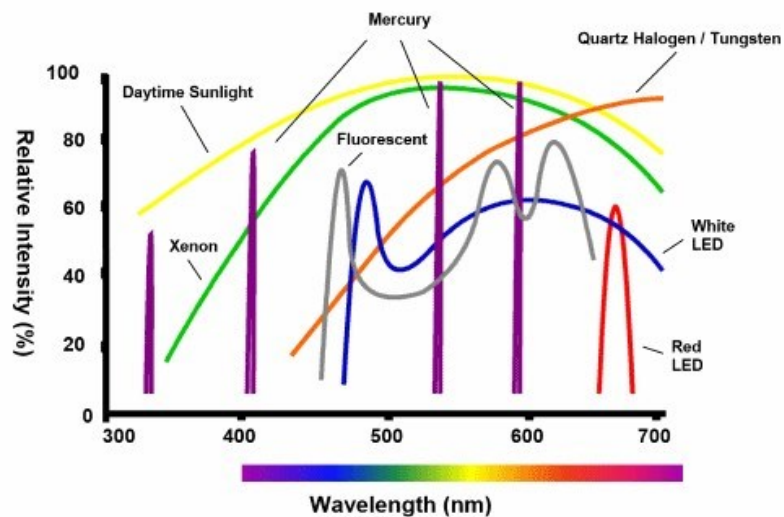
1.1.1 Osvětlení scény

Významnou, takřikajíc kritickou, součástí získávání obrazu určeného ke strojovému zpracování je korektní osvětlení scény v závislosti na vlastnostech prostředí, potřebách snímacího zařízení a typu aplikace. Důležitou vlastností správně vybraného osvětlení je vytvoření neměnných podmínek, díky kterým lze dosáhnout opakovatelných výsledků zajišťujících stále stejnou kvalitu obrazu neohledně na množství opakování, či okolní podmínky.

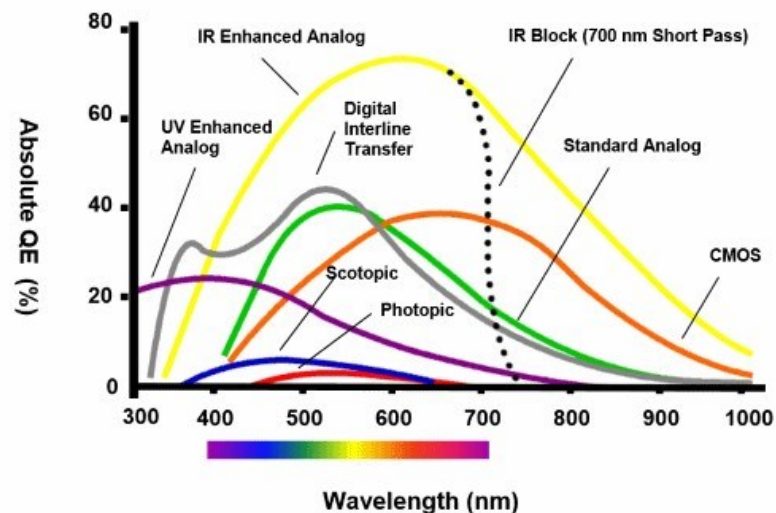
V případě rušivých vlivů prostředí, jako střídání různé intenzity světla v průběhu dne, či změny stavu umělého osvětlení okolního prostoru může být také vhodné provést kroky vedoucí k minimalizaci působení vnějších vlivů, jako například instalace odstínění, či absolutní izolace scény od jakýkoliv světelných vlivů, které nedokážeme kontrolovat.[1]

Také je vhodné sladit typ přísvitu způsobem, aby vlnová délka jeho nejvyšší intenzity byla co nejkompatibilnější s pásmem největší citlivosti použitého typu obrazového senzoru.

Z obrázku je dále patrná dominance slunečního světla po celé šířce pásma, z čehož plyne, že jakýkoliv kamerový systém odladěný na práci v denním světle bude mít významné potíže s udržení stejné kvality zobrazování v nočních hodinách, pokud nebudou dodatečným přísvitem suplovány podobné podmínky. Nejeefektivnější přístup by v tomto případě bylo zamezení jakémukoliv přístupu denního světla a optimalizace systému snímání obrazu na stálý zdroj osvětlení jednotného typu.



Obrázek 1. Intenzita světelných zdrojů v závislosti na vlnové délce [2]



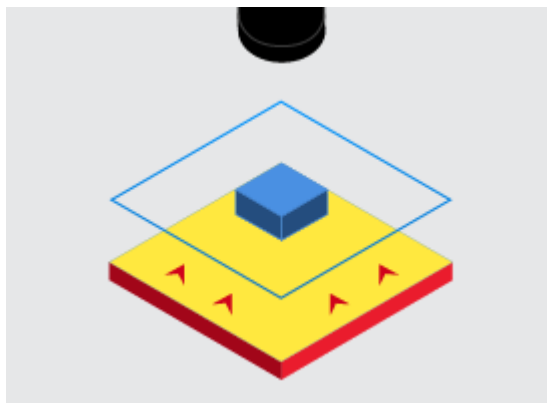
Obrázek 2 Efektivita obrazových senzorů při dané vlnové délce [2]

Chybná implementace způsobu osvětlení může mít za následek ztrátu informací, či úplné selhání jejich extrakce z důvodu nedostatečné kvality pořízeného obrazu.

1.1.1.1 Způsoby osvětlení scény

Zadní/spodní osvětlení

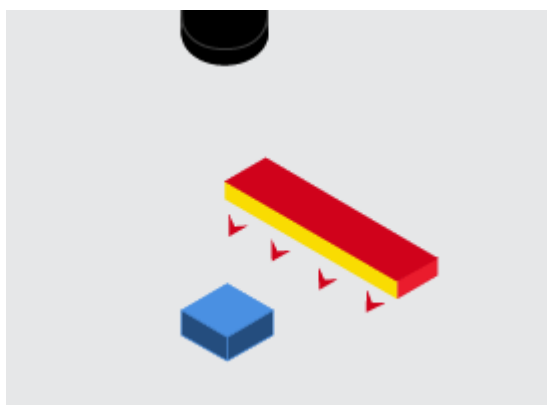
V tomto případě je osvětlení (žlutý čtverec na obrázku níže) rovnoměrně rozprostřeno tak, aby z pohledu snímače (černý objekt) rovnoměrně osvětlovalo celý snímáný objekt (modrý čtverec) ze zadní části, zvýrazňovalo jeho siluetu a jakékoliv případné perforace či okrajové nedokonalosti. Případně je také možné provést měření vnějších rozměrů a kontrolu správnosti tvaru daného objektu. Tento přístup je využíván, když není potřeba kontrola samotného povrchu. [1] [2]



Obrázek 3. Zadní osvětlení [1]

Pruhové osvětlení

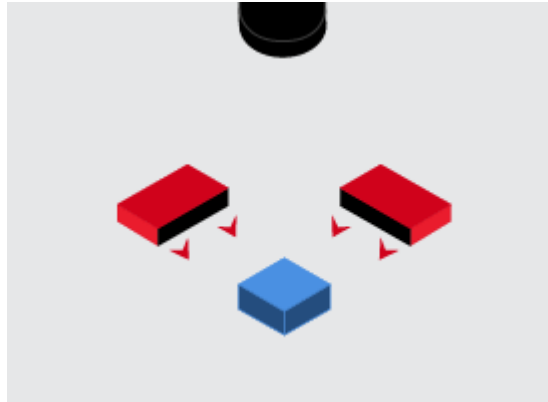
Osvětlení objektu, nebo jeho hrany pomocí jednolitého světelného pásu ve zvoleném prostoru. Je možné použít více zdrojů najednou pro dosažení kompletního osvětlení celého objektu. V závislosti na vzájemném úhlu mezi světlem a kamerou je možné kontrolovat intenzitu odrazu světla, pokud si to aplikace žádá. Tento způsob je používán ke zvýrazňování povrchových vlastností objektu. [1] [2]



Obrázek 4. Pruhové osvětlení [1]

Osvětlení temným polem

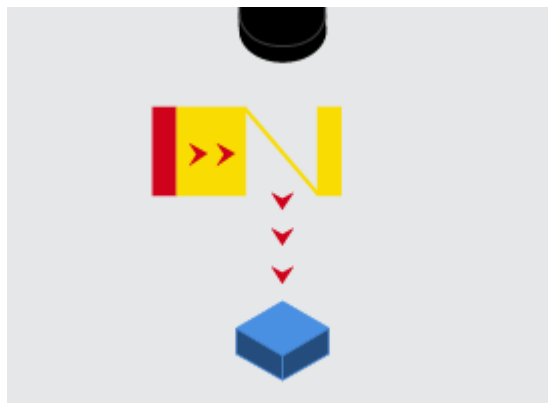
Objekt je osvětlován zdroji světla, která jsou s ním umístěna téměř v jedné rovině a osvětlení na něj tedy dopadá pod velmi ostrým úhlem. Cílem je dosáhnout stavu, kdy jakékoliv povrchové nedokonalosti jako škrábance, vrypy a otisky odrážejí světlo přímo na kameru, díky čemuž jsou na fotografii velmi výrazné, hlavně v porovnání se zbytkem povrchu, který zůstává tmavý. [1] [2]



Obrázek 5. Osvětlení temným polem [1]

Koaxiální osvětlení

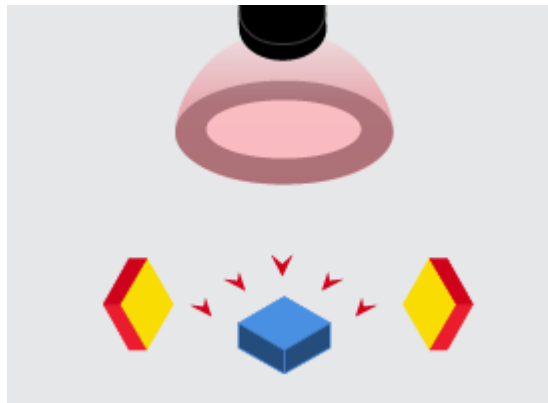
Princip koaxiálního osvětlení spočívá ve vysílání světelného paprsku kolmo na objekt přímo v ose objektivu kamery, čímž dojde ke zvýraznění povrchu kolmého ke kameře a ztmavení jeho částí, které vůči objektivu svírají jiný úhel. Tento přístup minimalizuje stíny a odlesky. [1] [2]



Obrázek 6. Koaxiální osvětlení [1]

Kupolové difuzní osvětlení

Difuzní technika využívající rozptylování světla uvnitř kupole za účelem rovnoměrného rozložení světla po povrchu a minimalizace odlesků na odrazivých materiálech. [1] [2]



Obrázek 7. Kupolové osvětlení [1]

1.1.2 Objektiv

Objektiv je optický prvek skládající se z jedné nebo více čoček, popřípadě soustavy čoček a zrcadel, vytvářející skutečný obraz objektu za pomoci shromažďování příchozího světla z pozorovaného předmětu a jeho soustředěním na optický senzor. V aplikacích strojového vidění se mohou vyskytovat objektivy s automatickým zaostřováním pomocí motorizovaných objektivů, nebo tekutých čoček stejně jako běžné spotřebitelské kamery. [1] [3]

Kromě toho lze na kamerách ve strojovém vidění nalézt objektivy i s ručně nastavitelným ostřením, případně verze, kdy je na kameře umístěna nehybná avšak vyměnitelná soustava čoček, jež bývá vybrána přesně s ohledem na podmínky aplikace, kde byl ten který objektiv nasazen. [1] [3]

1.1.3 Obrazový senzor

Schopnost kamery zachytit obraz dále závisí na obrazovém senzoru. Ty bývají založeny buď na CCD, nebo CMOS technologii a jejich účelem je převedení obrazu, tvořeného zachyceným světlem na jeho digitální interpretaci. [1]

Obraz je pak celek složený z jednotlivých pixelů, což je nejmenší adresovatelná a manipulovatelná část obrazu nesoucí informaci o své barvě či intenzitě. Čím více pixelů výsledný obraz obsahuje, tím více detailů z něj lze získat, což ve výsledku bude mít vliv na kvalitu výstupu po zpracování dané fotografie. [1]

CCD – Pole kondenzátorů, z nichž každý nese náboj o velikosti odpovídající intenzitě pixelu, který představuje. Celý koncept sběru nábojů z jednotlivých kondenzátorů za účelem expozice obrazu pak připomíná posuvný registr. Řídící obvod vynucuje postupný transfer náboje z jednoho kondenzátoru na druhý, aby pak poslední v poli uvolnil tento náboj do výstupního zesilovače, jehož výstupem je poté zpracovatelné napětí. [1] [4]

CMOS – Skládá se z pole křemíkových fotodiod a CMOS tranzistorových přepínačů sloužících jako samostatný zesilovač pro každý jeden pixel. Levnější, rychlejší a méně energeticky náročné než CCD, nicméně disponuje menším prostorem k zachytávání světla a disponuje vyššími tendencemi ke generování elektrického šumu v důsledku fluktuací výkonů jednotlivých fotodiod a zesilovačů v rámci celého sběrného pole. [1] [5]

Kromě typu konstrukce, je možné obrazové senzory dělit dále podle způsobu získávání obrazu. V tomto případě bychom mluvili o dělení na řádkové a maticové kamery. [1]

Při použití maticových kamer dochází k sejmutí celého obrazu v jednom okamžiku s tím, že výsledný obraz bude mít šířku a výšku odpovídající rozlišení použitého senzoru, protože v jejím případě je snímací plocha tvořena 2D polem fotodiod. Tento typ kamery je vhodný ke snímání statických objektů, které je možné umístit do vizuálního pole kamery v celé velikosti tak, aby na výsledné fotografii byly zachovány potřebné detaily. Nicméně problém nastává například v okamžiku, kdy je nutné nafotit objekt ve vysokém rozlišení s dostatečnými detaily za pohybu, tak aby výsledná kvalita nebyla ovlivněna například rozmazáním. [1]

Pro tento typ aplikace jsou vhodné takzvané řádkové kamery. Na rozdíl od maticových kamer nedisponují celým polem fotocitlivých článků, ale pouze jediným řádkem. Nejsem tedy schopny pořídit celý obraz v jednom okamžiku, ale místo toho provedou množství řádkových snímků za konstantního pohybu objektu, nebo svého, a z nich poté poskládají dohromady celý 2D obraz dané scény. Při snímání je nutné dodržovat stále stejnou rychlost, aby nedocházelo k deformaci výsledného obrazu. Tyto kamery bývají použity například v kancelářských skenerech k pořizování digitální verze papírových dokumentů. [1]

1.1.4 Zpracování obrazu

Zpracování obrazu je součástí strojového vidění, která zahrnuje implementaci extrakce žádaných objektů z pořízeného obrazu. V závislosti na typu zařízení může probíhat už na kameře, která tento obraz poříдила, nebo dojde k odeslání fotografie na jiné zařízení a vyhodnocení se provede tam.

V základě se sestává z několika kroků, mezi které patří předzpracování obrazu, kdy dojde v rámci možností k vyčištění fotografie, tedy k odstranění šumu a odlesků, jejímu následnému převedení do formátu vhodnému pro další krok a případnému zvýraznění požadovaných vlastností za účelem ulehčení detekce.

Poté program identifikuje hledané objekty, které extrahuje a dále se nad nimi provede operace, jejichž průběh záleží na požadavcích konkrétní aplikace. Například v případě této práce dojde k nalezení, extrakci a poté identifikaci čísel.

1.2 Optické rozpoznání znaků

Ústřední bod zpracovávané úlohy, tedy to, co nespadá pod vzájemnou integraci všech použitých zařízení, je ve své podstatě disciplína strojového vidění označovaná jako Optical Character Recognition (OCR), zabývající se konverzí obrazů tištěného, psaného, či jiným způsobem vytvořeného textu na strojovou/počítačovou reprezentaci souboru znaků. Jde tedy ve své podstatě o digitalizaci textu, což je mimo jiné výhodné z hlediska úspory fyzického prostoru, který by jinak byl zabrán v případě potřeby archivace dat v papírové podobě. Nehledě na čas potřebný ke katalogizaci, správě a následnému vyhledávání potřebných dokumentů. [6] [7] [8]

V minulých letech byl celkem běžný a jediný rychlý způsob digitalizace (pokud neuvažujeme přímý ruční přepis) skenování dokumentů, což vzhledem k tomu, že výsledkem skenování vznikla vlastně digitální fotografie, bylo znovu náročné na kapacity úložišť, ačkoliv se od papírových dob změnil typ používaného prostoru. [6] [7] [8]

Způsob digitalizace pomocí rozpoznání textu je tedy mnohem efektivnější z hlediska zabraného úložného prostoru a dalšího zpracování daného textu, jednak díky efektivitě elektronických dokumentů a pak také díky tomu, že se konečně nachází ve formátu, se kterým dokážou pracovat i právě počítače. [6] [7] [8]

Nicméně stále platí nutnost výstupní kontroly výsledného textu, jelikož kvalita výstupu je mimo jiné závislá na stavu vstupních materiálů. [6] [7] [8]

1.2.1 Typy programů OCR

Programy optického rozpoznávání textu je dále možné dělit podle typu implementovaného přístupu použitého k identifikaci.

Jednoduché optické rozpoznávání znaků

V tomto případě je v rámci databáze programu uloženo množství obrazových vzorů textu, které se navzájem liší fonty. Při čtení textu porovnává obrazy znak po znaku s těmi uloženými v interní paměti. V případě, že tento postup provádí s celými slovy, je program nazýván jako Optické rozpoznávání slov. [6] [7] [8]

Tento přístup má limity dané především faktem, že není schopen úspěšně zpracovat typ písma, který nemá uložený. Z tohoto důvodu je nepoužitelný ke čtení ručně psaného textu, jelikož je nereálné podchytit a uložit všechna písmena a jejich odchylky už jen v rámci textu psaného jedinou osobou, natož držet informaci o rukopisu členů nějaké skupiny netriviální velikosti. [6] [7] [8]

Chytré optické rozpoznávání znaků

V tomto případě byla problematika řešena pomocí využití inteligentních technik, například strojového učení a tedy neuronových sítí. Na rozdíl od předchozího způsobu tento přístup nepracuje s porovnáváním celých tvarů, ale spíše postupuje podle identifikovaných vlastností jednotlivých znaků a navíc dokáže pracovat s variacemi na tyto vlastnosti, což umožňuje flexibilně číst i písmo, na které nebyl implicitně naučen. Samozřejmě, tato schopnost záleží na kvalitě modelu umělé inteligence a jeho učení. [6] [7] [8]

1.2.2 Průběh OCR

Získání obrazu

Libovolným elektronickým zařízením schopným tvorby snímků je získán obraz v dostatečné kvalitě pro další zpracování. Pojem dostatečná kvalita se může lišit v závislosti na možnostech rozpoznávacího programu a jeho schopnostech kompenzace neostrotí, rozmazání, a nerovností obrazu. [7] [8]

Preprocessing/Předzpracování

Program pro rozpoznávání znaků provede korekci/minimalizaci odlesků na fotografii, převedení obrazu na binární data, srovnání dokumentu ať už jde o pootočení, či deformaci způsobenou neoptimální pozicí objektivu, očištění od digitálního šumu a podobně. Vesměš úpravy, které v následujícím kroku budou mít vliv na lepší kvalitu čtení a tedy i výsledku. [7] [8]

Rozpoznání textu

Zde jde o samotnou identifikaci znaků. V zásadě jde o dva přístupy: Porovnávání vzoru a extrakce vlastností. [7] [8]

Při porovnávání vzoru je každý jeden znak obraze zvaný glyf, například reprezentace čísla 9, viditelná na obrázku 8 níže, zvlášť izolován a porovnáván s dalšími podobně uloženými šablonami v databázi. Při největší podobnosti mezi zkoumaným a uloženým glyfem je

pak vyhlášena shoda a zkoumaný glyf je prohlášen za znak odpovídající šabloně. Tento přístup funguje především v případě, že uložené glyfy odpovídají tvarem a rozměry těm čteným. Tedy hlavně v případech, kdy je znám používaný font, je pro něj vytvořena sada šablon ve velikosti odpovídající výstupu čtecího zařízení, či je program schopen vykompenzovat rozdíly ve velikostech vstupních znaků a uložených glyfů. [7] [8]

Postup při extrakci prvků je takový, že izolované glyfy jsou rozloženy na základní vlastnosti jako linky, jejich směry a křížení, uzavřené smyčky a podobně, kteréžto jsou pak zase porovnávány s podobnými vlastnostmi již uložených glyfů za účelem nalezení největší podobnosti.[7] [8]

V zásadě hlavní rozdíl mezi těmito dvěma výše zmíněnými postupy je ten, že zatímco při porovnání vzoru jsou glyfy se šablonami porovnávány jako celky, tak při extrakci dochází k vyhodnocování shod jejich dílčích rozložených vlastností postačujících k jejich jednoznačné identifikaci.[7]



Obrázek 8. Glyfy čísla 9

Postprocessing

Zde dochází už k výslednému převedení přečteného textu do formátu počítačového dokumentu zvoleného uživatelem. [7] [8]

1.3 Strojové učení

Strojové učení je odvětví umělé inteligence zaměřené na zpracování dat a použití statistických algoritmů zacílených na použití UI takovým způsobem, aby imitovala lidský přístup k učení z daných vstupů, generalizaci a rozhodování za pomoci použití získaných znalostí. [9] [10]

Základní koncept při zpracování dat spočívá ve statistickém učení a optimalizaci metod, které umožňují počítači analyzovat datasey a identifikovat vzory v nich, které poté používá buď k vyhledávání těchto vzorů v dalších datech a poté jejich prezentaci v uživatelsky

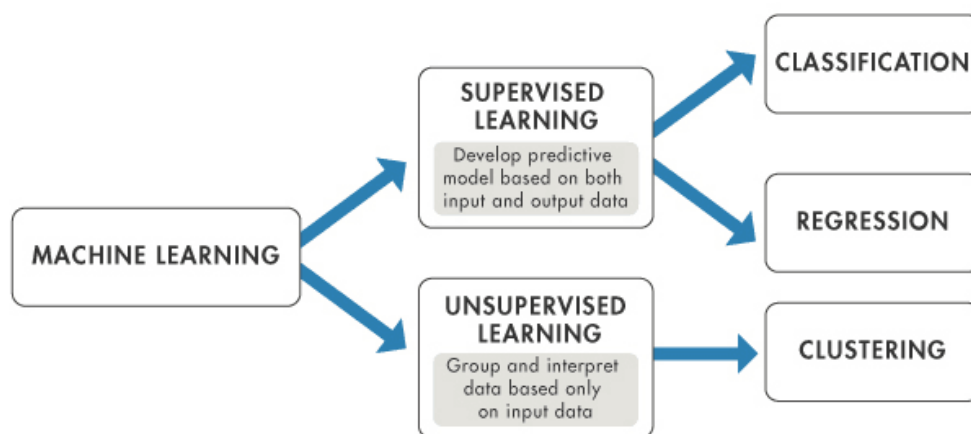
interpretované formě, tedy klasifikaci, nebo k predikci, kdy z trendů minulých vypočítává jejich budoucí průběh. [9] [10]

Hlavní výhodou tohoto způsobu je, že oproti standardnímu přístupu není potřeba poskytovat počítači specifické instrukce a přesně vysvětlit, co hledá a jak by to měl poznat, ale stačí pouze dodat soubor dat obsahující informace a příklady kýženého výsledku a algoritmus strojového učení si už z nich sám vyextrahuje vlastnosti a vzory podle kterých dokáže daný objekt definovat a při aplikaci těchto dat i identifikovat mezi jinými.[9] [10]

Umělá inteligence (UI) je zastřešující název pro vývoj programů snažících se o algoritmičticky implementovaný přístup k inteligentnímu chování. Vývoj je zaměřen na tři hlavní obory: učení, zdůvodnění výsledků a následnou samo-korekci. [9] [10]

Strojové učení je podobor UI soustředící se na programy schopné učení z dat za účelem tvorby predikcí, či klasifikace. Výsledků je možné dosáhnout pomocí učení s učitelem, kdy se program učí ze souborů dat se známými vzory, tedy na vstupu už je modelu řečeno, co daná data představují a jak je má interpretovat. Model si poté vytvoří spojení mezi daty a jejich interpretací podle vlastní logiky, díky čemuž je poté schopen identifikace podobných dat i v instancích, se kterými neměl žádné dočínění. [9] [10]

Další možnost je učení bez učitele, kdy se stroj sám snaží najít určitý systém v datech, které má k dispozici a z těch poté abstrahovat další výsledky například možného budoucího vývoje. [9][10]



Obrázek 9. Učení s učitelem a bez učitele [11]

Pak je zde ještě zpětnovazební učení (také nazývané učení s posilováním), kde se využívá interakce s prostředím za účelem rozhodování. Základním přístupem je koncept odměny a trestu v závislosti na akcích, kterých se algoritmus dopustí, přičemž jeho cílem je

dosáhnout maximální, či nejvyšší možné odměny. Tento přístup je vhodný v aplikacích, kde data přichází sekvenčně a rozhodnutí přijaté v každém kroku dokáže ovlivnit konečný výsledek. [9] [10]

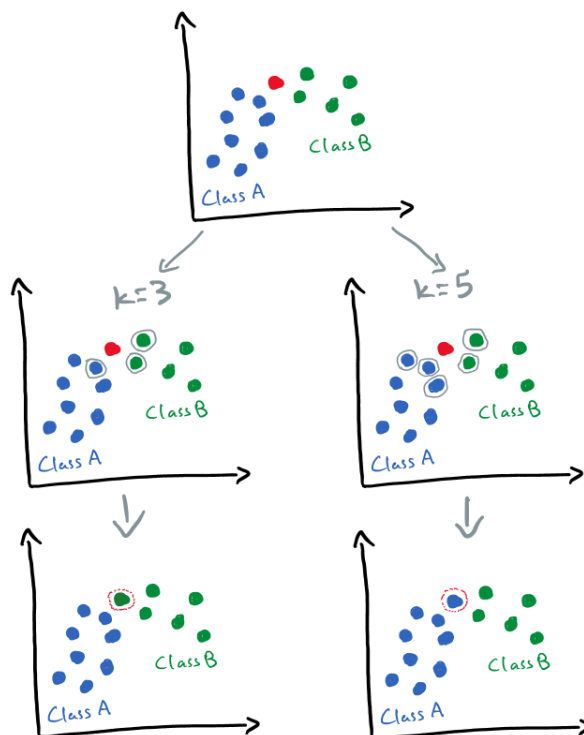
1.3.1 k – Nearest Neighbors

Algoritmus strojového učení k – nearest neighbors (kNN) je neparametrický klasifikátor spadající do skupiny učení s učitelem. Je třeba poznamenat, že v tomto případě je název učení poněkud nadsazený, jelikož kNN si na základě trénovacího datasetu nevytvoří diskriminativní funkci, ale uloží si celý dataset do paměti. Při inicializaci pak model postrádá krok učení a dopouští se nějaké abstrakce až v okamžiku, kdy je žádán provedení klasifikace. Nevýhodou tohoto přístupu je, že s rostoucí velikostí datasetu rostou nároky na paměť a strojový čas potřebný ke klasifikaci. [12][13]

Funguje na konceptu hledání nejbližších shod mezi testovanými daty v pracovní oblasti. Zjednodušeně jde o rozhodování přiřazení bodu do jedné ze dvou či více klasifikačních tříd v závislosti na jeho pozici a vzdálenosti od k jeho nejbližších sousedů. [12] [13]

Řekněme, že máme dvě různé třídy bodů rozmístěných ve 2D prostoru a jeden nezařazený bod, jehož příslušnost k jedné z těchto tříd je třeba klasifikovat. Jak už samotný název algoritmu napovídá, jde o nalezení nejbližšího souseda daného bodu. V případě nejjednodušší verze aplikace klasifikujeme nezařazený bod jako příslušníka té třídy, jejíž jeden člen je nejbližší. [12] [13]

Nicméně tento přístup je problematický v okamžiku, kdy se v blízkosti nachází větší množství příslušníků druhé třídy. V tom případě by tato třída měla vyšší váhu v podobě místní převahy. Z tohoto důvodu se v rámci algoritmu díváme po určitém množství (k) nejbližších sousedů a klasifikovaný bod následně přiřadíme do třídy s větším počtem nalezených instancí. Konečný výsledek klasifikace pak tedy, jak lze vidět na obrázku 10 níže, záleží na velikosti k, tedy na množství sousedů, které bere v potaz. Z toho plyne, že k by mělo být liché, aby nedocházelo ke konfliktu v okamžiku, kdy dojde k nalezení stejných počtů instancí tříd. V případě, že k je sudé a byl nalezen stejný počet členů pro obě třídy je pak možné použít váhy založené například na vzdálenosti, kdy je hodnocený bod přiřazen do třídy, která má více bližších instancí. [12] [13]



Obrázek 10. Vizualizace rozdílného výsledku v závislosti na velikosti k [13]

1.4 Binární obraz

Obsah digitální fotografie je popsán množstvím pixelů z nich každý si mimo jiné nese informaci o vlastní barvě případně intenzitě, pokud jde o obraz v odstínech šedi. Vzhledem k výpočetní náročnosti zpracovávání, prohledávání a dalších operací s barevným obrazem je běžně používáno převedení každého zpracovávaného obrazu na binární soubor, díky čemuž dojde k omezení hodnot, kterých pixel může nabývat na 0 (černá) a 1 (bílá). [14]

Toto zjednodušení má poté za následek menší náročnost na používání paměti, jelikož vzhled každého pixelu je interpretovatelný v rámci jednoho bitu. Dále převedení na monochromatický obraz usnadňuje operace nad jeho částmi, tedy nalezení a vyčištění případné oblasti zájmu. [14]

Je nutné poznamenat, že převod obrazu na binární hodnoty má také za následek ztrátu detailnějších informací a bývá tedy používán v aplikacích, kde samotná silueta čteného objektu dodává dostatek informací potřebných k jeho zpracování. [14]

Převod na monochromatický obraz lze provést pomocí segmentace obrazu, tedy rozdělení obrazu na oblasti se společnými vlastnostmi. V rámci této práce bylo použito prahování, kdy je možné dále pomocí nastavení vstupních parametrů docílit toho, aby se právě požadovaná část obrazu převedla do zpracovatelné podoby siluet, tedy v našem případě, aby

námi požadované hodnoty číselníku byly interpretovány jako část obrazu tvořící popředí, tedy jako bílé objekty. Obecně je tedy zapotřebí, aby objekt, jehož siluetu potřebujeme získat, byl jednoduše oddělitelný od okolí, čehož lze částečně dosáhnout nastavením hodnot parametrů prahování obrazu, nicméně úspěšnost také výrazně záleží na množství vstupních faktorů, které ovlivňují vizuální odstup hledaných předmětů od okolí, jako je osvětlení kontrast a barevný odstup oproti pozadí, odlesky a podobně. [14]

V této práci bychom tedy v ideálním případě po převodu fotografií číselníků disponovali pouze černou plochou, ze které by svítily siluety objektů reprezentující námi hledaná čísla, která bychom hned poté mohli podrobit identifikaci. Nicméně vzhledem ke kvalitativním omezením obrazu pořizovaného použitou kamerou, limitovaným (nicméně stabilním) světelným podmínkám obrázky získané v rámci této práce obsahují poměrně vysoké množství šumu a odlesků. Je také vhodné podotknout, že z důvodu barevné členitosti celých obrazovek měřících přístrojů ani v případě nepoškozeného obrazu v rámci prahování nedojde k dokonalému oddělení čísel od pozadí.

1.5 Knihovna OpenCV

OpenCV (Open Source Computer Vision Library) je knihovna s otevřeným zdrojovým kódem pod licenci BSD obsahující množství algoritmů používaných k operacím v rámci strojového vidění. V rámci této práce byly použity relevantní funkce k úpravě a manipulaci s obrazem. [15]

1.5.1 Odstranění šumu

Mezi algoritmy obsažené v knihovně OpenCV, které byly využity v této práci, patří funkce na odstranění šumu *cv.fastNlMeansDenoising()*, založená na přístupu nazývaném Non-local means. [16]

Základ této metody spočívá v poměrně jednoduchém přístupu, kdy dojde k nahrazení barvy pixelu průměrem barev dalších pixelů, které jsou tomuto podobné. Nicméně vzhledem k tomu, že tyto zmíněné podobné pixely nemají nejmenší důvod se nacházet v blízkosti pixelu, jehož barvu zamýšlíme nahradit, byl přístup hledání rozšířen dále, než na nejbližší okolí. Je tedy vhodné prohledat významnou část obrazu za účelem nalezení pixelů dostatečně shodných s pixelem, na kterém hodláme provést odstranění šumu. Tento přístup se nazývá Non-local Means Denoising. [16] [17]

Parametry:

- src – Vstupní obraz určený k odstranění šumu. 8 nebo 16 bitů
- dst – Výstupní obraz stejné velikosti a typu jako vstup.
- h – Pole parametrů regulující sílu filtru. Čím větší hodnota, tím dokonalejší odstranění šumu, ale za cenu snížení detailu obrazu.
- templateWindowSize – Velikost šablony v pixelech. Slouží pro výpočet vah.
- searchWindowSize – Velikost okna v pixelech. Slouží k výpočtu váženého průměru pro daný pixel. Čím vyšší hodnota, tím délejší je čas potřebný k odstranění šumu z obrazu.
- normType – Typ normy použité k výpočtu vah. NORM_L1 nebo NORM_L2. [16]



Obrázek 11. Příklad výsledku algoritmu NLmeans Originální, zašuměný a znovu vyčištěný obrázek [17]

1.5.2 Prahování obrazu

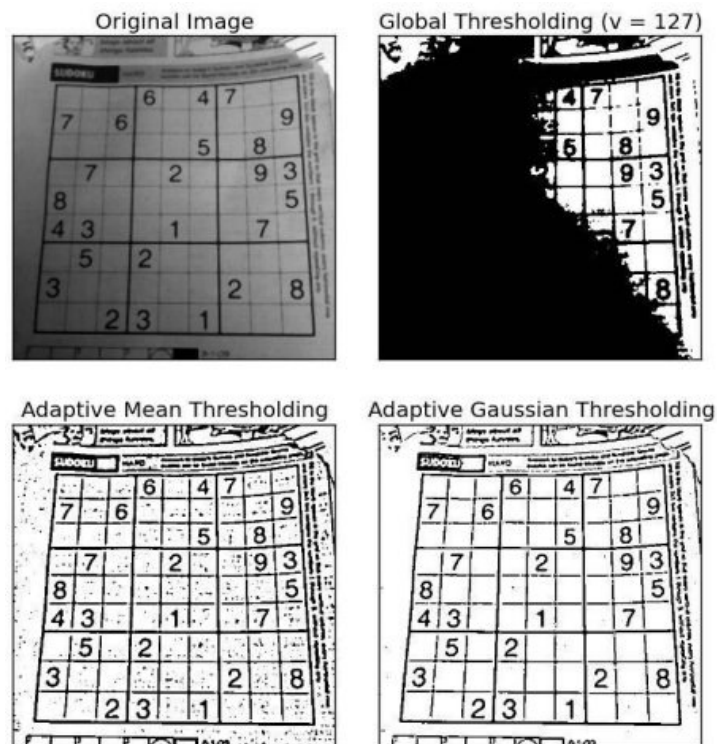
Jak již bylo zmíněno, je jedním ze základních postupů strojového vidění převedení zpracovávané fotografie na monochromatický obraz. To lze docílit nějakou z metod segmentace obrazu, jako je například prahování. [16]

V základní verzi jde o konceptuálně jednoduchý přístup. Pro každý pixel je použita stejná velikost prahu. Pokud je hodnota pixelu menší než velikost prahu, pak je nastaven na hodnotu nula, pokud je větší pak dojde k nastavení na maximální hodnotu. Tento jednoduchý či globální přístup bolestně selhává v okamžiku, kdy obraz nebyl pořízen v homogenních světelných podmínkách a každá jeho část je nasvícena jiným způsobem, výsledkem čehož může být binární obraz nevhodný pro další zpracování. [16]

Z tohoto důvodu bylo použito adaptivní prahování implementované ve funkci *cv.adaptiveThreshold()*. V tomto případě algoritmus sám určuje hodnotu prahu pro pixel v závislosti na jeho blízkém okolí. Tímto způsobem získáme různé hodnoty prahu pro různé oblasti obrazu v takovém stavu, aby se celý obraz jevil jako pořízený za stejných podmínek po své celé ploše [16]

Parametry:

- *src* – Vstupní obraz, 8-bitový jednobarevný.
- *dst* – Výstupní obraz stejné velikosti a typu jako vstup.
- *maxValue* – Nenulová hodnota přiřazovaná pixelům splňujícím podmínky.
- *adaptiveMethod* – Konkrétní algoritmus pro adaptivní tresholding.
- *thresholdType* – Typ prahovací logiky.
- *blockSize* – Velikost pixelové oblasti použité k výpočtu prahové hodnoty pixelu. Liché číslo.
- *C* – Konstanta odečtená od váženého průměru. [16]



Obrázek 12. Rozdíl mezi globálním a adaptivním prahováním [18]

1.5.3 Kontury

Při práci s obrazem v rámci strojového vidění je zapotřebí disponovat funkční schopností k nalezení a separování zájmových objektů. Jednou z možností je použití kontur. Kontura je metoda segmentace obrazu, která může být jednoduše popsána jako křivka propojující body o stejné barvě nebo intenzitě, které definují hranici objektu, čímž ho oddělují od pozadí. [16]

Vhodný výchozí formát pro identifikaci a práci s konturami je binární obraz z důvodu snadné separovatelnosti objektů od pozadí na základě jasně rozdílného stavu pixelů (černá, bílá). Kvalita získaných kontur přímo závisí na vlastnostech obrazu, mimo jiné na rozlišení, přítomného šumu, osvětlení scény a podobně. Dále taky záleží na správně provedeném převodu na monochromatický obraz, tak aby hranice požadovaných objektů nesplývaly s dalšími. [16]

Knihovna OpenCV disponuje funkcí *cv.findContours()* sloužící k nalezení kontur a jejich prezentaci ve formě dvourozměrného pole souřadnic jednotlivých hraničních bodů v prostoru prohledávaného obrazu. [16]

Parametry:

- *image* – Zdroj ve formě 8-bitového jednobarevného obrazu. Nenulové pixely jsou interpretovány jako 1, nulové zůstávají stejné, zpracováváný obraz je tedy považován za binární.
- *contours* – Detekované kontury. Každá kontura je uložena jako dvourozměrné pole.
- *hierarchy* – Volitelný výstupní vektor obsahující informaci ohledně topologie obrazu. Počet členů odpovídá množství nalezených kontur.
- *mode* – Způsob vyhledávání kontur definovaný členem výčtu.
- *method* – Způsob aproximace kontur v obraze.
- *Offset* – volitelný posun, o který jsou posunuty jednotlivé body tvořící konturu. [16]

II. PRAKTICKÁ ČÁST

2 HARDWARE

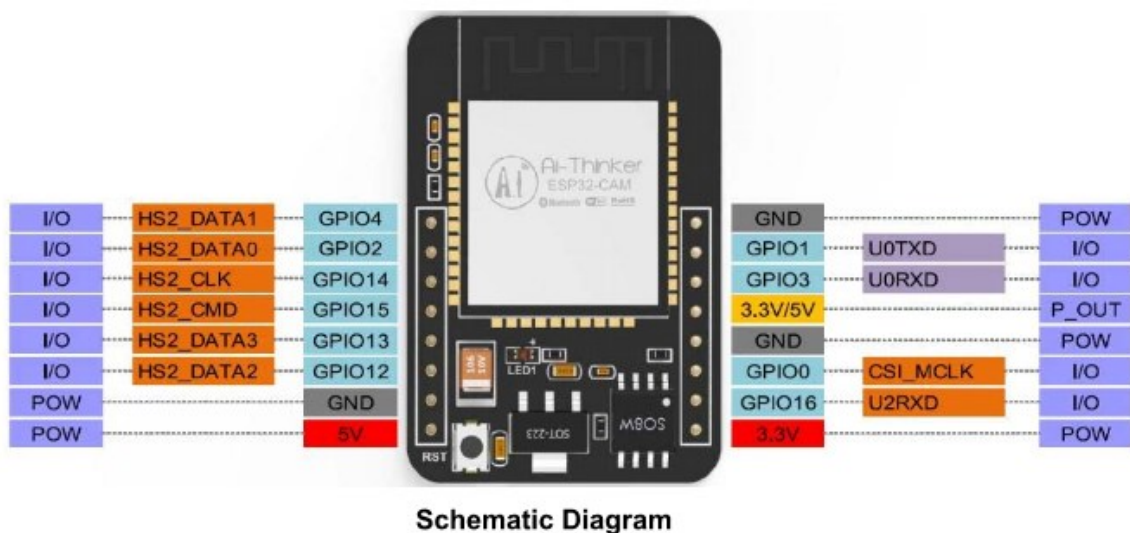
Použitý HW pro zpracovávanou aplikaci byl volen v závislosti na požadavcích minimálního zabraného prostoru a kompromisu ohledně cenové náročnosti a výkonu. Nároky na periferní zařízení zahrnovaly schopnost pořízení obrazu a jeho odeslání do centrální jednotky nejlépe pomocí bezdrátového způsobu komunikace, při nízké pořizovací ceně jednotky.

Požadavek na centrální jednotku byla kompatibilita s módem komunikace periférií a dostatečným výkonem na splnění požadované úlohy.

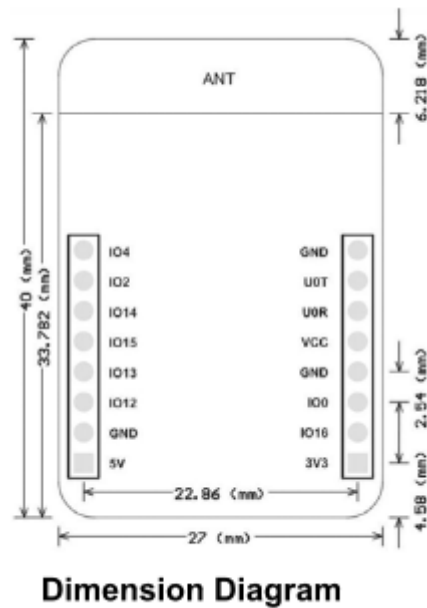
2.1 ESP-32 CAM

ESP-32 CAM je relativně výkonná nízkonákladová vývojová deska s minimálními rozměry, založená na ESP32 platformě, schopná jak samostatné existence, tak fungování v rámci sítě IoT, což jí činí vhodnou k počátečnímu prototypování, testování a funkční implementaci projektu vyžadujícího bezdrátové zařízení. [19]

Výpočetní jádro samotné desky se skládá ze dvou 32-bitových LX6 procesorů se sedmistupňovou architekturou řetězení instrukcí. Dále obsahuje kompatibilitu s rozhraními UART, SPI, I2C, PWM, dedikované rozhraní pro kameru OV2640 a integrovaný WiFi modul kompatibilní se standardy 802.11b/g/n a Bluetooth 4.2. [19]



Obrázek 13. Schématický diagram ESP-32 CAM [20]



Obrázek 14. Rozměrový diagram ESP-32 CAM [21]

2.2 OV2640

Základní součást výše zmíněné desky a důležitý prostředek pro tuto aplikaci je obrazový senzor OV2640, což je nízkonapěťová CMOS jednočipová UXGA kamera a obrazový procesor v kompaktním balení, komunikující po SCCB rozhraní. Ačkoliv zde bez výhrad platí poučka, že kvalita výsledného obrazu odpovídá ceně, tak ve výsledku lze říci, že pro zamýšlenou aplikaci je dostačující a to i při ztížených světelných podmínkách. [22]

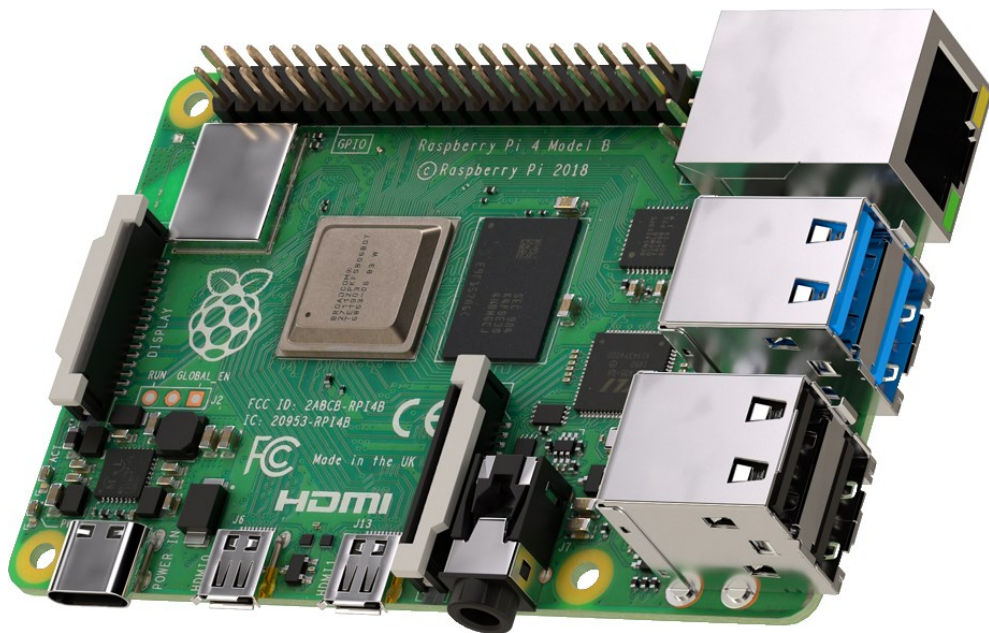


Obrázek 15. Kamera OV2640 [22]

2.3 Raspberry Pi 4 model B

Jako centrální výpočetní jednotka celého systému bylo zvoleno Raspberry Pi B 4GB. Je vybaveno procesorem Broadcom BCM2711 což je čtyř-jádrový Cortex-A72 procesor s instrukční sadou ARM v8 založený na 28 nm technologii, přičemž konfiguraci velikosti RAM paměti je možno vybírat z vícero možností (1, 2, 4 a 8 GB). Vybraný model disponuje 4 GB paměti, díky čemuž v kombinaci s výše zmíněným procesorem dosahuje výpočetního výkonu schopnému sloužit jako plnohodnotná náhrada osobního počítače. Přinejmenším na poli vstupních 32 bitových systému. [23]

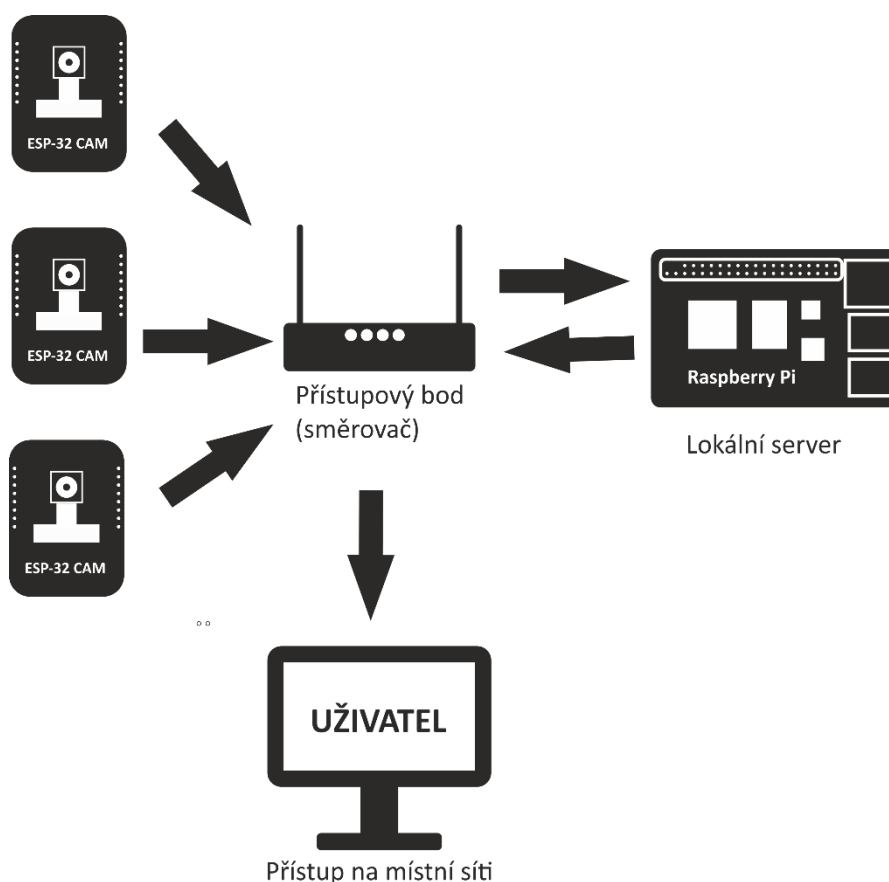
Mezi další výbavu patří dva mikro HDMI konektory a čtyři USB porty, díky čemuž je možné k Raspberry připojit obrazové a ovládací periferie. Vzhledem k dnešnímu důrazu na konektivitu, a zároveň komunikační potřeby projektu zpracovávaného v této práci, zařízení obsahuje Ethernetový port, WiFi anténu na Bluetooth 5.0. [23]



Obrázek 16. Raspberry Pi 4B [24]

3 INTEGRACE OCR

Integrace celého systému rozpoznávání znaků z číselníků měřících přístrojů spočívá jednak v implementaci dílčích částí řešení – tedy pořizování snímků obrazovek, zpracování těchto snímků na čitelný formát, extrakce zobrazených čísel a prezentace výsledných dat uživateli. Další část spočívá ve vzájemné integraci těchto dílčích kroků do jednoho fungujícího celku včetně návrhu umístění, fixace a nastavení periferií, které provádějí čtení obrazovek na konkrétních zařízeních.



Obrázek 17. Schéma sítě systému

3.1 Čtení obrazovek

Jako vzdálená periferie využitá ke snímání obrazovky byla použita výše zmíněná deska ESP-32 CAM s kamerou OV2640. Mimo implementace sběru dat (tedy pořizování fotografií měřícího přístroje) disponuje vlastním asynchronním webserverem umožňujícím uživatelskou úpravu základních parametrů kamery ukládaných do trvalé paměti, což slouží k optimalizaci kvality výsledného obrazu v závislosti na místních podmínkách tak, aby byla pořízená fotografie v co nejvhodnějším dosažitelném stavu pro další zpracování.

Následně dojde k odeslání pořízené fotografie na lokální server běžícím na Raspberry Pi po dotazovací metodě HTTP POST, kde dojde k identifikaci čísel jejich uložení a prezentaci na vlastním webu místního serveru. Server neuchovává všechny pořízené fotografie, ale vždy pouze jednu pro každý měřič sejmoutou v okamžiku posledního vyhodnocovacího cyklu.

Tento postup se opakuje v předem nastavených intervalech, mezi kterými celá deska ESP-32 CAM přechází do režimu spánku.

3.2 Zpracování obrazu

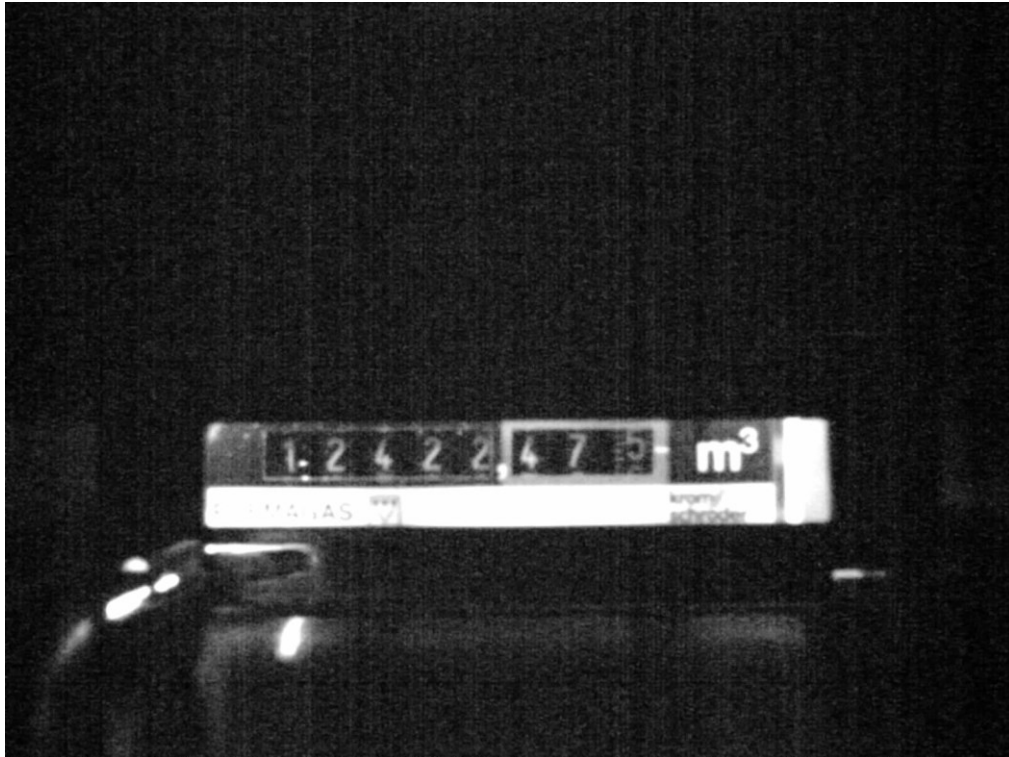
Extrakce čísel z pořízených obrazovek probíhá na Raspberry Pi pomocí Python skriptu. Po obdržení fotografie z kamery je tato přebrána cyklicky běžícím skriptem, v rámci kterého dojde k přečtení číselníku a uložení výsledné hodnoty do csv souboru ve formátu čas, typ zařízení a přečtená hodnota.

Zpracování obrazového souboru probíhá v několika etapách:

1. Převedení obdrženého souboru na binární obraz
2. Vyčištění šumu a odstranění nadbytečných objektů
3. Nalezení samotného číselníku a opětovné odmazávání irelevantních objektů
4. Identifikace objektů představujících hledaná čísla a jejich případnou úpravu za účelem zvýšení čitelnosti.
5. Rozpoznání samotných čísel.

3.2.1 Úprava fotografie a nalezení číslic

Výchozí obraz použitý ke zpracování je nijak neupravovaná fotografie pořízená výše zmíněnou kamerou. Vzhledem k tomu, že nastavení kamery a konstrukce jejího prostředí byly podřízeny získání co nejlépe osvětleného číselníku v displeji měřicího přístroje za maximální možné minimalizace jakýchkoliv odlesků v oblasti samotných čísel a předejití oslnění kamery přidavným přisvětlením, dochází v rámci kvality celé fotografie k výraznému nedostatku. V tomto případě jde o obrazový šum, viditelný na obrázku níže, který je v tomto případě zapříčiněn nastavením maximální světelné citlivosti kamery a tendencí LED osvětlení generování vizuálního a elektromagnetického šumu.



Obrázek 18. Výchozí obraz ke zpracování

Z tohoto důvodu je krajně nevhodné provést segmentaci obrazu hned po jeho získání a převedení do odstínů šedi, jelikož použitý přístup adaptivního prahování nedokáže sám o sobě šum kompenzovat, ale naopak ho zvýrazní oddělením od pozadí spolu s hledanými objekty, což by eventuálně mohlo přinejmenším komplikovat následné kroky.

K prvotnímu odstranění šumu byla použita funkce *cv.fastNlMeansDenoising()*, která sice výsledný obraz plně nevyčistí, ale zpracuje veškerý optický šum do stavu, kdy je velmi jednoduché ho po provedení prahování odstranit. Ačkoliv knihovna OpenCV umožňuje odstranění šumu i z barevné fotografie, v práci je z důvodu šetření strojového času použita operace nad obrazem v odstínech šedi.

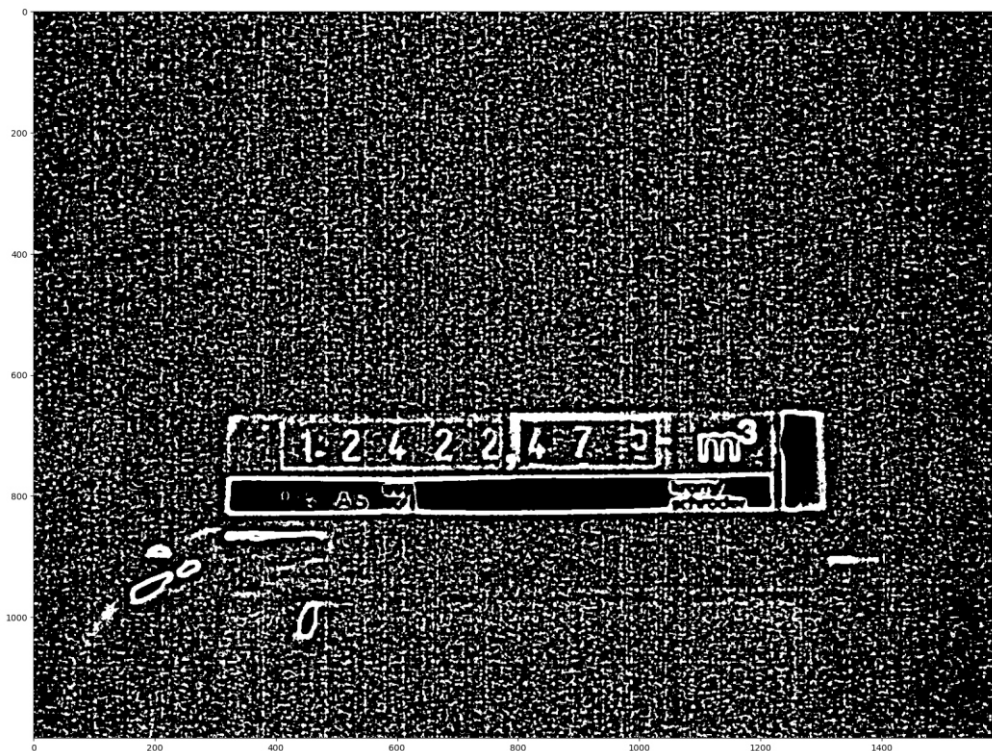
Nutno poznamenat, že výsledek odstranění šumu v barevném obraze a jeho následné převedení na odstíny šedi je vesměs totožný, nicméně za cenu vyšší časové náročnosti, která pro nás nemá žádný význam, jelikož s barvami stejně nepracujeme. Cenou této úpravy je mírné rozostření obrazu a ztráta některých detailů, která nicméně v našem případě není příliš důležitá.



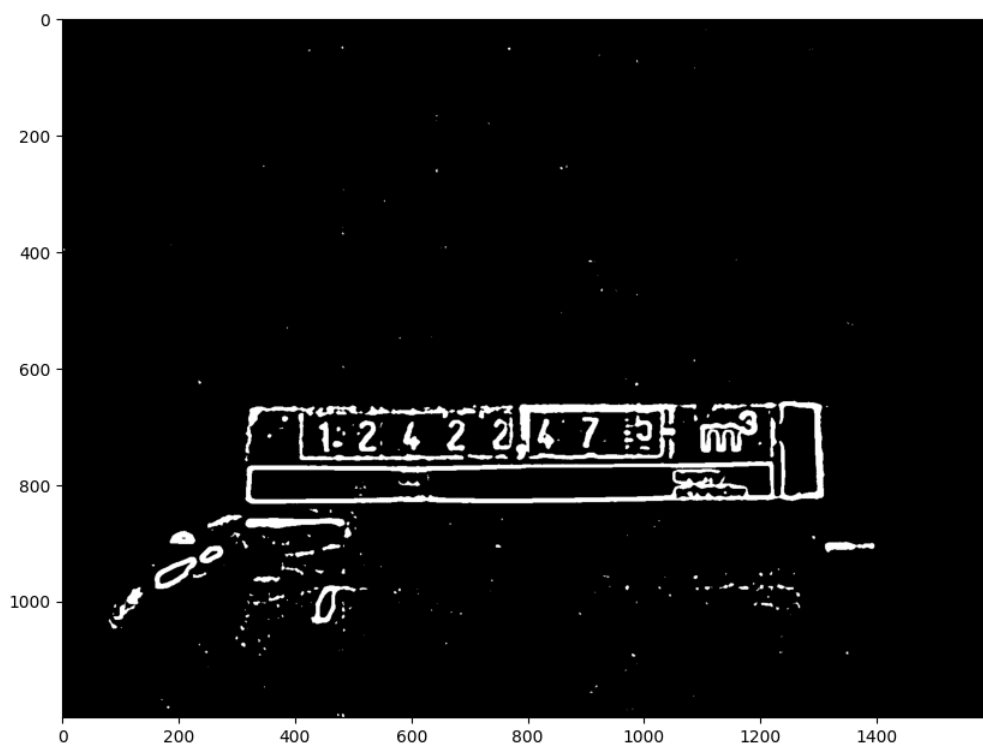
Obrázek 19. Výsledek odstranění šumu

Další krok je využití funkce *cv.adaptiveTreshold()* s cílem převést náš obraz v odstínech šedi na monochromatický. Nastavitelné parametry *boxLimit* a *C* byly v případě zpracování číselníku plynoměru nastaveny 15 a -5, tyto hodnoty byly určeny a ověřeny experimentálně jako nejvhodnější pro danou aplikaci. Vzhledem k nastavení a udržení konzistentních podmínek pořizování fotografií je po instalaci a nasazení čtecího zařízení zajištěna stabilita a opakovatelnost výsledku bez nutnosti dalšího zásahu zvenčí. Tyto hodnoty se samozřejmě lišily pro každé měřicí zařízení s tím, že v případě elektroměru byly nastaveny na 66, 14 a u vodoměru na 15 a 7.

Na obrázcích 20 a 21 níže se pak nachází fotografie číselníku plynoměru, převedená na binární obraz bez jakýchkoliv úprav a pod ním obrázek po odstranění šumu, nicméně stále před konečným vyčištěním.



Obrázek 20. Segmentace neupraveného obrazu



Obrázek 21. Segmentace obrazu po odstranění šumu

K veškerým shlukům bílých pixelů (neboli kontur), lze v rámci zpracovávaného obrazu přistoupit funkcí `cv.findContours()`. Tato funkce vrátí pole obsahující množství $2 \times X$ matic z nichž každá obsahuje hraniční body každé jedné kontury, přičemž X je v tomto případě počet těchto bodů. Z tohoto je poté možné získat další informace jako rozměry, pozici a plochu kontury v obraze. To je možné využít k dalšímu vyčištění obrazu od redundantních prvků.

Jak lze také vidět, v rámci námi zpracovávaného binárního obrazu se nachází hraniční kontury číselného displeje, které lze využít k přesnější lokalizaci číselníku v obraze.



Obrázek 22. Konečná lokace číselníku



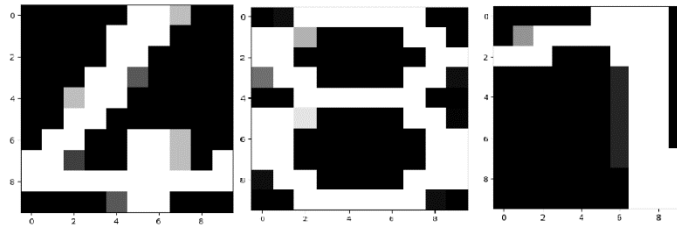
Obrázek 23. Vyčištěný analogový číselník s nalezenými objekty číslic

3.2.2 Rozpoznání číslic

V okamžiku získání jednotlivých číslic v prezentované formě přichází na řadu samotná identifikace číslic. V této práci jsou použity dva přístupy.

Pro plynoměr a vodoměr s analogovými číslicemi byl použit identifikační přístup pomocí klasifikačního algoritmu strojového učení k-Nearest Neighbors. V rámci tohoto přístupu byla vytvořena databáze číselných glyfů z fotografií analogových číselníků různých relevantních měřicích zařízení. Následně byl každý jeden glyf převeden na binární obraz, extrahován od pozadí, zmenšen na uniformní velikost 10×10 pixelů a ve formátu vektoru, ve kterém je každý jeden pixel reprezentován svojí intenzitou, uložen do souboru hodnot `glyphSamples.data` o počtu řádků odpovídajících množství souborů v trénovací sadě.

Zrcadlově byl vytvořen soubor s odpověďmi `glyphResponses.data`, kde se pro každou matici reprezentující číselný objekt nachází odpovídající číselná interpretace. Tato data pak byla použita k naučení klasifikačního algoritmu.



Obrázek 24. Glyfy 4,8 a 1 ve formátu 10x10 pixelů

Podobným způsobem funguje i samotné rozpoznání čísel, kdy se výsledného obrazu z objektů představujících číslice vezme vždy jedna kontura, ta je převedena na velikost 10x10 pixelů a poté prezentována modelu k-Nearest Neighbors, za účelem nalezení třídy (čísla), která je této kontuře nejbližší. Počet porovnávaných sousedů k byl nastaven na jednoho, s tím že experimenty s většími hodnotami k nepřinášely lepší výsledky identifikace.

Rozpoznání číslic elektroměru je vzhledem k použití digitálního číselníku pro změnu založeno na kontrole momentálně rozsvícených segmentů každého čísla. Každé číslo digitálního číselníku se skládá ze sedmi dílů a je jednoznačně identifikovatelné na základě toho kolik a kterých dílů je v daném momentě rozsvíceno.



Obrázek 25. Vyčištěný digitální číselník s nalezenými objekty číslic

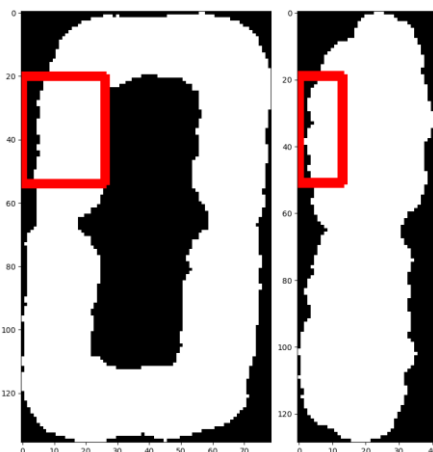
Rozpoznávání čísel bylo poté implementováno způsobem, kdy se na předem definovaných pozicích pro každý jeden segment zkontroluje jeho přítomnost, a pokud je v oblasti určené pro daný segment zhruba 60 procent plochy bílá, segment se považuje za přítomný a získává příznak 1.

```
segmentPos = []#vytvoreni pole s identifikovanymi polohami segmentu pro dane cislo
segmentPos.append([0, h*0.16, w*0.33, w*0.66, 0])#horni segment
segmentPos.append([h*0.15, h*0.40, 0, w*0.33, 0])#levy horni segment
segmentPos.append([h*0.15, h*0.40, w*0.66, w, 0])#pravý horni segment
segmentPos.append([h*0.15, h*0.40, w*0.33, w*0.66, 0])#horni prostredni segment
segmentPos.append([h*0.45, h*0.55, w*0.33, w*0.66, 0])#prostredni segment
segmentPos.append([h*0.60, h*0.84, w*0.33, w*0.66, 0])#spodni prostredni segment
segmentPos.append([h*0.60, h*0.84, 0, w*0.33, 0])#levy spodni segment
segmentPos.append([h*0.60, h*0.84, w*0.66, w, 0])#pravý spodni segment
segmentPos.append([h*0.84, h, w*0.33, w*0.66, 0])#spodni segment
```

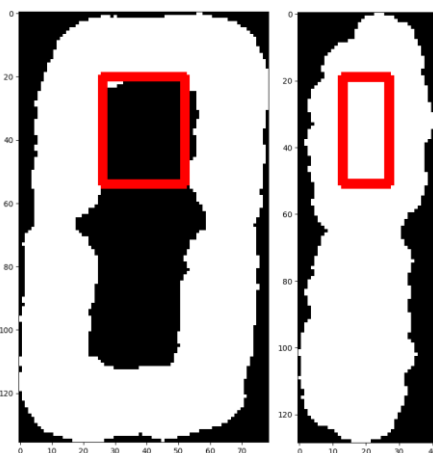
Obrázek 26. Definice pozice segmentů v kontuře číslice

V případě, že je vyhodnocen jako chybějící, pak získává příznak 0. Po projití a vyhodnocení všech sektorů objektu dojde k porovnání se slovníkem, který definuje, které díly musí být aktivní pro které číslo a podle shody je daná číslice identifikována. V případě, že není nalezena shoda, je místo číslice dosazeno X.

V rámci testování se ukázalo, že se zvoleným způsobem extrakce čísel a následným vyhledáváním jednotlivých segmentů dochází ke kolizím při identifikaci číslice 1. Při kontrole segmentů, jejichž pozice a velikost je definovaná relativně k velikosti prohledávané kontury, se předpokládá, že každé jedno číslo bude extrahováno v šíři, které odpovídá plné délce příčného dílu. V případě číslic 2-9 to není problém, jelikož každá z nich obsahuje alespoň jeden příčný díl a dosahují tedy potřebné šířky. Číslo jedna nicméně zůstává rovno pouze v šíři vertikálního dílu, což mělo za následek neodpovídající hodnocení stavu segmentů, kvůli čemuž buď docházelo k selhání (nedefinovaná kombinace aktivních a neaktivních segmentů), či byla rozpoznána jako úplně jiné číslo (nejčastěji šlo o 6 a 8).



Obrázek 27. Porovnání pozice pravého horního segmentu



Obrázek 28. Porovnání pozice horní dutiny

Z tohoto důvodu byly zavedeny další dva segmenty vyskytující se v pozicích, kde mají všechna čísla prázdné místo (například uprostřed smyček u čísla 8). Díky této úpravě bylo dosaženo dostatečného rozdílu mezi číslicí jedna a ostatními čísly, majícího za výsledek jednoznačnou identifikaci i v rámci širkového rozdílu.

Rozpoznání číslice jedna také naráželo na problémy se změnou šířky testovacích oblastí a to především v pozici levého horního a pravého dolního segmentu. V závislosti na kvalitě, nebo spíše velikosti hraničních rozměrů kontury extrahovaného objektu v kombinaci spolu s nakloněním číslice docházelo na různých fotografiích k rozdílným výsledkům vyhodnocení přítomnosti výše zmíněných segmentů v testovací oblasti, kdy občas byl segment vyhodnocen jako přítomen aby v dalším případě zůstal uveden jako neexistující.

Výsledné řešení spočívalo v zadání veškerých možných kombinací, kterých může správně extrahovaný objekt číslice jedna nabývat, do slovníku definujícího rozložení segmentů pro každé číslo zvlášť, což bylo možné především proto, že tyto nové kombinace nekolidují s žádným z ostatních čísel.

Jak lze vidět na obrázku níže, zatímco pro čísla 2-9 se ve slovníku nachází vždy jedna možnost, tak pro konzistentní identifikaci čísla 1 musely být vloženy 4 kombinace.

```
DIGITS_LOOKUP = {
    (1, 1, 1, 0, 0, 0, 1, 1, 1): 0,
    (1, 0, 1, 1, 1, 1, 1, 0, 1): 1,
    (1, 1, 1, 1, 1, 1, 1, 0, 1): 1,
    (1, 0, 1, 1, 1, 1, 1, 1, 1): 1,
    (1, 1, 1, 1, 1, 1, 1, 1, 1): 1,
    (1, 0, 1, 0, 1, 0, 1, 0, 1): 2,
    (1, 0, 1, 0, 1, 0, 0, 1, 1): 3,
    (0, 1, 1, 0, 1, 0, 0, 1, 0): 4,
    (1, 1, 0, 0, 1, 0, 0, 1, 1): 5,
    (1, 1, 0, 0, 1, 0, 1, 1, 1): 6,
    (1, 0, 1, 0, 0, 0, 0, 1, 0): 7,
    (1, 1, 1, 0, 1, 0, 1, 1, 1): 8,
    (1, 1, 1, 0, 1, 0, 0, 1, 1): 9,
}
```

Obrázek 29. Slovník definující číslo na základě přítomnosti segmentů

3.2.3 Shromáždění dat

V rámci Python skriptu, starajícího se o zpracování obrazu dále dochází k vytvoření a naplnění jednoduchého csv souboru hodnotami sejmутými z měřicích přístrojů. Tento soubor

obsahuje data v jednoduchém formátu: časové razítko, informaci o zařízení (gas, water, electricity), sejmutou hodnotu. Sejmutá hodnota může obsahovat jednak přečtené číslo z číselníku měřícího zařízení, popřípadě informaci o selhání identifikace.

Skript také kontroluje dobu posledního čtení jednotlivých přístrojů, a pokud během určité doby neobdrží od fotografii od periferního zařízení uloží do kolonky sejmuté hodnoty informaci, že neobdržel žádná data.

Výsledný soubor data.csv je pak možné stáhnout ze stránky lokálního serveru přístupné na jeho adrese. Na dané stránce lze také zhlédnout čísla z posledního čtení, spolu s pořízenými obrazy číselníků pro kontrolu kvality rozpoznání číslic.

3.3 Fixace zařízení

Součástí zadání práce byl i návrh a realizace fixace čtecích periferií na daných zařízeních pomocí 3D tisku. Filozofie návrhu se řídila níže zmíněnými podmínkami, s tím že v případě třetí podmínky došlo k určitým kompromisům, jelikož snadná rozebíratelnost v některých případech kolidovala s konečným zvoleným způsobem osvětlení.

- Zajištění konzistentních podmínek pořizování snímku daného měřiče.
- Minimalizace odlesku a oslnění displeje a kamery
- Snadné odnímání fixačních přípravků a jejich opakovatelná použitelnost.

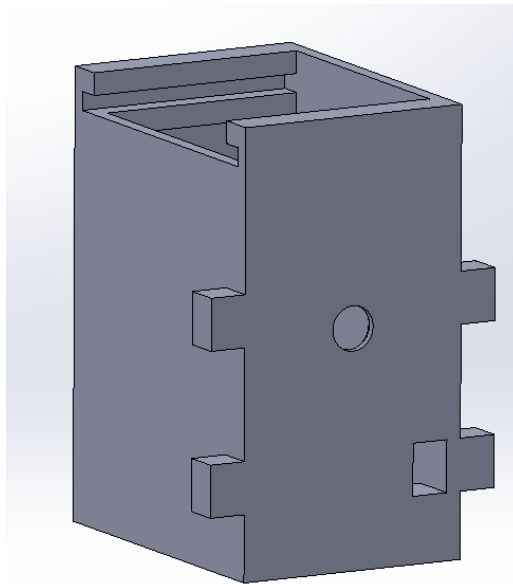
Návrh probíhal v programu SolidWorks. Největší potíže se vyskytly při návrhu rámu na plynoměr. Oproti tomu s elektroměrem byly problémy nejmenší, alespoň v rámci instalačních případů, ke kterým jsem měl přístup.

Významným souhrnným problémem při návrhu a zkoušení komponent se ukázalo správné stanovení míry tolerancí tak, aby jednotlivé části do sebe zapadaly s minimální vůlí. 3D tiskárna jaksí není CNC soustruh a při tisku se dopouští o něco vyšších nepřesností, nebo přesněji řečeno, její preciznost má poněkud variabilní kvalitu. Dalším faktorem jsou nekonzistentní vlastnosti materiálů od různých výrobců, které, nehledě na ekvivalentní typ a stejné deklarované vlastnosti, se pak při výrazně projevují při samotném tisku a následně i při kompletaci vyrobených sestav.

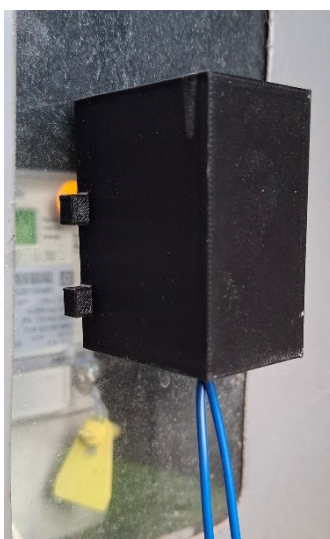
3.3.1 Elektroměr

V případě elektroměru stačí disponovat boxem obsahujícím kameru, měnič a přívod pro externí zdroj napájení. Vzhledem k existujícímu podsvícení digitálního displeje není potřeba řešit přísvit pro pořizování obrazu a stabilní světelné podmínky jsou zajištěny uzavřením

rozvaděče. Kameru stačí připevnit na plexisklový kryt plomby před číselníkem, popřípadě na dveře rozvaděčové skříně v závislosti na přání uživatele. V rámci testování v této práci byl použit typ upevnění na plexisklo pomocí oboustrannné lepicí pásky.



Obrázek 30. Úložný box pro ESP-32 CAM

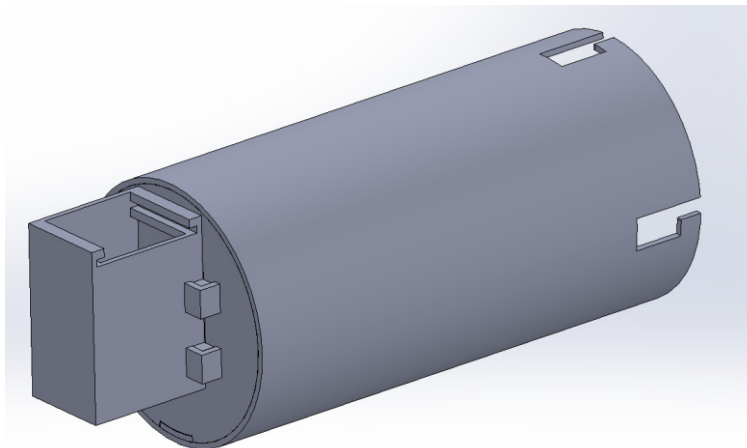


Obrázek 31. Umístění kamery na elektroměru

3.3.2 Vodoměr

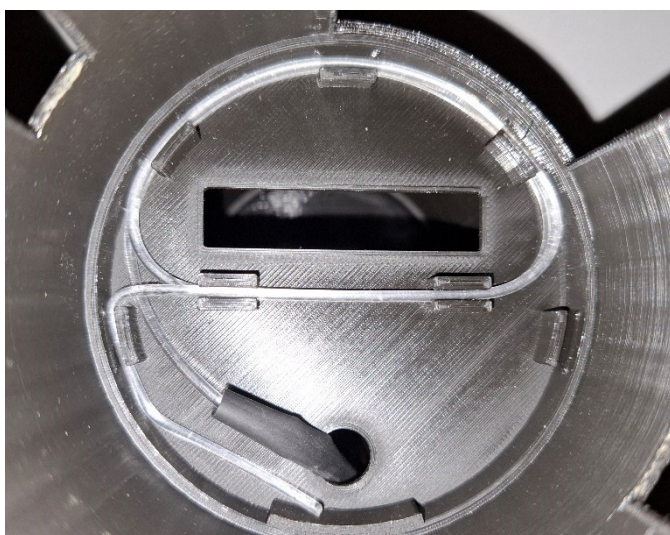
Pro vodoměr byl vytvořen světelný tunel, sloužící zároveň jako nástavec, blokující většinu nežádoucích zdrojů světla přicházejících z exteriéru, nasaditelný přímo na pouzdro měřiče

s výřezy pro zaháknutí za výstupky po stranách objímky vodoměru. Tím je mimo jiné zajištěné, že tubus bude držet na vodoměru v jakékoliv poloze, mimo jiné i na vodoměru s číselníkem otočeným směrem dolů.



Obrázek 32. Nástavec na vodoměr

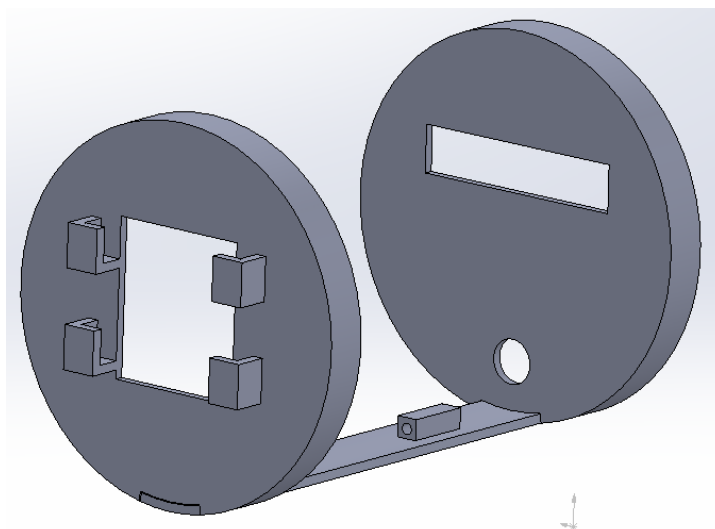
Osvětlení bylo řešeno aplikací světlovodného vlákna vedeného od LED osvětlení přítomného na ESP-32 směrem k číselníku a přibližně po jeho obvodu, čímž se předešlo většině nežádoucích odlesků a zároveň se docílilo rovnoměrnějšího osvětlení, než by bylo k dispozici za použití jednoho bodového zdroje. Pro umístění a přichycení vlákna byla dovnitř tubusu doplněna vložka, která rovněž slouží jako stínidlo, aby obrazový senzor nebyl oslňován přímým pohledem na vlákno.



Obrázek 33. Detail umístění světelného vlákna ve vložce v pohledu od číselníku

Vzhledem k vzájemně nepředvídatelné pozici číselníku a výstupků na objímce vodoměru bylo rozhodnuto, že umístění kamery na tubusu bude pohyblivé, respektive bude možno kamerou rotovat okolo pomyslné osy procházející středem tubusu, tak aby bylo kameru vždy

možno umístit kolmo k číselníku nehledě na pozici tunelu. Také byla vytvořena spojnice mezi světelnou vložkou a poklopem na umístění kamery, aby byla zaručena jejich korektní vzájemná orientace. Celá tato sestava je samozřejmě rozdělená na jednotlivé části tak, aby bylo možné odstranit kameru z tubusu bez nutnosti manipulace s osvětlovací vložkou.



Obrázek 34. Sestava vložky a přichytného bodu kamery

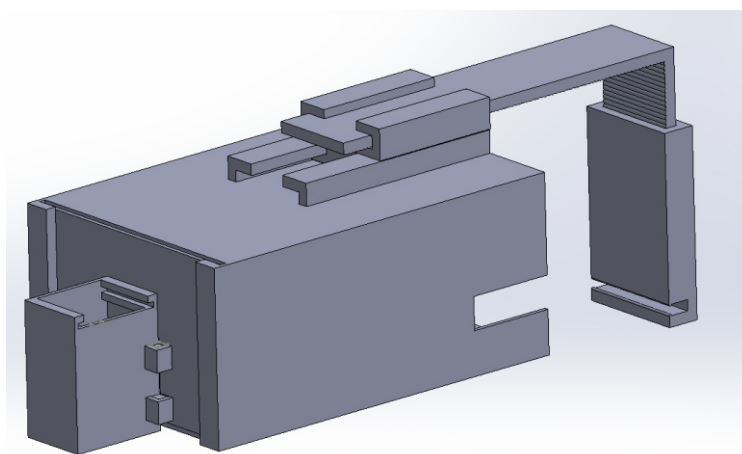


Obrázek 35. Umístění kamery na vodoměru

3.3.3 Plynoměr

U plynoměru se vyskytly problémy hned od počátku. Není žádoucí provádět jakékoliv ukotvení na samotný kryt číselníku, jelikož tento je konstruován záměrně jako destrukční. Zároveň vzorek pokusných plynoměrů měl výraznou deviaci tvaru číselníků a jejich rozmístění.

Výsledkem těchto vstupních omezení je poněkud překombinovaná, nicméně funkční sestava, jejíž způsob úchyty spočívá v zaháknutí na zadní straně plynoměru za jeho obvodový límec. Z dílu sloužícího jako zadní příchyt pak vede profil ve tvaru L, který drží světelný tunel přitisknutý k číselníku plynoměru.



Obrázek 36. Sestava pro snímání plynoměru

Stejně jako v případě vodoměru, bylo i zde k osvětlení číselníku použito světelné vlákno vedoucí od vlastního LED světla kamery do vložky uvnitř tunelu. K obrázku níže je také nutno konstatovat, že mezi vložku s osvětlením a kameru bylo umístěno stínítko, protože v opačném případě měla vlákna tendenci kameru oslňovat, čímž se číselník ocital ve stínu a tedy nečitelný.



Obrázek 37. Detail umístění světelného vlákna ve vložce v pohledu od kamery



Obrázek 38. Umístění kamery na plynoměru

3.4 Napájení

Dalším problémem týkajícím se spolehlivé a korektní funkce periferií bylo kromě způsobu jejich umístění a přichycení na daná místa i napájení. ESP-32 CAM může být napájena z jednoho ze dvou vstupů, které jsou na zařízení k dispozici a to stejnosměrným napětím o velikostech 3,3V nebo 5V, přičemž druhý z těchto vstupů je veden přes napěťový regulátor AMS1117-3.3 schopný ustát napětí až 15V, proud 1,5A a vyžadující minimální stabilní napětí 4,5V. [25] [26]

Z těchto důvodů byl pro napájení vybrán vstup 5V, jelikož mimo jiné zajišťuje funkci kamery i pokud nějakým způsobem dojde k přivedení vyššího než požadovaného napětí na vstup.

V ideálním případě by jako zdroj napájení pro kamerovou periferii měl sloužit napájecí adaptér s požadovaným výstupem do 6V, nicméně vzhledem k limitacím instalace, kdy nemusí být v dosažitelné vzdálenosti k dispozici vhodný zdroj elektrické energie (třeba zásuvka), z nějakého důvodu není vhodné, či možné přivést střídavých 230V pro adaptér a tažení kabelů pro 5 až 6 Voltů napětí na delší vzdálenosti by generovalo neúměrné ztráty, bylo rozhodnuto do systému implementovat akumulátor.

Jak již bylo dříve řečeno, celá periferie na desce ESP-32 CAM bude během nasazení uvedena do režimu hlubokého spánku, ze kterého se bude probouzet v uživatelem zvolených intervalech, aby provedla sejmутí číselníku přiděleného měřicího zařízení. Bylo změřeno, že

spotřeba energie v režimu spánku nepřesahuje 5mA, nicméně v okamžiku probuzení se spotřeba může skokem vyšplhat až na 500mA a to z důvodu náročnosti samotného obrazového senzoru a WiFi antény.

Jedním z požadavků na použitou baterii je její schopnost napájet kamerový systém po nějaký netriviální časový úsek, aby například nemusela být měněna, či dobíjena každý týden. Bylo by též vhodné, aby se baterie nacházela v přijatelné cenové relaci, přičemž akumulátor schopný dobíjení je preferovaná možnost z důvodu opakovatelné použitelnosti. Z těchto důvodů ve výsledku padla volba na dvanácti voltový olověný akumulátor.

Otázka zvolené kapacity záleží čistě na uživateli, jak velkou (a především těžkou) baterii bude ochoten a schopen umístit v blízkosti čtecích zařízení, nicméně v rámci práce byla zvolena baterie i kapacitě 12Ah.

Jak již bylo zmíněno výše, spotřeba periferního zařízení je 5mA v režimu spánku a 500mA při plnění funkce, přičemž samotný cyklus probuzení, pořízení snímku, jeho odeslání a následného opětovného přechodu do režimu spánku vychází průměrně na 10 vteřin. V případě, že bude vyžadováno, aby docházelo k sejmutí číselníku každou hodinu, vychází v ideálních podmínkách doba výdrže kapacity zvoleného akumulátoru přibližně na 75 dní chodu. Tuto dobu lze samozřejmě prodloužit například zvýšením délky intervalů mezi snímáním číselníků. Je také vhodné poznamenat, že se nedoporučuje nechávat baterii dosáhnout absolutního vybití, protože to bude mít s největší pravděpodobností negativní vliv na její životnost a kapacitu.

Jak již bylo popsáno, zvolený akumulátor má výstupní napětí 12V přičemž kamera požaduje vstupní napájení 5V. Vzhledem k existenci výše zmíněné přítomnosti napěťového regulátoru s maximálním 15V vstupním limitem je možné připojit baterii k napájecímu vstupu napřímo, aniž by došlo k poškození snímacího senzoru. Nicméně tento přístup nelze doporučit z důvodu generování velkých tepelných ztrát. Proto byl do boxu obsahujícího kameru integrován DC-DC step down měnič, tedy zařízení sloužící k převodu stejnosměrného napětí znovu na stejnosměrné (DC-DC) a dále jeho snížení (step down) při téměř žádných ztrátách.

Velikost výstupního napětí je limitována velikostí vstupního, které může být v intervalu 4,5V-40V, a v rámci tohoto intervalu je regulovatelná podle potřeb uživatele v intervalu 3V-35V. V našem případě jde tedy o převod z 12V na 5V. [27]

4 VYHODNOCENÍ

Po implementaci dílčích řešení, ze kterých se tato práce skládá, došlo na jejich vzájemnou integraci a testování celého systému jako kompletního celku. Instalace a spuštění probíhalo v zamýšleném rozsahu, tedy upevnění tří vzdálených periférií založených na platformě ESP32 – CAM na číselníky jednotlivých přístrojů měřících energie, tedy elektroměr, vodoměr a plynoměr. Každá periferie byla zvlášť připevněna na své místo pomocí dedikovaného fixačního zařízení navrženého v rámci této práce. Dále pak byla spuštěna platforma Raspberry Pi s běžícím lokálním serverem a skriptem sloužícím k vyhodnocování fotografií číselníků zaslaných z kamerových systémů. Komunikace mezi periferními zařízeními a serverem probíhala bezdrátovým připojením přes směrovač, sloužící v tomto případě jako místní přístupový bod.

Během tohoto kompletního spuštění, a nepřeberného množství předchozích částečných oživení systému bylo dosaženo následujících zjištění, výsledků a zkušeností.

Sběr a zasilání dat na lokální server funguje spolehlivě očekávaným způsobem. Integrované antény na vývojových deskách ESP32 – CAM mají dostatečný dosah bezdrátového připojení v rámci sklepení rodinného domu na 2,5Ghz síti, nehledě na překážky (včetně kompletního uzavření do rozvaděče) dokud se s přijatelně výkonným směrovačem nacházejí ve stejném patře. Pro větší spolehlivost je také vhodné si pohlídat rozsah obsazených kanálů. Platforma obrazového senzoru případně umožňuje instalaci externí antény, nicméně v rámci tohoto testování nebylo takové vylepšení nutné.

Samotný obrazový senzor (ne celá deska, pouze CMOS kamera) se při použití velmi rychle zahřívá takovým způsobem, že už po minutě má teplo negativní vliv na kvalitu fotografie, což poněkud komplikuje provedení optimalizace nastavení samotného obrazového senzoru. Nicméně při samotné operaci, kdy se v rámci poměrně krátkého časového intervalu kamera probudí, sejme číselník, odešle fotografii a zase usne, nedostane obrazový senzor dostatek času k tomu, aby se mohl zahřát, a veškeré vygenerované teplo se pak ztratí do okolí v době spánku, který je násobně delší, než doba činnosti kamery.

Rozpoznání číslic přinášelo výsledky, které byly velmi závislé na kvalitě osvětlení číselníku a přítomnosti odlesků v zájmových oblastech.

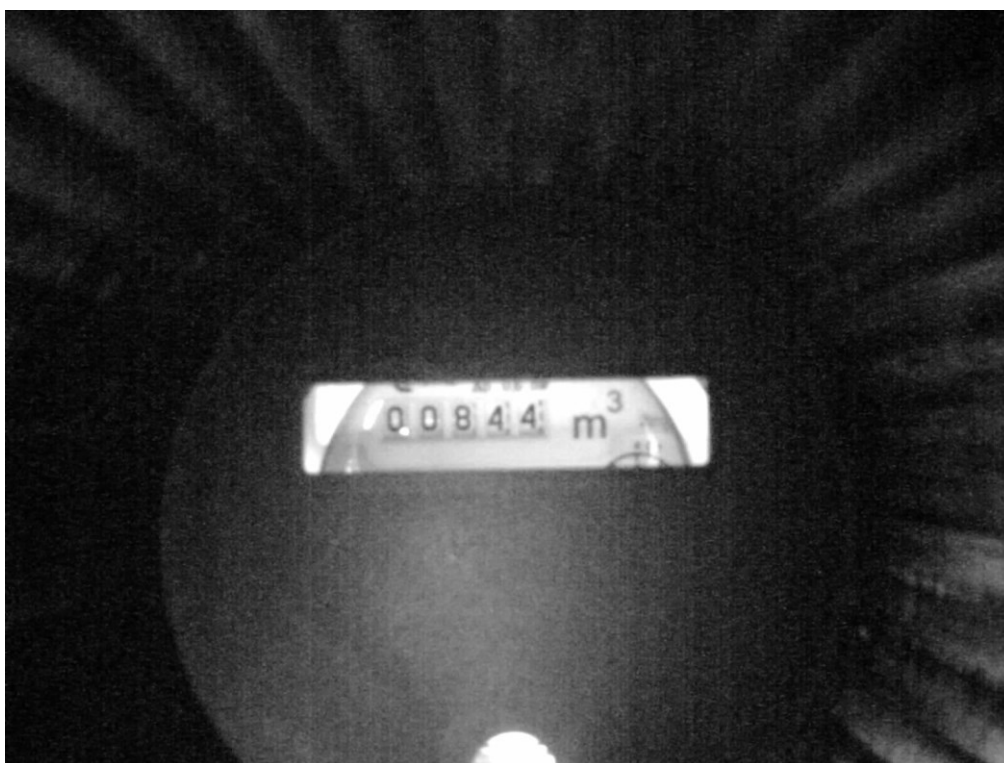
Ke každému typu číselníku byla provedeno několik sérií měření. Pro demonstraci v této práci byly vybrány fotografie s různou kvalitou výsledků za účelem diskuze vzniklých chyb a jejich případných řešení.

4.1 Rozpoznání číslic vodoměru

Na obrázcích níže jsou viditelné výsledky rozpoznání znaků z číselníků vodoměru.

V prvním případě došlo k identifikaci obsahu číselníku číslicemi 0944. První nula byla vynechána z identifikace, protože splývala s blízkým odleskem, respektive jeho hranicí, která byla při segmentaci obrazu prahována na podobnou hodnotu jako objekt číslice. Z tohoto důvodu se kontura vzniklá spojením odlesku a nuly nekvalifikovala jako číslo a byla ignorována.

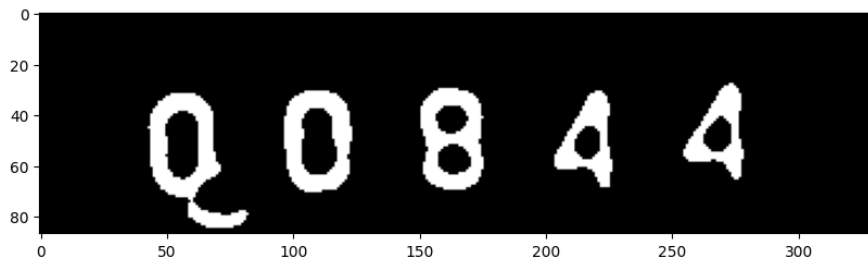
Pokud jde o záměnu osmičky za devítku, nenabízí se žádné na první pohled evidentní vysvětlení tohoto přehmatu, kromě ztráty informace při vytvoření fotografie a během jejích následujících úprav, které předcházely samotné identifikaci.



Obrázek 39. Vodoměr č. 1

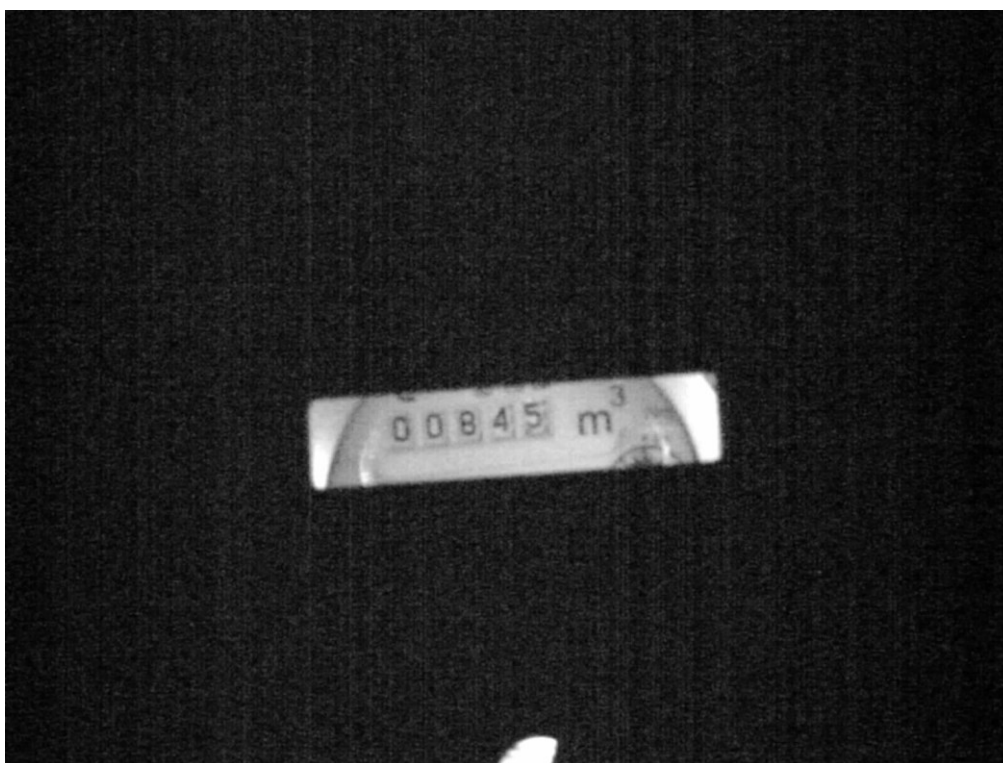
Lze tedy říci, že vzhledem k tomu, že při vyčištění obrazu a jeho segmentaci došlo k určité ztrátě detailů, která se v případě této špatně klasifikované osmičky projevila jako smazání většiny rozdílu mezi středovým zúžením a smyčkami a to především na pravé straně objektu, zatímco nalevo je stále znatelná určitá křivka, stala se tím tato kontura pro kNN dostatečně podobná číslu devět.

Je také vhodné připomenout, že při průběhu samotného klasifikačního algoritmu je zpracováván obraz číslice o rozměrech 10 na 10 pixelů, což celkem bez diskuze má další vliv na ztrátu většiny nedostatečně výrazných detailů.



Obrázek 40. Separované znaky vodoměru č. 1

Další způsob chybné klasifikace čísla 8 se nachází na následujícím obrázku. Zde z hodnotícího skriptu vypadla hodnota 00045, tedy číslo osm bylo v tomto případě interpretováno jako 0.



Obrázek 41. Vodometer č. 2

Není mi příliš jasné, z jakého důvodu došlo při prahování obrazu k seříznutí levé strany čísla osm. Určitá nabízející se teorie odkazuje na možnou kolizi mezi objektem číslice a obvodového rámečku při prahování, z čehož je patrné, že ani v případě téměř dokonalého odstranění, či spíše předejití tvorbě odlesků v zájmové oblasti nemusí nutně nastat ideální podmínky pro rozpoznání znaku.

Pravděpodobně by v rámci optimalizace tohoto obrazu bylo vhodné jednak dosažení lepší (vodorovné) polohy číselníku vůči kameře a především vyššího kontrastu mezi znaky a pozadím číselníku.



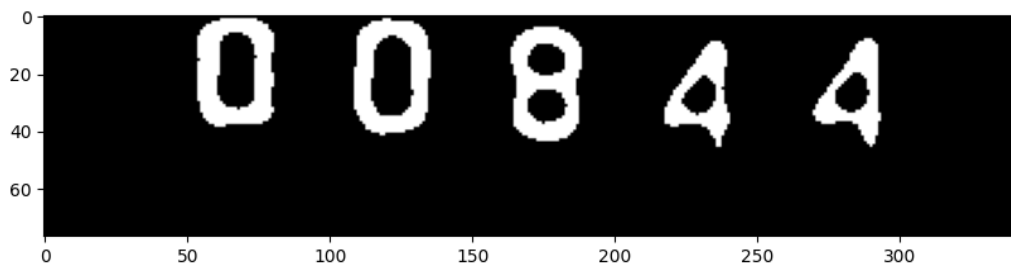
Obrázek 42. Separované znaky vodoměru č. 2

Poslední (i když ne chronologicky) případ vyhodnocení proběhl korektně, kdy výsledná číselná řada odpovídala fotografii, tedy 00844. Hlavní rozdíl mezi tímto a předchozími obrázky je výraznější osvětlení číselníku, kdy celé pozadí prakticky září, zatímco číslice si ponechávají dostatek kontrastu.



Obrázek 43. Vodometer č. 3

Na zpracovaném obrázku pak není pouhým okem příliš patrné, že by tento způsob nasvícení nějak výrazně změnil k lepšímu detaily čísel nula a čtyři v porovnání s předchozími výsledky, nicméně číslo osm si viditelně uchovalo dostatečné množství výrazných vlastností, aby byla provedena jeho úspěšná klasifikace.



Obrázek 44. Separované znaky vodoměru č. 3

4.2 Rozpoznání číslic elektroměru

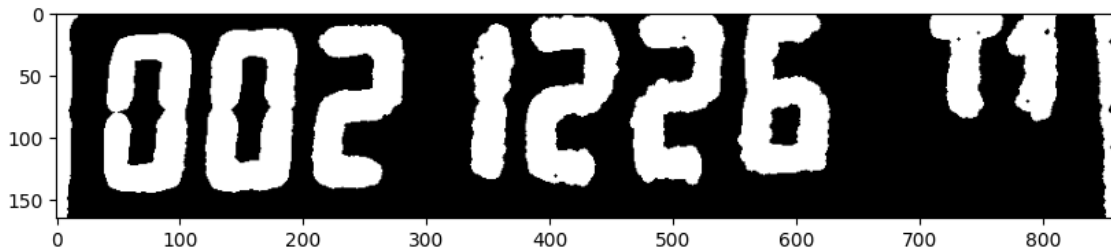
Zde je diskutováno provedení a vyhodnocení rozpoznání číslic na digitálních displejích elektroměrů. Nutno poznamenat, že jejich snímání a zpracování bylo asi technicky nejjednodušší a to z důvodu integrovaného podsvícení jejich displeje, díky čemuž je generován dostatečný kontrast k separaci objektů od pozadí a zároveň si člověk nemusí dělat hlavu s řešením, které by předcházelo tvorbě odlesků při pokusu o osvětlení zájmové oblasti.

Další výhodou digitálních displejů je, že segmentová čísla jsou velmi jednoduše vzájemně rozlišitelná do takové míry, že smývání jemnějších avšak důležitých vlastností jednotlivých čísel při pořízení a úpravě fotografie není v tomto případě zdaleka tak relevantní problém jako v případě analogových číselníků.



Obrázek 45. Elektroměr č. 1

Výsledek identifikace číselníku tohoto elektroměru skončil korektním způsobem, tedy číslem 0021226. Dokonce se z průběhu rozpoznání znaků povedlo vyřadit i přítomné značení tarifu T1.



Obrázek 46. Separované znaky elektroměru č. 1

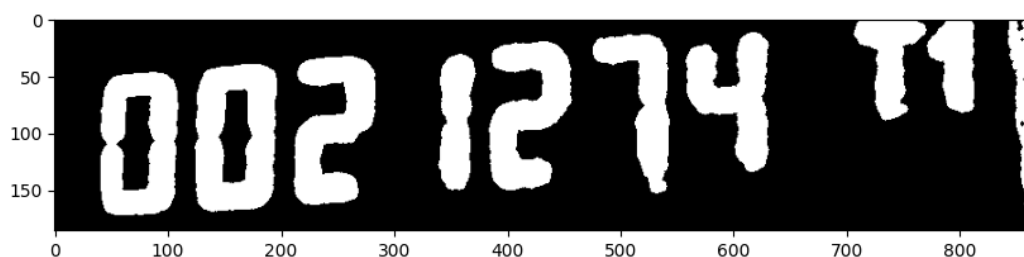
Výsledek rozpoznání znaku na číselníku tohoto elektroměru byl 0021274X. Vzhledem k relativně dobré kvalitě fotografie s dobrým osvětlením bez odlesků není příliš velkým překvapením, že i výsledek korektně zobrazuje přítomnou hodnotu na číselníku s jednou výjimkou, kterou je písmeno X nacházející se na konci řady.



Obrázek 47. Elektroměr č. 2

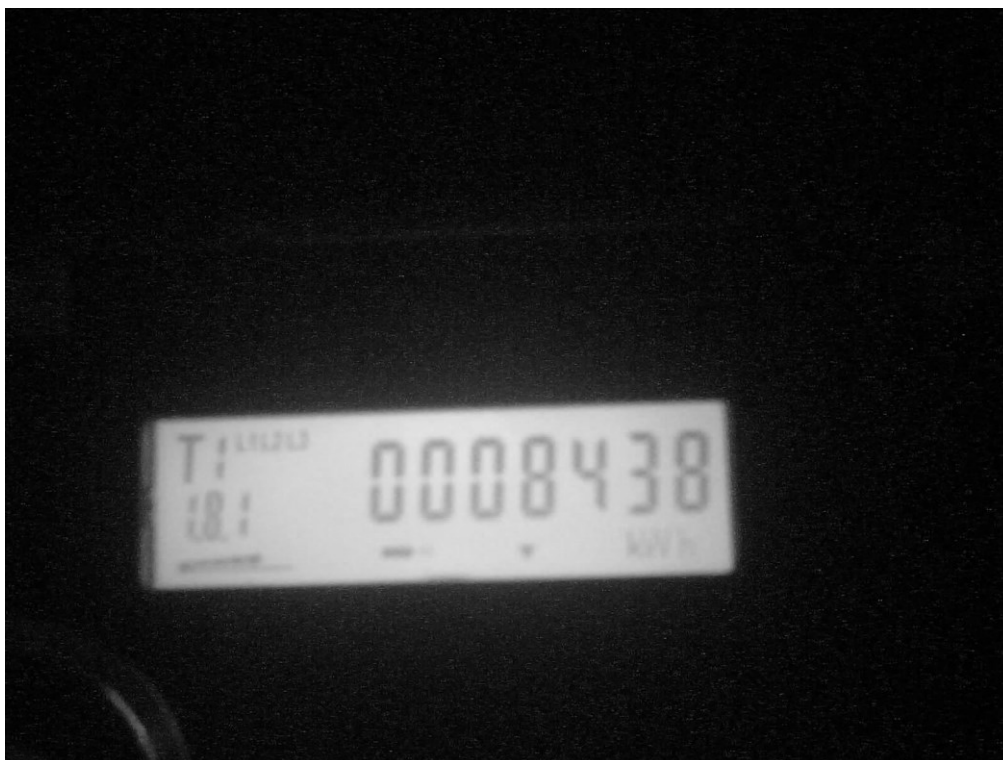
Jak již bylo zmíněno dříve v této práci, písmeno X značí nedefinovanou kombinaci jednotlivých dílů, ze kterých se každá jedna číslice skládá. Zde tedy došlo i k extrakci jednoho

nečíselného objektu, či spíše přesněji kontury, jejíž vlastnosti nespádaly do intervalu definovaných možností. V tomto případě jde o zbytek okraje displeje na pravém konci zpracovávaného obrázku. Tato nedokonalost by se dala upravit nastavením hodnot ořezávající okraje obrázků. Při testování byly tyto hodnoty v případě vertikálních okrajů ponechány na nule, z důvodu co největší univerzálnosti při čtení číselníků.



Obrázek 48. Separované znaky elektroměru č. 2

Na posledním prezentovaném obrázku elektroměru je demonstrativně vyobrazen jiný typ obrazovky, lišící se především celkovým rozložením objektů v rámci displeje. I tento číselník byl identifikován korektně, tedy čísly 0008438.

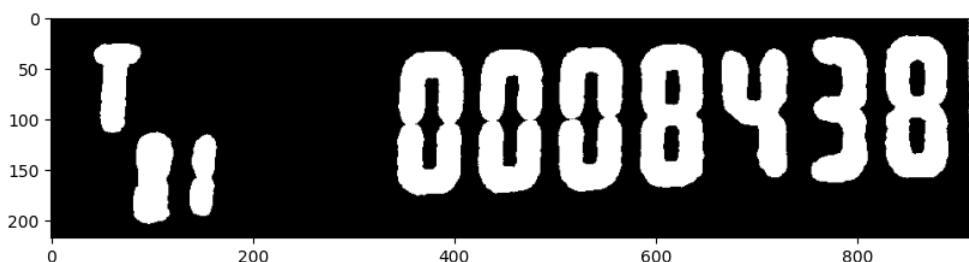


Obrázek 49. Elektroměr č. 3

Jak lze vidět, i zde se ve vyčištěném binárním obraze nacházejí pro nás redundantní informace, nicméně na rozdíl od předchozích obrazovek se zde vyskytují na levé straně, přičemž i zde se je podařilo vyloučit ze samotné identifikace.

Ještě lepší a spolehlivější způsob vyhýbání se zpracování nežádoucích prvků by spočíval v oříznutí číselníku do té míry, kdy se ve výhledu vyskytují jen požadovaná čísla. Skript provádějící identifikaci je sice schopen najít pozici číselníku v celé fotografii jako na obrázku výše, ale má poměrně omezené možnosti vytyčení zájmové oblasti přesnějším způsobem.

Toho lze naopak dosáhnout nastavitelnými hodnotami specifikujícími ořezávání okrajů, které se nicméně musí stanovit manuálně při prvotním oživení s ohledem na typ číselníku.



Obrázek 50. Separované znaky elektroměru č. 3

4.3 Rozpoznání číslic plynoměru

Jako poslední jsou zde prezentovány výsledky ze zařízení měřících spotřebu plynu. U nich docházelo asi k největší kvalitativní variaci v průběhu měření. Jedním z důvodů tohoto stavu je rozdílná barva pozadí a rámování číslic v jednotlivých částech číselníku, což významně komplikovalo separaci žádaných objektů od zbytku obrazu, a dále neuniformní poloha jednotlivých čísel, které buď byly částečně překryty okrajovými lištami, kvůli čemuž pak splývaly s okolím, popřípadě se rovnou nacházely v mezipoloze, kdy ve výhledu zrovna docházelo ke střídání dvou číslic, kdy se pak ani jedna nekvalifikovala jako objekt vhodný k identifikaci.

Na obrázcích níže se nachází první diskutovaný výsledek ze snímání plynoměrů. Zde při rozpoznávání obsahu číselníku došlo ke vcelku neuspokojivému výsledku 178757, který poměrně významným způsobem neodpovídá skutečnému stavu. Do této práce byl zařazen jako vhodný příklad pro demonstraci způsobů, které mohou negativně ovlivnit průběh vyhodnocování a to i v případě obrázku, se kterým by člověk neměl při čtení čísel nejmenší problém.

Už na první pohled lze vidět světlý odraz v horní části čísla 0, který po zpracování vytváří cosi na způsob ploché střechy s převisem. A vzhledem k tomu, že kNN algoritmus nedokáže rozhodovat, které části zpracovávaného objektu jsou nepodstatné, ale bere všechno

jako jeden celek, který se snaží přiřadit k existující třídě, tak není nepředstavitelné, že nula se po kompresi na 10 pixelů jevila jako číslo 1.



Obrázek 51. Plynoměr č. 1

Další případ na toto téma je následující číslo dva, které kvůli působení odlesku také dostalo převis do své přední horní části a bylo identifikováno jako číslo 7.

Celkem výraznou chybou je absolutní vymazání celé jedné číslice (v tomto případě dvojky) z číselníku. K této chybné úpravě došlo proto, že daný objekt kvůli odlesku ve své dolní části splýval s osvětleným okolím, aby pak byl při úpravě obrazu zaměřené na nalezení číselníku považován za součást hraniční oblasti a došlo tedy k jeho odstranění spolu s ní.

Poslední pozice na obrázku nacházející se v přechodu mezi čtyřkou a pětkou se při vyhodnocování nekvalifikovala jako číslo a nebyla brána v potaz pro klasifikaci.



Obrázek 52. Separované znaky plynoměru č. 1

Zde byla provedena identifikace znaků na obrázku níže obsahující stejný číselník jako předchozí příklad, ale za změněných světelných podmínek, kdy byly provedeny úpravy za účelem odstranění odlesků způsobujících selhání klasifikace. Díky tomu se rozpoznávání oběšlo téměř bez problémů, s výjimkou vyskytující se znovu na poslední pozici, kdy se znovu ani jedno z částečně viditelných čísel nekvalifikovalo jako vhodný objekt.



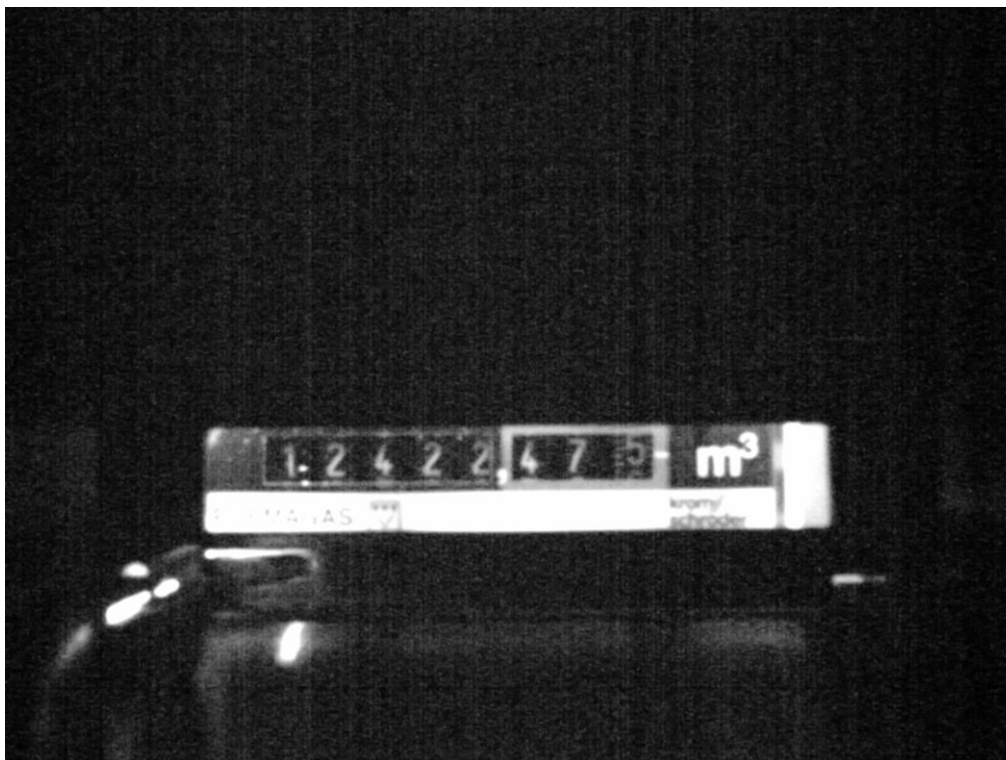
Obrázek 53. Plynoměr č. 2

Výsledek rozpoznání znaků v obraze je tedy v tomto případě 0287257.



Obrázek 54. Separované znaky plynoměru č. 2

Jako poslední příklad byl vybrán následující obrázek číslo 55. Důvodem jeho výběru je, že na první pohled působí jako pro zpracování obrazu méně vhodný než obraz 53 a to z důvodu znatelně nižší intenzity osvětlení. Ta má za následek nižší kontrast číslic oproti pozadí, což by mohlo vyústit v horší průběh separace a vyšší pravděpodobnost, že některá čísla, případně jejich části budou ve výsledném segmentovaném obraze chybět. Nicméně výsledek je 124224711, a tedy nám dokonce jedno číslo přebývá.



Obrázek 55. Plynoměr č. 3

Jak lze vidět při porovnání originálního a binárního obrazu, špatně klasifikovaný objekt čísla 5 nebyl určen nesprávně kvůli přilepenému odlesku, ale kvůli splynutí s jednou z okolních rysek. Mimochodem, v případě manuálního zásahu, kdy došlo k odmazání rysky, byl pak daný objekt určen správně jako pětka.

Jako přebytečná jednička byl pak klasifikován zbytek okraje, který v obraze zůstal i po předchozích krocích provedeného čištění za účelem odstranění redundantních objektů.



Obrázek 56. Separované znaky plynoměru č. 3

4.4 Komentář

Jak lze z jednotlivých kapitol diskutujících výsledky sebrané rozpoznáním obrazu z fotografií číselníků měřících zařízení vyčíst, hodnota získaných informací je silně závislá na schopnosti pořízení vhodné fotografie, jejíž kvalita je závislá na způsobu provedení osvětlení zajišťujícího dostatečný kontrast obrázku.

Další důležitý parametr týkající se osvětlení je co nejdokonalejší potlačení odlesků v okolí objektů, se kterými hodláme pracovat, jelikož i jen v malém množství dokážou při identifikaci znaků zmást klasifikační program tak, že produkuje výsledky, které potom neodpovídají skutečnosti. Případně také mohou způsobit, že daný objekt bude odstraněn už v rámci čištění obrazu od redundantních kontur a tedy se potenciální číslo ani nemusí ke klasifikaci dostat.

Je vhodné poznamenat, že v rámci práce bylo pro program zabývající se čištěním a segmentováním obrazu použito obecné nastavení, které zajišťovalo přijatelné výsledky i při lehce se měnících podmínkách osvětlení či umístění, které vznikaly v rámci projektování a testování fyzického vybavení systému. Zvolené nastavení též umožňovalo obecné použití v případě, kdy byly přítomny nějaké rozdíly mezi testovanými číselníky měřičů energií.

V ostrém nasazení, kdy dojde k jednorázovému usazení obrazových senzorů na jejich pozice, přičemž se v dohledné době nepředpokládá žádná manipulace ani úprava jejich fyzické stránky, je doporučeno projít si kvalitu zpracování obrazu a provést úpravu parametrů kamery a samotného programu vzhledem k stabilně panujícím světelným podmínkám. Program také umožňuje nastavení parametrů ořezu zájmové oblasti, což dále může mít kladný vliv na selekci relevantních informací a odstraňování nepotřebných částí obrazu.

Poznámka na závěr. Projektování fixačních zařízení kamery bylo sice prováděno s ohledem na co nejširší použitelnost, nicméně je nemožné odhadnout vhodnost jejich použití na přístrojích s číselníky o jiném formátu, na nichž zařízení nebyla testována z důvodu chybějícího přístupu. Momentálně jsem si vědom existence plynových měřičů, jejichž číselník ukazující spotřebu se nachází v horní části informačního rámečku, místo typu s číselníkem v dolní pozici, na kterém bylo vyčítání testováno. Tento problém by měl být řešitelný opačným umístění kamery a vnitřní osvětlovací vložky (takříkajíc v pozici vzhůru nohama), nicméně jestli výřez opravdu pasuje, tak aby byla čísla čitelná, v současné době není známo.

ZÁVĚR

Cílem této práce bylo podle jednotlivých bodů zadání vytvořit a otestovat systém snímání číselníků přístrojů pro měření spotřeby energií. Tento soubor úkolů zahrnoval jednak teoretickou část spočívající ve zpracování informací souhrnně referujících o problematice strojového vidění, postupů a nástrojů, které jsou za tímto účelem využívány.

Následující praktická část spočívala ve výběru vhodných zařízení obsluhujících jednotlivé části aplikace, mezi které patřila kamera schopná bezdrátové komunikace za účelem odesílání pořízených fotografií a centrální jednotka hostující místní server za účelem příjmu a zpracování obrazu a dále sloužící k prezentaci výsledků uživateli v rámci místní sítě.

V případě periférií sloužících ke snímání obrazu bylo tedy třeba poskládat jejich funkční integraci s prostředím zájmové oblasti tak, aby byl výsledek využitelný k dalšímu zpracování v požadovaném rozsahu. To mimo jiné znamenalo zajištění dostatečné kvality osvětlení snímaného měřiče nehledě na případné světelné podmínky okolního prostředí a vyprojektování spolehlivého umístění obrazového senzoru se stabilním a konzistentním výhledem na požadovaný objekt.

Ústřední součástí práce bylo vytvoření programu schopného přebírat fotografie zaslané od jednotlivých zařízení a jejich zpracování za účelem extrakce informace o udávané energetické spotřebě. V rámci tohoto úkolu byla provedena série operací sloužící k nalezení číselníku v obdrženém obraze, jeho vyčištění od rušivých vlivů a vytěžení dat ve formě rozpoznání jednotlivých číslic získaných ze segmentovaného obrazu.

Pro samotné rozpoznání znaků byly využity dva způsoby přístupu k řešení této problematiky. Identifikace čísel na analogových přístrojích byla provedena za nasazení algoritmu využívajícího prvky strojového učení k – Nearest Neighbours a pro identifikaci jednotlivých číslic přečtených z digitálních displejů byl vytvořen postup založený na předpokladu, že jednotlivé číslice digitálního displeje se skládají ze stejného počtu dílčích segmentů, jejichž pozice je známa a kombinace, v jakém se tyto segmenty na číselníku rozsvítí, jednoznačně identifikuje každé číslo.

Ve výsledku lze konstatovat, že vytvořený systém pracuje podle očekávání, nicméně v případě každého nasazení je doporučeno projít nastavitelné vlastnosti kamery a rozpoznávacího systému za účelem optimalizace jejich funkce s ohledem na vlastnosti prostředí a získaného obrazu.

SEZNAM POUŽITÉ LITERATURY

- [1] *Introduction to machine vision* In: *cognex.com* [online]. c2024 [cit. 2024-04-17]. Dostupné z: <https://www.cognex.com/en-cz/what-is/machine-vision>
- [2] *A Practical Guide to Machine Vision Lighting* In: *ni.com* [online]. 21. listopadu 2023 [cit. 2024-04-17]. Dostupné z: <https://www.ni.com/en/shop/choosing-the-right-hardware-for-your-vision-applications/a-practical-guide-to-machine-vision-lighting.html>
- [3] *Anatomy of a Machine Vision System* In: *emergentvisiontec.com* [online]. c2024 [cit. 2024-04-17]. Dostupné z: <https://emergentvisiontec.com/tech-portal/anatomy-of-a-machine-vision-system/>
- [4] *Canon Science Lab: CCD Sensors* In: *global.canon* [online]. [cit. 2024-04-17]. Dostupné z: https://global.canon/en/technology/s_lab/light/003/04.html
- [5] *Canon Science Lab: CMOS Sensors* In: *global.canon* [online]. [cit. 2024-04-17]. Dostupné z: https://global.canon/en/technology/s_lab/light/003/05.html
- [6] *What Is Optical Character Recognition (OCR)?* In: *ibm.com* [online]. 5 ledna 2022 [cit. 2024-04-17]. Dostupné z: <https://www.ibm.com/blog/optical-character-recognition/>
- [7] *What is OCR (Optical Character Recognition)?* In: *aws.amazon.com* [online]. c2024, [cit. 2024-04-17]. Dostupné z: <https://aws.amazon.com/what-is/ocr/>
- [8] WOODFORD, Chris. *Optical character recognition (OCR)* In: *www.explainthatstuff.com* [online]. 22. února 2023 [cit. 2024-04-17]. Dostupné z: <https://www.explainthatstuff.com/how-ocr-works.html>
- [9] CRABTREE, Matt. *What is Machine Learning? Definition, Types, Tools & More* In: *datacamp.com* [online]. c2024, červenec 2023 [cit. 2024-04-17]. Dostupné z: <https://www.datacamp.com/blog/what-is-machine-learning>
- [10] UC BERKELEY School of information. *What Is Machine Learning (ML)?* In: *ischoolonline.berkeley.edu* [online]. c2024, 26 června 2020 [cit. 2024-04-17]. Dostupné z: <https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/>
- [11] *Comparing supervised and unsupervised learning.* In: *datacamp.com* [online]. c2024 [cit. 2024-04-17]. Dostupné z: https://images.datacamp.com/image/upload/v1689699751/Comparing_supervised_and_unsupervised_learning_af4d5eccb0.png

- [12] *Understanding k-Nearest Neighbour* In: *docs.opencv.org* [online]. 15. května 2024 [cit. 2024-04-17]. Dostupné z: https://docs.opencv.org/4.x/d5/d26/tutorial_py_knn_understanding.html
- [13] ALIZADEH, Esmail. *What K is in KNN and K-Means* In: *ealizadeh.com* [online]. 21. března 2022 [cit. 2024-04-17]. Dostupné z: <https://ealizadeh.com/blog/knn-and-kmeans/>
- [14] JAIN, Ramesh, KASTURI, Rangachar, SCHUNCK, Brian G. *Machine vision*. Mcgraw-Hill, 1995 [cit. 2024-04-17]. ISBN: 978-0-07-032018-5.
- [15] BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000
- [16] OPENCV In: *OpenCV Reference Manual c2024*, [cit. 2024-04-17]. Dostupné z: <https://docs.opencv.org/>
- [17] BUADES Antoni, COLL, Bartolomeu, MOREL, Jean-Michel. *Non-Local Means Denoising, Image Processing On Line*, str. 208–212. In: *ipol.im* [online]. 13. září 2011. [cit. 2024-04-17] Dostupné z: https://doi.org/10.5201/ipol.2011.bcm_nlm
- [18] *image*. In: *docs.opencv.com* [online]. 14. května 2024 [cit. 2024-04-17]. Dostupné z: https://docs.opencv.org/4.x/ada_threshold.jpg
- [19] *ESP32 –CAM Development Board* In: *DFRobot.com* [online]. [2019] [cit. 2024-04-17]. Dostupné z: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf
- [20] *Schematic diagram*. In: *DFRobot.com* [online]. [2019] [cit. 2024-04-17]. Dostupné z: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf
- [21] *Dimension diagram*. In: *DFRobot.com* [online]. [2019] [cit. 2024-04-17]. Dostupné z: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf
- [22] *Arducam OV2640*. In: *ArduCam.com* [online]. c2024 [cit. 2024-04-17]. Dostupné z: https://www.arducam.com/wp-content/uploads/2021/07/Arducam_-_OV2640_-_Camera_Module_M0031-1.jpg
- [23] *Raspberry Pi 4B*. In: *Arvutitark.ee* [online]. c2024 [cit. 2024-04-17]. Dostupné z: <https://arvutitark.ee/en/pcs-and-accessories/raspberry-pi/raspberry-pi-4b-4x-15-ghz-4-gb-ram-wifi-bt-soc-mini-m-1272985>

- [24] *Raspberry Pi 4B*. In: *Arvutitark.ee* [online]. c2024 [cit. 2024-04-17]. Dostupné z: <https://arvutitark.ee/en/pcs-and-accessories/raspberry-pi/raspberry-pi-4b-4x-15-ghz-4-gb-ram-wifi-bt-soc-mini-m-1272985#images>
- [25] *ESP32 – CAM schematic diagram* In: *docs.ai-thinker.com* [online]. 30. ledna 2018 [cit. 2024-04-17]. Dostupné z: https://docs.ai-thinker.com/_media/esp32/docs/esp32_cam_sch.pdf
- [26] *AMS1117: 1A Low Dropout Voltage Regulator* In: *advanced-monolithic.com* [online]. [cit. 2024-04-17]. Dostupné z: <http://www.advanced-monolithic.com/pdf/ds1117.pdf>
- [27] *Step down nastaviteľný měnič s LM2596 DC-DC* In: *dratek.cz* [online]. c2021 [cit. 2024-04-17]. Dostupné z: <https://dratek.cz/docs/produkty/1/1243/1463386307.pdf>
- [28] LUTZ, Mark. *Learning Python. 4th ed.* Beijing O'Reilly, 2009. ISBN 978-0-596-15806-4
- [29] HALFACREE, Gareth. *The Official Raspberry Pi Beginners's Guide. 4th edition.* Raspberry Pi Trading. ISBN 978-1-912047-62-8
- [30] SZELISKI, Richard. *Computer vision: algorithms and applications. Texts in computer science (Springer)*. London: Springer, 2011. IBN 978-184-8829-343.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

A	ampér
Ah	ampérhodina
CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide-Semiconductor
CNC	Computer Numerical Control
csv	Comma separated values
kNN	k-Nearest Neighbours
LED	Light Emitting Diode
mA	miliampér
OCR	Optical character recognition
PWM	Pulse-width modulation
SCCB	Serial Camera Control Bus
UART	Universal asynchronous receiver-transmitter
UI	Umělá inteligence
V	volt

SEZNAM OBRÁZKŮ

Obrázek 1. Intenzita světelných zdrojů v závislosti na vlnové délce [2].....	10
Obrázek 2 Efektivita obrazových senzorů při dané vlnové délce [2].....	10
Obrázek 3. Zadní osvětlení [1]	11
Obrázek 4. Pruhové osvětlení [1]	11
Obrázek 5. Osvětlení temným polem [1].....	12
Obrázek 6. Koaxiální osvětlení [1].....	12
Obrázek 7. Kupolové osvětlení [1].....	13
Obrázek 8. Glyfy čísla 9	17
Obrázek 9. Učení s učitelem a bez učitele [11]	18
Obrázek 10. Vizualizace rozdílného výsledku v závislosti na velikosti k [13]	20
Obrázek 11. Příklad výsledku algoritmu NLmeans Originální, zašuměný a znovu vyčištěný obrázek [17]	22
Obrázek 12. Rozdíl mezi globálním a adaptivním prahováním [18].....	23
Obrázek 13. Schématický diagram ESP-32 CAM [20]	26
Obrázek 14. Rozměrový diagram ESP-32 CAM [21]	27
Obrázek 15. Kamera OV2640 [22].....	27
Obrázek 16. Raspberry Pi 4B [24].....	28
Obrázek 17. Schéma sítě systému.....	29
Obrázek 18. Výchozí obraz ke zpracování	31
Obrázek 19. Výsledek odstranění šumu	32
Obrázek 20. Segmentace neupraveného obrazu	33
Obrázek 21. Segmentace obrazu po odstranění šumu	33
Obrázek 22. Konečná lokace číselníku.....	34
Obrázek 23. Vyčištěný analogový číselník s nalezenými objekty číslic	34
Obrázek 24. Glyfy 4,8 a 1 ve formátu 10x10 pixelů	35
Obrázek 25. Vyčištěný digitální číselník s nalezenými objekty číslic	35
Obrázek 26. Definice pozice segmentů v kontuře číslice	35
Obrázek 27. Porovnání pozice pravého horního segmentu	36
Obrázek 28. Porovnání pozice horní dutiny	36
Obrázek 29. Slovník definující číslo na základě přítomnosti segmentů.....	37
Obrázek 30. Úložný box pro ESP-32 CAM	39
Obrázek 31. Umístění kamery na elektroměru	39

Obrázek 32. Nástavec na vodoměr	40
Obrázek 33. Detail umístění světelného vlákna ve vložce v pohledu od číselníku	40
Obrázek 34. Sestava vložky a přichytného bodu kamery	41
Obrázek 35. Umístění kamery na vodoměru	41
Obrázek 36. Sestava pro snímání plynoměru	42
Obrázek 37. Detail umístění světelného vlákna ve vložce v pohledu od kamery	42
Obrázek 38. Umístění kamery na plynoměru	43
Obrázek 39. Vodoměr č. 1	46
Obrázek 40. Separované znaky vodoměru č. 1	47
Obrázek 41. Vodoměr č. 2	47
Obrázek 42. Separované znaky vodoměru č. 2	48
Obrázek 43. Vodoměr č. 3	48
Obrázek 44. Separované znaky vodoměru č. 3	49
Obrázek 45. Elektroměr č. 1	49
Obrázek 46. Separované znaky elektroměru č. 1	50
Obrázek 47. Elektroměr č. 2	50
Obrázek 48. Separované znaky elektroměru č. 2	51
Obrázek 49. Elektroměr č. 3	51
Obrázek 50. Separované znaky elektroměru č. 3	52
Obrázek 51. Plynoměr č. 1	53
Obrázek 52. Separované znaky plynoměru č. 1	53
Obrázek 53. Plynoměr č. 2	54
Obrázek 54. Separované znaky plynoměru č. 2	54
Obrázek 55. Plynoměr č. 3	55
Obrázek 56. Separované znaky plynoměru č. 3	55

SEZNAM PŘÍLOH

Příloha P I: Adresářová sktruktura přiloženého CD

PŘÍLOHA P I: ADRESÁŘOVÁ STRUKTURA PŘILOŽENÉHO CD

- diplomova_prace.pdf
- 3D_tisk
 - box_kamera.SLDPRT
 - světlo_objímka.SLDPRT
 - plyn
 - držák_plyn_1.SLDPRT
 - držák_plyn_2.SLDPRT
 - kryt_plyn.SLDPRT
 - nástavec_plyn.SLDPRT
 - stínítko_plyn.SLDPRT
 - tunel_plyn.SLDPRT
 - vložka_plyn.SLDPRT
 -
 - voda
 - kryt_voda.SLDPRT
 - spojka_voda.SLDPRT
 - tunel_voda..SLDPRT
 - vložka_voda.SLDPRT
- SW
 - Arduino
 - index_html.h
 - webserver_settings.ino
 - PHP
 - index.php
 - upload.php
 - Python
 - glyphResponses.data
 - glyphSamples.data
 - ocr_script.py