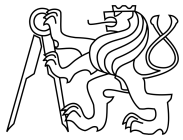
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Aplikace pro podporu výuky matematiky na základních školách
Student:	Bc. Petr Zavadil
Vedoucí:	doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Studijní program:	Informatika
Studijní obor:	Znalostní inženýrství
Katedra:	Katedra teoretické informatiky
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Matematika není při současném způsoby výuky příliš atraktivní. Do popředí se ale dostávají principy tzv. Hejného metody, která přistupuje k výuce matematiky alternativně a mnohem zábavněji. Cílem práce je navrhnout a implementovat algoritmy pro softwarovou aplikaci na tabletová zařízení, která umožní pomocí vestavěné kamery vyhodnocovat uspořádání očíslovaných kostek, které žák fyzicky před kamerou staví, aby splnil zadanou úlohu.
1) Seznamte se s Hejného metodou a možnostmi, které výpočetní technika nabízí pro řešení úloh této výukové metody.
2) Navrhněte algoritmy, které umožní snímat pomocí vestavěné kamery v tabletu scénu, ve které žák řeší zadanou úlohu s pomocí kostek s čísly, a které následně umožní vyhodnotit dosažené řešení.
3) Navržené algoritmy implementujte s pomocí vhodného programového prostředí.
4) Implementované algoritmy ověřte na reálných úlohách a vyhodnoťte dosažené výsledky.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 1. března 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra teoretické informatiky

Diplomová práce

Aplikace pro podporu výuky matematiky na základních školách

Bc. Petr Zavadil

Vedoucí práce: doc. RNDr. Ing. Marcel Jiřina, Ph.D.

9. května 2017

Poděkování

Rád bych poděkoval panu doc. RNDr. Ing. Marcel Jiřina, Ph.D. za vedení práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 9. května 2017 .....

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2017 Petr Zavadil. Všechna práva vyhrazena.
Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Zavadil, Petr. *Aplikace pro podporu výuky matematiky na základních školách.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Diplomová práce obsahuje návrh tabletové aplikace s rozšířenou realitou, která je ovládána interakcí s fyzickými objekty v okolí tabletu. Především se pak práce věnuje problematice zpracování obrazu z kamery tabletového zařízení za účelem rozpoznávání a klasifikace fyzických objektů v obraze, které slouží pro ovládání navržené aplikace. Výstupem práce je funkční posloupnost algoritmů zpracování obrazu, která nalezne a klasifikuje dané objekty v obraze. SW část práce je implementovaná v prostředí Matlab. Část práce věnující se zpracování obrazu je zhotovena dle metodiky CRISP-DM.

Klíčová slova Zpracování obrazu, rozpoznávání objektů, klasifikace čísel, rozšířená realita, tablet, aplikace, technologie, vzdělávání, Hejného metoda, matematika

Abstract

This diploma thesis contains design of tablet application with augmented reality. The application is designed to be controlled by manipulation with real objects in surroundings of a tablet device. The major part of thesis deals with image processing task of recognition and classification of defined real objects. The output of thesis is working sequence of image processing algorithms, which

is able to recognize given objects in image. The software part of thesis is implemented in environment of Matlab. The part which is aimed to image processing is developed according to CRISP-DM methodology.

Keywords Image procesing, object recognition, digits classification, augmented reality, tablet device, application, technology, education, Hejný methodology, math

Obsah

Úvod **1**
Vzdělávání a škola 1
Hejného metoda výuky matematiky 3
Technologie a vzdělávání 4
Vize práce 6

1 Cíle práce **7**
1.1 Zaměření práce v kontextu Hejného metody 8

2 Analýza problému **9**
2.1 Analýza základního fungování zamýšlené aplikace 9
2.2 Analýza používání zamýšlené aplikace 10
2.3 Výběr varianty používání aplikace 10
2.4 Ovládání zamýšlené aplikace 11
2.5 Definice scény 13
2.6 Analýza problému snímání scény 14
2.7 Volba referenčního zařízení 15

3 Vývoj potřebných HW komponent **17**
3.1 Návrh HW komponent 17
3.2 Výroba komponent 23

4 Vývoj SW dle CRISP-DM **25**
4.1 Pochopení problému 26
4.2 Pochopení dat 33
4.3 Příprava dat 39
4.4 Modelování 54
4.5 Evaluace procesu 59

Závěr **63**

A Seznam použitých zkratek **65**

Literatura **67**

B Obsah přiloženého CD **69**

Seznam obrázků

1.1 Ukázka úloh Součtové trojúhelníky, Pavučiny, Hadi 8

2.1 Varianty umístění tabletu 10
2.2 Nákres zvoleného řešení polohy tabletu 12
2.3 Nákres scény (pohled shora) 13
2.4 Model snímání scény 14

3.1 Nákres řezu zrcátkovým klipem umístěným na frontální kameře tabletu 18
3.2 Model celé sestavy v nástroji Geogebra 20
3.3 Detail modelu zrcátkového klipu v rámci celé sestavy v nástroji Geogebra 20
3.5 Detail modelu zrcátkového klipu v rámci celé sestavy v nástroji Geogebra 22
3.8 Realizovaná sestava tablet - stojan - zrcátkový klip, žetony 24

4.1 Realizovaná sestava tablet - stojan - zrcátkový klip, žetony 26
4.2 Hledané algoritmické workflow zpracování obrazu v kontextu celé aplikace 27
4.3 Funkční bloky hledaného algoritmického workflow 28
4.4 Model projektivního a ortogonálního promítání scény 29
4.6 Ukázková počáteční data 35
4.7 Odraz modrého světla na hraně zrcátka 36
4.10 Výrazný šum ve snímcích špatně osvětlených scén scén 37
4.11 Splynutí okraje žetonu s pozadím scény 38
4.12 Nehomogenního osvětlení v obraze 38
4.13 Funkční bloky hledaného algoritmického workflow 41
4.14 Ukázka jednoho snímku v připraveném datovém setu pro blok Transformace vstupních dat 42
4.17 Princip Bottom-hat transformace 48

4.18 Výsledek aplikování metody adaptivního histogramu s limitovaným kontrastem 49
4.19 Grafické znázornění výstupu rozpoznávání („bounding boxů“) . . . 50
4.20 Segmentace obrazu předzpracovaného Bottom-hat transformací . . 51
4.21 Ukázka datasetu pro trénování rozhodovacího stromu 52
4.22 Model rozhodovacího stromu 52
4.23 Nerozpoznaný žeton v testovacích datech 53
4.25 Architektura sítě LeNet-5 56
4.26 Metice záměn 60
4.27 Výstup celého algoritmu 61
4.28 Selhání klasifikátoru na scéně s vypouklým povrchem 61
4.29 Nepřesná segmentace žetonu ve vzdálenějších polohách od kamery 62

Seznam tabulek

2.1 Porovnání výhod a nevýhod jednotlivých přístupů umístění tabletu 11

4.1 Porovnání frameworků pro rozšířenou realitu 33

Úvod

O důležitosti vzdělávání v životě člověka a v rozvoji společnosti jistě není pochyb. Vzdělání lidem dává možnosti, rozšiřuje obzory a tím dodává větší svobodu při výběru a rozhodování o vlastní budoucnosti. Bezesporu lze říci, že dobré vzdělání je jedním z důležitých faktorů pro vedení radostného, stabilního a spokojeného života.

Vzdělávání a škola

Systematické pojetí vzdělávání je zde v českých zemích zavedeno již od doby Marie Terezie. Systémové školní vzdělávání by dnes mělo dle ministerstva školství žáky připravit na jejich budoucí život a plnit funkci vzdělávací, výchovnou, ochranou, kvalifikační, selekční a integrační [6].

Častým problémem je dnes však to, že dle průzkumů PISA 2012 [5] majorita žáků základních škol v ČR škola nebaví nebo k ní nemají pozitivní vztah, necítí s ní sounáležitost. Například vztah k matematice a výsledky žáků se dlouhodobě zhoršují. Přitom poznávání, které by škola žákům měla zprostředkovat, je nám lidem přirozené. Každé malé dítě má vnitřní touhu po poznávání a objevování nového. Tato vnitřní touha je nám dána od přírody. Dle filozofie svodného vzdělávání „jsou lidé přirozeně zvídaví, neboť se narodili do neznámého světa, jenž touží poznat, z čehož vyplývá neutichající touha učit se“[7]. Každý z nás zná radost, z objevení, poznání, pochopení něčeho nového. Je to stejná radost jako když malé dítě udělá svůj první krok. Je to radost plynoucí z poznání, objevení, naučení se něčemu novému. Náš mozek je pravděpodobně nastaven dávat nám tuto odměnu proto abychom byli vnitřně motivování dále zkoumat a poznávat okolní svět. Čím je tedy způsobeno to, že mají děti povětšinou spíše negativní vztah k prostředí školy, i když je to prostředí, ve kterém by měli zažívat radost z poznávání nového?

Lidé, kteří se dlouhodobě problematice vzdělávání věnují často tvrdí, že pojetí školy, tak jak je nyní nastaveno, nedostatečně motivuje k vlastnímu

objevování a nenapomáhá k pěstování vlastních dovedností[8], nebo že nám nedopřává radost z vlastního objevování. Namísto toho je stroze zaměřeno na přenášení znalostí. Přičemž nemá respekt k individualitě každého jedince a předkládá všem znalosti najednou, stejnou formou, tak nějak standardizovaně, jednotně. Žákům je tak jednak odpírána svoboda ve výběru toho jak, kdy a co se chtějí učit nebo lépe řečeno „poznávat“ a za druhé často není brán zřetel na potřeby každého žáka ve vzdělávání. Pan profesor Milan Hejný, dnes uznávaný pedagog, matematik, působící na Karlově univerzitě tvrdí: „Nerespektujeme zákonitosti poznávacího procesu, nerespektujeme, jak je každé dítě nastavené na učení se.“. Jinými slovy mluví o potřebě individualizovat namísto standardizovat vzdělávání. Dále pak pan profesor zmiňuje důležitost vlastního objevení poznatků. Znalosti převzaté si nikdy člověk nezapamatuje tak dobře jako ty, které přijme za své, z důvodu toho, že měl možnost na ně sám přijít a zažít si tak radost z vlastního objevení, pochopení.

Svoboda a důležitost vlastního poznávání ve vzdělávání byla v historii již mnohokrát zmiňována významnými osobnostmi. Např. Lev Nikolajevič Tolstoj na základě své pedagogické zkušenosti rozvinul názor: „...svoboda je nezbytnou podmínkou každého opravdového vzdělávání jak pro žáky, tak pro učitele, tzn. že jak hrozba tresty, tak slibování odměn, jež činíme předpokladem získávání těch či oněch vědomostí, nejenže nepřispívají k opravdovému vzdělávání, nýbrž ze všeho nejvíce mu překážejí.“[9].

Sir Ken Robinson ve svém (velmi populárním) projevu na TED konferenci mluví o tom, že škola přímo ničí kreativitu a touhu po poznání. A to z důvodu toho, že jsou žáci penalizování za chybu, přičemž reálně je chyba prostředkem k poznávání. Malé dítě udělá nespočet omylů, chyb na cestě k naučení se chodit. Chyba je v životě přirozeným prostředkem učení se.

Ke klasickému školství vzniklo a dnes existuje mnoho alternativ, které se snaží různým způsobem udělat vzdělávání pro děti systematicky přirozenější. Častými společnými prvky těchto alternativních přístupů jsou vysoká snaha o individualizaci, určité principy svobodného vzdělávání, společná diskuze a další. Uvedené principy jsou ale velmi náročné na osobu učitele. Zejména individualizace je velmi obtížná a s větším počtem žáků ve třídě téměř nemožná.

Jedná se například o známé Walsdorfské či Montesory školy ale i jiné. Jednou z alternativ ve výuce je i tzv. Hejného metoda výuky matematiky, která je, jak z názvu vyplývá, zaměřena na matematiku ale její principy se dají dle různých názorů využít obecněji i v jiných předmětech. Této metodě se u nás v posledních letech dostává veliké popularity.

Hejného metoda výuky matematiky

Takzvaná Hejného metoda výuky matematiky, které položil základy Vít Hejný (otec zmíněného prof. Milana Hejného), nabízí alternativu ke klasickému vzdělávání matematice na základní škole. Metoda je ve svém základu postavena na radosti z objevování, která dle autorů posiluje touhu po dalším poznávání. Děti učené touto metodou sami objevují matematiku za pomoci svého učitele, který ale neplní funkci vševědoucí autority, nýbrž předkládá problémy/úlohy žákům k řešení a následně ve třídě vede a moderuje diskuzi mezi žáky. Tím, se kromě matematiky, děti učí argumentovat a respektovat cizí pohledy na věc. Dochází tak mimo jiné k budování „soft-skills“.

Ač je metoda postupně vyvíjena již téměř 70 let, většímu rozšíření a popularizaci se jí dostalo až v posledních deseti letech. Za tu dobu se rozšířila do více jak 20 % základních škol. Každý rok se pomocí ní tak učí přibližně každý desátý žák první třídy v ČR. Metodě se pomalu daří pronikat i do zahraničí.

Metoda výuky je postavena tak aby u žáků byly při výuce budovány mentální modely a představy namísto učení se nazpaměť postupů vzorových problémů. Oproti klasickému přístupu nedochází k prezentaci matematických „pravidel“, nýbrž k jejich postupnému objevování a pochopení. Tomu napomáhá princip „prožívání“ výuky, kdy žáci ve třídě např. krokují a tleskají, „jezdí autobusem“, čímž se vžívají do problému a učí se ho si zřetelně představit. Tzv. „vzorečky“ se žáci proto neučí nazpaměť, naopak pomocí vhodného postupu, uvažování, prožívání a diskuze ve třídě si utvářejí vlastní představu o problému, který pak ve skupině řeší diskuzí. V této diskuzi pak děti hledají obecně správná řešení. Dle zkušeností učitelů si žáci ony „vzorečky“ na problémy pak často naleznou sami za účelem ulehčení si vlastní práce. Praktikování těchto principů ovšem není jednoduché a zvládne je pouze proškolený učitel, který má vlastní zájem tímto způsobem výuku vést.

Technologie a vzdělávání

S rozvojem moderních digitálních technologií, zejména pak internetu a mobilních zařízení se neustále otevírají ohromné možnosti napříč všemi dnešními obory. Propojení IT technologií s obory usnadňuje komunikaci, urychluje a zefektivňuje práci. Stejně tak je tomu v oblasti vzdělávání.

Využívání technologií ve vzdělávání (v masovějším měřítku) započalo na přelomu tisíciletí. Internet a digitální technologie od té doby v kontextu vzdělávání umožnili:

1. **Masivní dostupnost informací**
2. **Sdílení informací, propojení:** učitel-žák, žák-žák, učitel-učitel
3. **Interaktivita a škálovatelnost** (oproti učebnicím)
4. **Názorná vizualizace** - díky vizualizačním nástrojům můžeme sn8ze porozumět studovanému obsahu

V některých případech již dnes ale především do budoucna technologie dle mého názoru napomohou k:

1. **Individualizace vzdělávání** - díky rekomendačním systémům lze doporučovat vzdělávací obsahu na míru potřebám a progresu studenta
2. **Prožívání výuky/poznávání** - simulace prostředí, rozšířená realita

Dalším krokem zapojení technologií do vzdělávání je dle mého názoru vhodné využití rozšířené a virtuální reality. Vzdělávání za pomocí rozšířené reality nám v budoucnu umožní informace zažít takřka na vlastní kůži, tím, že umožní interakci s reálným prostředím, s reálnými předměty a nabídne nám vzdělávací obsah ve virtuální podobě. Vzdělávání tak bude spojeno s „reálným“ cítěním těla, což nám umožní nabité znalosti si ještě lépe a efektivněji zapamatovat.

Zapojení technologií do Hejného metody

Hejného metoda je postavena na 70 letech učitelské praxe, přičemž klade veliké nároky na osobu učitele. Praktikování principů Hejného metody ovšem není jednoduché a zvládne je pouze proškolený učitel. Technologie by v tomto případě mohli pomoci jak žákům, tak učitelům. Žáci se dostanou k atraktivnímu obsahu, který oproti učebnicím nabídne interaktivitu a velmi dobrou škálovatelnost. Učitelům technologie mohou umožnit lépe zvládnout individualizaci výuky a jiné.

Osobně se problematice zapojení technologií do vzdělávání Hejného metodou věnuji již 2 roky. Vzniká tabletová aplikace pro domácí použití, která

staví na principech Hejného metody. Jedná se o vzdělávací hru, ve které dítě postupně objevuje matematiku.

Jako jeden z dalších kroků zapojení technologií do vzdělání HM osobně považuji vhodné využití rozšířené reality. Rozšířená realita by mohla umožnit krom přehledné vizualizace matematických principů i jisté „prožívání“ poznávání. Představme si například děti učící se (nejen) matematiku venku v přírodě ve městě, kde poznávání probíhá za kombinace průchodem reálným prostředím a řešení virtuálních rébusů, jejichž řešení umožňují další postup v reálném prostředí. Takovéto zapojení všech smyslů a schopností vede na prožívání poznávání, což umožňuje dle Mgr. Simony Pekárkové „práci mozku na mnohem vyšší mentální úrovni, díky čemuž dochází k efektivnějšímu zapamatování si“.

Rozšířená realita

Jako pojem „rozšířená realita“ se běžně označuje reálný obraz světa doplněný o počítačem vytvořené objekty. Dle mého názoru se však nemusí nutně jenom jednat o počítačem upravený obraz. Realitu můžeme rozšiřovat jakýmkoliv uměle generovaným signálem. Krom obrazového signálu můžeme tak využít pro rozšíření reality i zvukových či tlakových signálů. Dále se ale budu v kontextu rozšířené reality zabývat pouze rozšířenou realitou v obraze pro využití na tabletových zařízeních, a to díky jejich dnešní dostupnosti.

V dnešní době je hojně diskutovaná otázka tabletů a obecně technologií v rukou dětí a ve vzdělávacím procesu. Jednou z často zmiňovaných nevýhod využití tabletů při vzdělávání malých dětí je fakt, že děti při práci s dotykovou obrazovkou tabletu nerozvíjí jemné motorické funkce, jako například při hraní si se stavebnicí. Z toho důvodu se záměrně tato práce ubírá směrem ovládání tabletu pomocí fyzických objektů, kdy na základě konfigurace těchto objektů na scéně, dochází k ovládání aplikace.

Vize práce

Smyslem této práce je učinit první krok směrem k vizi využití rozšířené reality na tabletových zařízeních při vzdělávání postaveném na principech Hejného metody výuky matematiky. Vizí práce je tabletová aplikace, která bude snímat okolí tabletu pomocí kamery, do kterého následně vykreslí virtuální grafiku reprezentující konkrétní úlohu z HM. Tento obraz bude následně zobrazen na display zařízení. Aplikace bude ovládaná interakcí s reálnými předměty v prostředí před tabletem.

Kapitola 1

Cíle práce

Jako cíl této práce jsem stanovil základně prozkoumat možnosti řešení rozšířené reality ve vzdělávání Hejného metodou a navrhnout jednoduchou vzdělávací aplikaci pro zařízení typu tablet. Aplikace bude využívat kameru tabletového zařízení pro snímaní okolí tabletu. Hlavním cílem práce je pro tuto zamýšlenou aplikaci nalézt vhodné workflow algoritmů a technik zpracování obrazu, které umožní ve snímaném obraze nalézt definované fyzické objekty, skrze které bude probíhat interakce se zařízením. To vše se záměrem tento algoritmický základ později proměnit ve vizi zamýšlenou aplikaci. Cílem práce proto není vyvinout hotovou aplikaci pro tabletová zařízení, nýbrž prozkoumat a připravit vhodné algoritmy pro rozpoznání definovaných objektů v okolí tabletu na základě obrazových dat kamery zařízení.

V práci se budu nejprve okrajově zabývat samotným návrhem zamýšlené aplikace s rozšířenou realitou. Především se však budu zabývat problematikou snímání okolí tabletu (scény), návrhem a ozkoušením vhodného algoritmického workflow zpracování obrazu. Těžiště práce tedy tkví v problematice zpracování, rozpoznávání a klasifikace objektů z obrazových dat.Tuto problematiku můžeme z pohledu procesu zpracování dat považovat za velmi blízkou oblasti tzv. „dataminingu“ (DM).

Obecně se v rámci DM procesů často objevují úkony, které jsou potřeba provést vždy ať se jedná o různé DM problémy. Zároveň se ukázalo, že je dobré mít každý DM proces dobře strukturovaný (standardizovaný) pro případy, kdy je potřeba na konkrétním DM procesu stavět či v něm pokračovat. Z těchto důvodu v minulosti vzniklo několik standardů, metodik, které napomáhají nastavit celý DM proces a jistým způsobem ho standardizovat.

Cílem této práce je připravit potřebné algoritmické workflow pro budoucí implementaci zamýšlené aplikace. Lze předpokládat, že při budoucí implementaci aplikace pro konkrétní platformy mohou nastat problémy, které budou mít dopad i na navržené algoritmy zpracování obrazu a že se tudíž bude třeba vrátit k jednotlivým experimentům a algoritmy příslušně upravit, optimalizovat. Z tohoto důvodu považuji za výhodné a dále si kladu za cíl, rámec vývojové

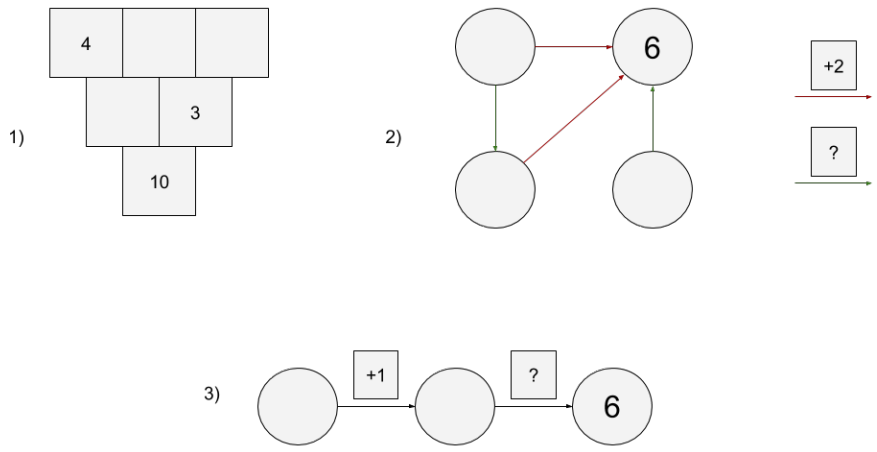
části této práce pojmout dle jedné z metodik určené pro DM procesy.

1.1 Zaměření práce v kontextu Hejného metody

Při výuce Hejného metodou se pracuje v tzv. didaktických prostředích. Tyto prostředí většinou vystihují nějaký reálný problém a jsou zaměřena na budování konkrétních dovedností žáka. Při úvahách o zapojení technologie do HM má proto smysl uvažovat o vhodnosti začlenění těchto jednotlivých prostředí.

Pro další účely této práce je potřeba se omezit na určitá prostředí. Proto jsem vybral prostředí „Součtové trojúhelníky“, „Hadi“ a „Pavučiny“ (viz obrázek 1.1), jelikož úlohy z těchto prostředí jsou snadno převoditelné do digitálního prostředí a jsou si z pohledu modelování velmi podobné. Jedná se totiž o grafové struktury, do kterých mají žáci za úkol hledat správná řešení pomocí doplňování čísel či operací.

Zamýšlená aplikace by ve finále měla zobrazit uživateli virtuální zadání úloh do obrazu reálného prostoru. Uživatel by pak pomocí rozmisťování čísel (v podobě fyzických žetonů) na stole, či hrací desce, s aplikací interagoval.



Obrázek 1.1: Ukázka úloh z prostředí „Součtové trojuhelníky“ (1), „Pavučiny“ (2), „Hadi“ (3)

Kapitola 2

Analýza problému

Závěrem úvodní kapitoly jsem popsal vizi prvního kroku zapojení rozšířené reality do vzdělávání HM, kterou je tabletová aplikace poskytující rozšířenou realitou ovládaná pomocí fyzických objektů. Dále jsem určil cíl této práce v přípravě algoritmů zpracování obrazu potřebných pro budoucí implementaci této zamýšlené aplikace.

V následující kapitole se nejprve hrubě zabývám funkční analýzou zamýšlené aplikace. Dále se pak věnuji problematice samotného používání a ovládání oné aplikace. Z těchto důvodů analyzuji možná umístění tabletového zařízení při používání aplikace a diskutuji výhody a nevýhody jednotlivých přístupů na základě čehož vybírám konkrétní řešení. Na závěr kapitoly definuji pojmy scéna a referenční zařízení, které jsou důležité a často používané ve zbytku práce, a zabývám se problematikou snímání scény zvoleného řešení.

2.1 Analýza základního fungování zamýšlené aplikace

Účelem zamýšlené aplikace je nabídnout uživateli matematickou úlohu ve virtuálním prostředí, kterou bude možné vyřešit pomocí interakce s reálnými předměty. Aplikace bude snímat pomocí kamery obraz okolí tabletu, do kterého vykreslí virtuální úlohu. Uživatel posléze manipulací s fyzickými objekty úlohu řeší. Na základě toho stanovuji základní funkční požadavky, které by finální aplikační řešení mělo splňovat:

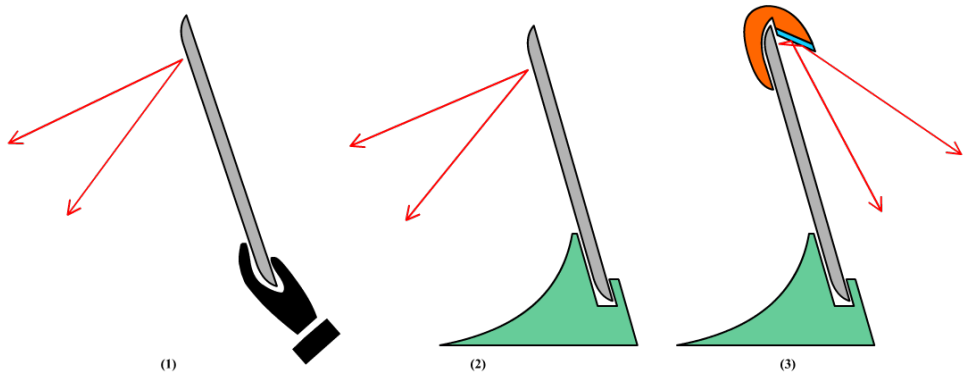
1. Aplikace je schopna vykreslit virtuální grafiku (zadání úlohy, dodatečná vizualizace, . . .) do obrazu reálné scény, který je snímán kamerou tabletu

2. Aplikace dokáže v obrazu scény rozpoznat předem definované fyzické objekty

2.2 Analýza používání zamýšlené aplikace

Před samotným návrhem je třeba míti na paměti i to, jakým způsobem uživatel bude výslednou aplikaci používat. Při řešení úlohy potřebuje mít uživatel možnost manipulovat s žetony, zároveň však potřebuje vidět snímanou scénu i skrze display zařízení aby viděl aplikací generovanou virtuální grafiku/úlohu. Z tohoto pohledu uvažuji, že se tabletové zařízení může nacházet v následujících polohách (viz obrázek 2.1):

1. Tablet je držen v ruce a snímá scénu pomocí zadní kamery z různých úhlů
2. Tablet je usazen ve stojanu a snímá scénu zadní kamerou
3. Tablet je usazen ve stojanu a snímá scénu frontální kamerou pomocí nasazeného klipu se zrcátkem



Obrázek 2.1: Varianty možných typů řešení umístění tabletu

Každá z výše uvedených variant má své výhody a nevýhody, které pro přehlednost shrnuji v následující tabulce:

2.3 Výběr varianty používání aplikace

Po uvážení všech výhod a nevýhod se mi jako vhodná varianta jeví varianta č. 3, a to především z důvodu uživatelské komfortnosti. Není totiž vyžadováno držení tabletu v ruce, uživatel tak má volnost při práci s aplikací, což je velmi důležité v případě, že má k interakci docházet pomocí fyzických objektů.

Další důležitá výhoda je fakt, že se scéna, prostor, kde dochází k manipulaci s objekty, nachází před samotným zařízením. Uživatel tudíž vidí zároveň reálnou scénu a obraz scény doplněný o virtuální prvky na obrazovce tabletu.

Jistou nevýhodou je nutnost přípravy modelů stojanu na tablet a zrcátkového klipu na frontální kameru tabletu. V dnešní době však díky technologii

Tabulka 2.1: Porovnání výhod a nevýhod jednotlivých přístupů umístění tabletu

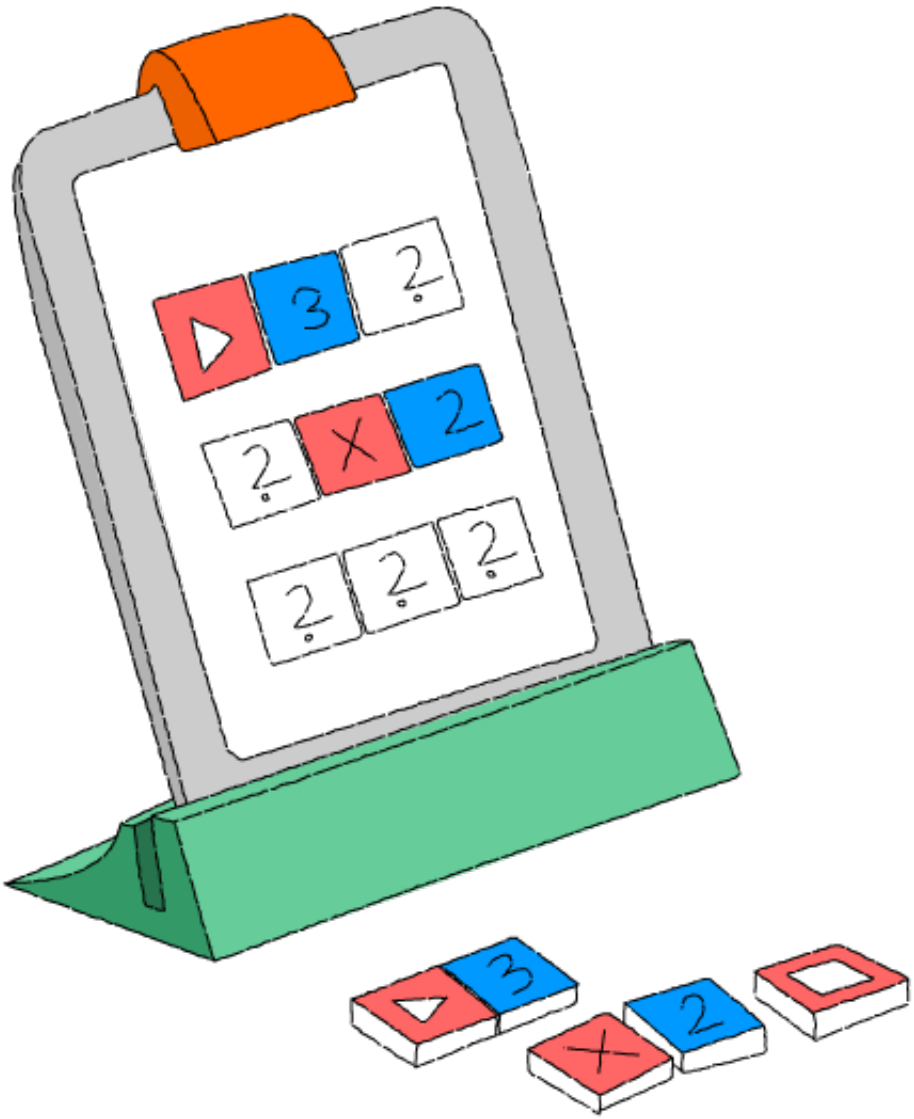
	Výhody	**Nevýhody**
1	obecnost - neomezená manipulace s tabletem	neustále měnící si poloha a směr kamery vůči desce stolu; nutnost držet tablet v ruce
2	uživatel má volné ruce; konstantí poloha kamery vůči desce stolu	nutná výroba stojanu; stojan uzpůsobený pro různé typy tabletů; scéna se nachází za tabletem; scéna je snímána pod velkým úhlem a je poměrně vzdálená od tabletu
3	uživatel má volné ruce; konstantí poloha kamery vůči desce stolu; scéna se nachází před tabletem	náročná a přesná výroba stojanu a klipu se zrcátkem; stojan a klip uzpůsobený pro různé typy tabletů

3D tisku není samotná výroba prototypu tak náročná. Proto tuto nevýhodu nepovažuji za zásadní.

Jako zásadní by se do budoucna mohl zdát nevýhoda nutnosti přizpůsobit modely stojanu a zrcátkového klipu pro různé typy tabletových zařízení. Nutnost nabízet pro každý typ tabletu různé varianty klipu se stojanem by mohlo bránit případnému šíření aplikace. Tento problém je ovšem v budoucnu do jisté míry řešitelný vhodným designem obou komponent.

2.4 Ovládání zamýšlené aplikace

Cílem zamýšlené aplikace je uživateli nabídnout ovládání (řešení) úloh pomocí fyzických objektů. Těchto objektů může být celá řada. V kontextu vzdělání Hejného metodou to závisí vždy na konkrétním didaktickém prostředí metody. Pro zvolená didaktická prostředí („Hadi“, „Pavučiny“, „Součtové trojúhelníky“) lze uvažovat o základních ovládacích prvcích jako o žetonech potištěných obecně různými symboly především pak numerickými. Aplikace tedy vykreslí do obrazu scény zadání úlohy. Uživatel se posléze snaží vyřešit úlohu vhodnou konfigurací dostupných žetonů se symboly v prostoru před tabletem (na snímané scéně).



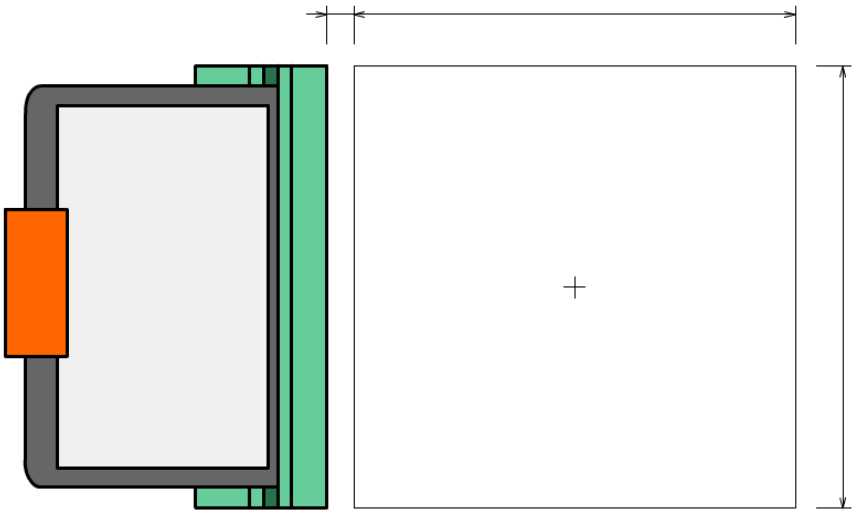
Obrázek 2.2: Nákres zvoleného řešení polohy tabletu

2.5 Definice scény

V předchozím textu jsem již několikrát užil slovo scéna ve smyslu prostoru, který snímá tablet pomocí kamery a ve kterém se odehrává interakce a manipulace s fyzickými objekty. Různé takové scény se ovšem od sebe mohou významně lišit. Například druhem osvětlení, dominující barvou scény, tvarem, velikostí apod. Proto je dobré se omezit a pro následnou práci pojem „scéna“ přesně zadefinovat.

Scénu definuji jako rovnou plochu (stolu, podlahy, ...), na které je prováděna interakce s aplikací pomocí fyzických objektů. Scéna je dále definována těmito omezeními:

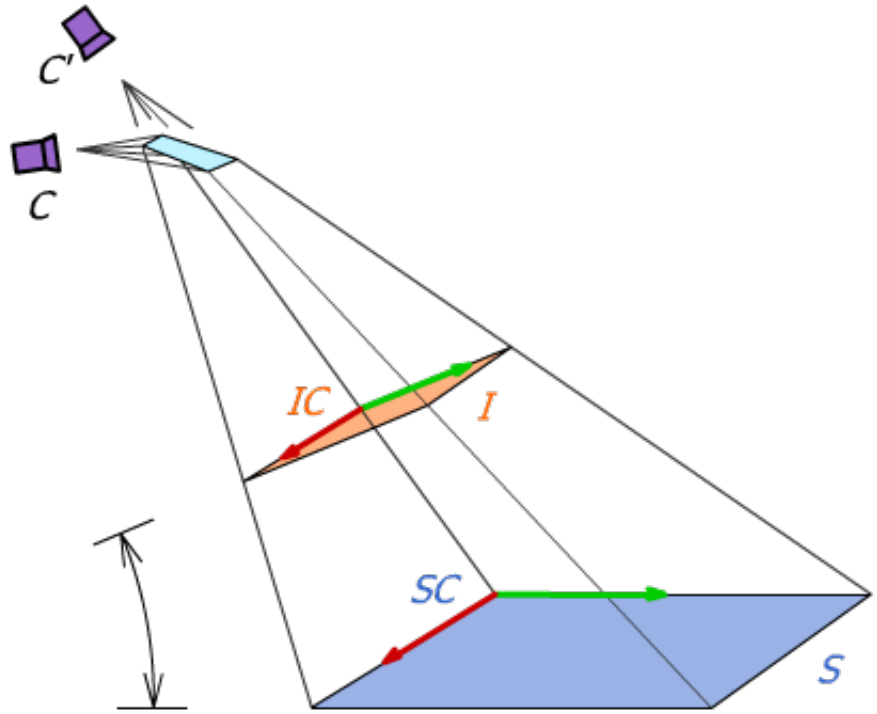
1. na barvě či textuře scény nezáleží
2. scéna je osvětlena dostatečně, tak aby obraz objektů ležících na scéně, nasnímaný kamerou zařízení byl jasně rozpoznatelný lidským okem
3. Tvar scény je definován pomyslným čtvercem o rozměrech 21 x 21cm, který je umístěn zarovnán ku tabletu ve stojanu dle obrázku níže
4. Všechny průsečnice rovin *plocha scény - plocha tabletu, plocha tabletu - plocha zrcátka, plocha zrcátka - plocha scény* jsou navzájem rovnoběžné



Obrázek 2.3: Nákres scény (pohled shora)

2.6 Analýza problému snímání scény

Kamerou snímaná obrazová data jsou výsledkem promítnutí obrazu scény na snímací chip (promítací rovinu) kamery tabletového zařízení. Jedná se o tzv. středové promítání (zobrazení euklidovského 3D prostoru na 2D rovinu), při kterém dochází k perspektivnímu zobrazení scény. Středové promítání má za následek zmenšování velikosti objektů v obraze, které leží na scéně dále od kamery oproti těm, které leží blíže.



Obrázek 2.4: Model snímání scény kamerou zařízení umístěného v sestavě

Sestavu stojan-tablet-zrcátkový klip lze modelově nahradit pouze kamerou s vhodně upravenou polohou a směrem. Potom celou sestavu a problematiku promítání a perspektivního zobrazení scény popisuje obrázek 2.4.

Na obrázku znázorněnou kameru C můžeme nahradit vhodně posunutou a natočenou kamerou C'. Obraz I nasnímaný kamerou má pak vlastní lokální souřadný systém IC, na kterém je vidět scéna S. Jelikož promítací rovina I kamery není rovnoběžná s plochou scény S, dochází k perspektivnímu zkreslování obrazů objektů ležících na scéně v závislosti na poloze objektu v rámci scény.

2.7 Volba referenčního zařízení

V dnešní době existuje celá řada tabletových zařízení s různými operačními systémy a různou hardwarovou konfigurací. Obecně při vývoji software pro mobilní zařízení je téměř nemožné odladit výsledný produkt tak aby fungoval bez problémů na všech platformách a HW konfiguracích. Finální implementace zamýšlené aplikace s sebou nese i problematiku optimalizace a adaptace algoritmů pro jednotlivé mobilní platformy a hardwarové konfigurace. To je ovšem mimo rámec této práce. Pro účely této práce je proto potřeba vybrat referenční zařízení na kterém budou prováděny veškeré experimenty a sběr dat.

Jako referenční zařízení jsem vybral tablet Samsung Galaxy tab S1 (první verze tabletu od Samsungu), Android verze 3.1. Tento model byl vybrán, jelikož byl dostupný a splňoval základní podmínky:

1. Dostatečné rozlišení přední a zadní kamery (alepoň 1600x1200 px)
2. Stáří modelu - model je starší více jak 5 let a proto je vhodný pro testování. Lze předpokládat, že pokud implementace zamýšlené aplikace obstojí na takto staré hardwarové konfiguraci, z hlediska výkonu obstojí i na dnes nových zařízeních podobné třídy

Je samozřejmé, že v případě implementace zamýšlené aplikace bude potřeba se více věnovat problematice nastavení správného testovacího prostředí vč. volby testovacích zařízení. Pro účely této práce je však zvolené referenční zařízení dostatečné.

Kapitola 3

Vývoj potřebných HW komponent

Z návrhu řešení zamýšlené aplikace v minulé kapitole vyplývá nutnost přípravy potřebných HW komponent. V rámci této kapitoly se zabývám především návrhem prototypu těchto komponent. Největší technickou výzvou je zde správné navržení stojanu pro tablet a zrcátkového klipu pro kameru zařízení, tak aby výsledná sestava umožnila kameře snímat celou scénu tak jak byla definovaná v analytické části.

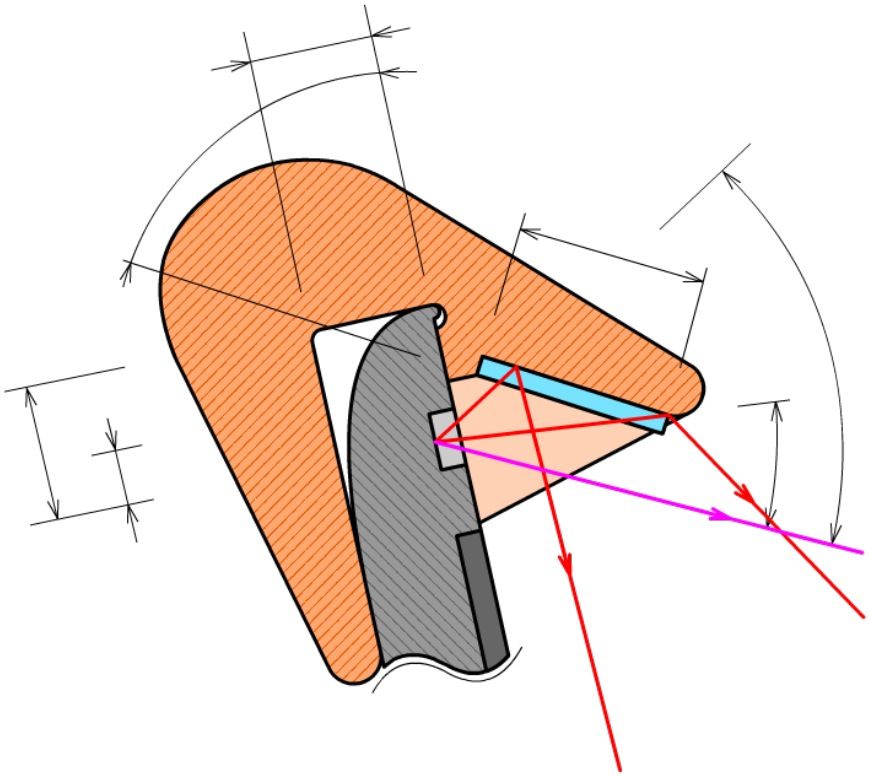
Během návrhu nejprve popisuji známé a neznámé parametry tabletu dané volbou referenčního zařízení. Posléze popisuji parametry sestavy, které je třeba navrhnout, a definuji na ně kladené požadavky. Dále popisuji postup návrhu jednotlivých komponent. Závěr kapitoly věnuji výrobě navržených komponent.

3.1 Návrh HW komponent

Zvolené řešení polohy tabletového zařízení pro zamýšlenou aplikaci pro svoji funkčnost vyžaduje realizaci dalších fyzických komponent. Jedná se o model stojanu pro tablet, zrcátkového klipu umístitelného na kameru tabletu a samotné žetony se symboly čísel. Finální design všech těchto komponent, který by byl dostatečně univerzální pro použití na různých zařízeních je kapitolou sám pro sebe. Proto jsem se prozatím zaměřil na návrh prototypů jednotlivých komponent, tak aby se daly využít společně se zvoleným referenčním zařízením.

Technické návrhy 3D modelů stojanu a zrcátkového klipu spolu silně souvisejí. Zároveň silně závisejí na parametrech zvoleného referenčního tabletu. Celá sestava stojan-tablet-zrcátkový klip musí být navržena tak aby umožňovala přední kameře tabletu snímat celou scénu. Při navrhování klipu a stojanu se lze setkat s několika proměnnými, které jsou odvislé od fyzických parametrů tabletu a parametrů jeho kamery. Celou problematiku popisuje následující ná-

črt (viz obrázek 3.1), na kterém je znázorněn řez sestavou tablet (tmavě šedá) - klip (oranžová) se zrcátkem (modrá).



Obrázek 3.1: Nákres řezu zrcátkovým klipem umístěným na frontální kameře tabletu

Popis proměnných parametrů daných konkrétním zařízením:

1. parametr a „tloušťka tabletu“ je obecně různá pro každý tablet
2. parametr k „vzdálenost frontální kamery od hrany tabletu“ je taktéž specifická pro každé zařízení, navíc tato hodnota není dostupná z technického popisu konkrétního zařízení, je jí proto potřeba přesně změřit
3. parametr s „úhel pohledu kamery“ lze dohledat v rámci technické specifikace; ovšem nikde není zaručeno, že je snímací chip kamery umístěn přesně rovnoběžně k ploše obrazovky tabletu

Popis navrhovaných parametrů definující tvar zrcátkového klipu:

1. parametr g „nevyužitý úhel pohledu kamery“ určuje míru ztráty rozlišení obrazu snímané scény ve vertikálním směru; při návrhu je dobré tento úhel minimalizovat;

2. Na míru ztráty obrazu scény má také veliký dopad parametr *k2* „vzdálenost úpatí zrcátka od kamery“. Ideální hodnota parametru *k2* je ovšem odvislá od parametru k konkrétního zařízení tudíž se napříč různými tablety bude drobně měnit.

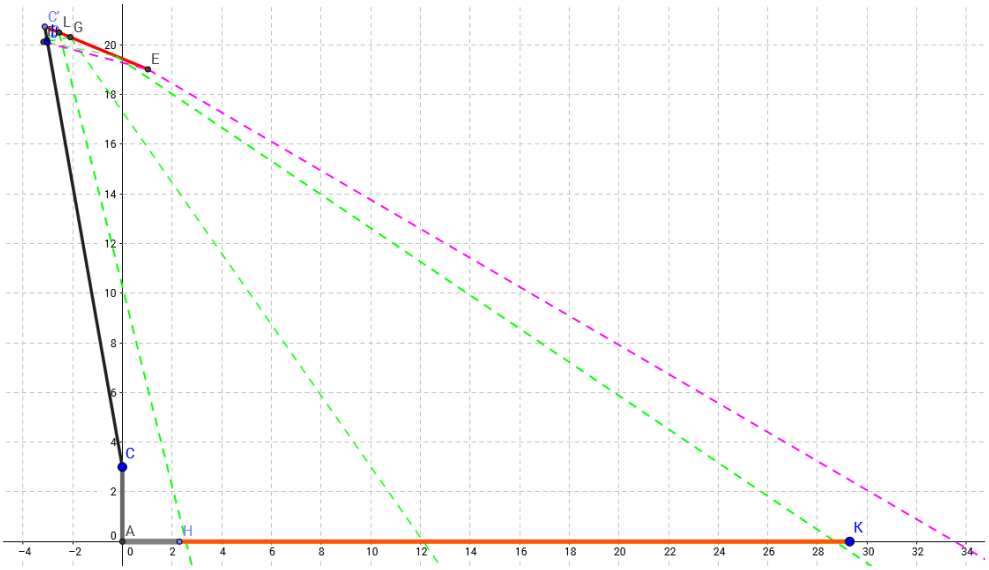
3. parametr u „úhel svírající zrcátko s obrazovkou tabletu“ je potřeba navrhnout tak aby kamera mohla skrze zrcátko snímat celou scénu; tento parametr zároveň velmi ovlivňuje parametr g

4. parametr z „délka zrcátka“ je potřeba navrhnout tak aby společně s parametrem u zajistil kameře snímání celé scény; snahou je ovšem délku zrcátka minimalizovat jelikož čím větší je zrcátko tím snáze dojde ke znečištění, zároveň by bylo dobré z praktických důvodů aby byl celý klip co nejmenší (spotřeba materiálu při výrobě, manipulace při práci s tabletem, . . .)

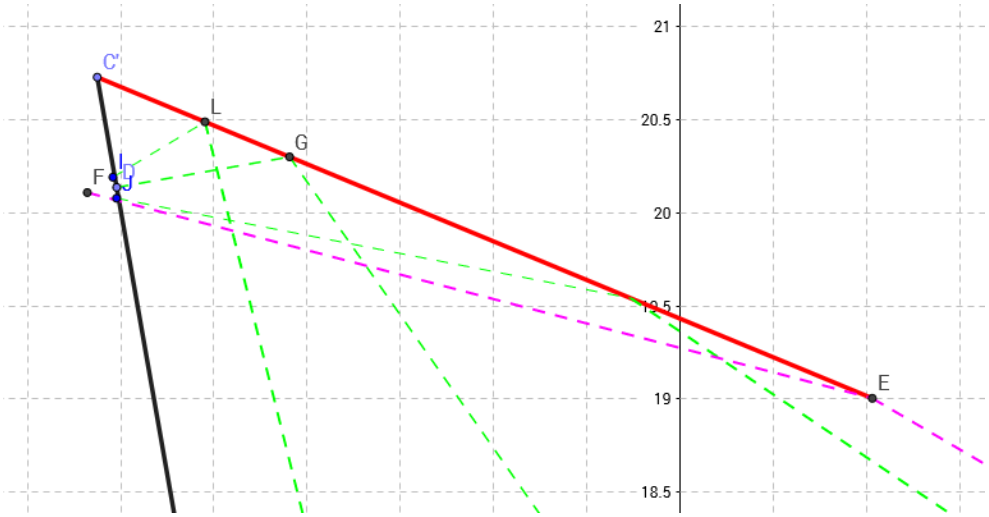
Pro účely návrhu správných parametrů klipu a stojanu je potřeba často měnit a experimentovat s hodnotami jednotlivých parametrů. Rozhodl jsem se proto za účelem zefektivnění procesu návrhování problém namodelovat v nástroji GeoGebra, který je určen pro vizualizaci matematických funkcí, vykonání jednoduchých matematických výpočtů, modelování a geometrii. Nástroj mi umožnil postupně najít vhodné parametry modelu, díky nímž ve výsledku celá sestava splňuje základní podmínku, kterou je záběr celé scény kamerou.

V modelu na obrázku 3.2 figurují všechny důležité části sestavy (stojan, tablet, zrcátko) a jim příslušné parametry. Dále je v modelu znázorněna oranžově i definovaná scéna a čárkovanými čárami rozsah pohledu kamery .

Na základě nalezených parametrů v nástroji Geogebra jsem přistoupil k samotnému modelování všech komponent. Pro tvorbu 3D modelů byly využity CAD nástroje NX Unigraphics a Inventor.



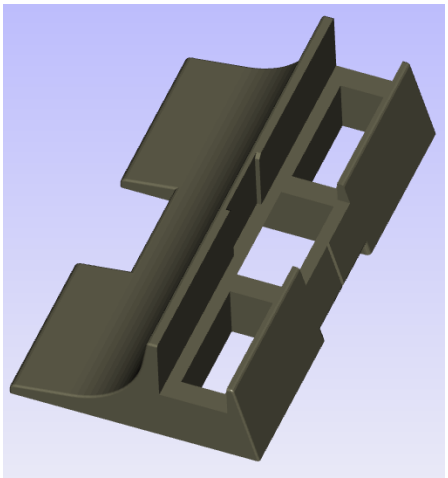
Obrázek 3.2: Model celé sestavy v nástroji Geogebra



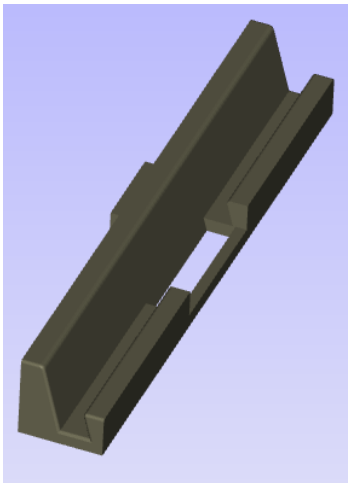
Obrázek 3.3: Detail modelu zrcátkového klipu v rámci celé sestavy v nástroji Geogebra

3.1.1 Návrh modelu stojanu

3D Model stojanu byl z praktických důvodů rozdělen na 2 části: tělo a vložku. Vložka určuje tloušťku a sklon tabletu ve stojanu a proto je praktické aby byla snadno vyměnitelná a nahraditelná za jinou. Pro případné experimentování s jinými tablety lze vyměnit pouze jen vložku a není potřeba vyrábět celé tělo stojanu znovu. Náhledy finálních modelů lze vidět na obrázcích 3.4a a 3.4a.



(a) Model těla stojanu



(b) Model vložky

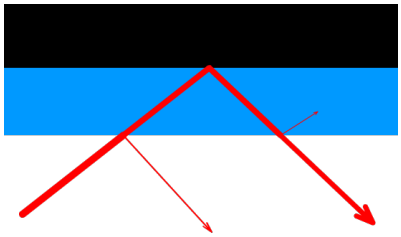
3.1.2 Návrh modelu stojanu

Při návrhu modelu zrcátkového klipu bylo potřeba řídit se krom nalezených parametrů i zvolením vhodného zrcátka a vytisknutelností samotného modelu na 3D tiskárně. Klip je obecně možné osadit třemi typy zrcátek:

1. Klasické skleněné zrcátko s tloušťkou skla 2 nebo 4mm
2. Leštěná nerezová ocel
3. Chromovaná ocel

Při použití skleněného zrcátka dochází k dvojodrazu paprsku světla na rozhraní sklo-vzduch a sklo-kov (viz obrázek 3.5), což má za následek mírné rozostření obrazu ve vertikálním rozměru obrazu. Čím silnější je skleněná vrstva, k tím výraznějšímu rozostření dochází. Proto je vhodnější použít zrcátko s jednou odrazivou vrstvou.

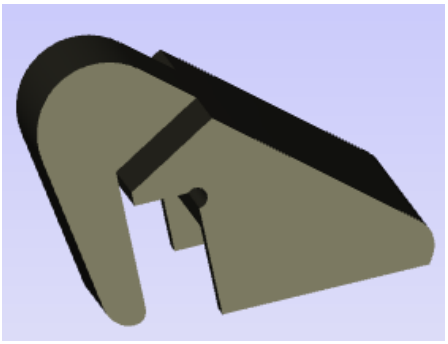
Použití leštěné nerezové oceli se však ukázalo jako nevhodné pro její odrazivost. Obraz v nerezovém zrcadle byl na první pohled matný a rozostřený. Jako ideální se proto jeví použití chromované oceli, kde nedochází k viditelnému rozostření. Bohužel na míru vytvořené chromované zrcátko pro model



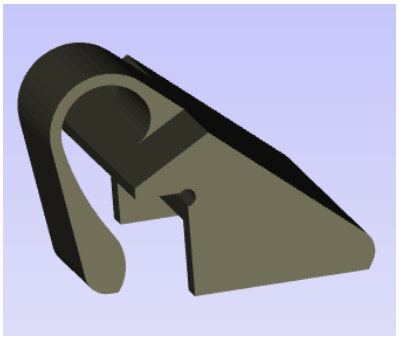
Obrázek 3.5: Detail modelu zrcátkového klipu v rámci celé sestavy v nástroji Geogebra

klipu bylo nedostupné a tak jsem zvolil lehce dostupnou variantu skleněného zrcátka o tloušťce 4mm.

Model zrcátkového klipu jsem navrhl ve dvou provedeních. Jeden model pasující na fixní šířku referenčního tabletu (viz obrázek 3.6a) a druhý experimentální model, určený pro proměnou tloušťku tabletu (viz obrázek 3.6a).



(a) Klip pro fixní tloušťku tabletu



(b) Experimentální klip pro různé tloušťky tabletů

3.1.3 Návrh modelu žetonu

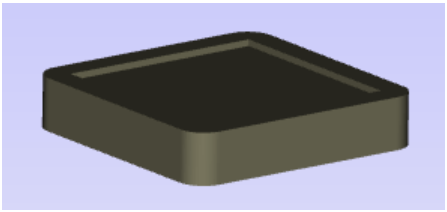
Při návrhu modelu žetonu bylo potřeba určit několik základních parametrů:

1. tvar a dostatečná velikost pro snadnou manipulovatelnost
2. barevnost
3. font použitý pro numerické znaky

Tvar žetonků jsem zvolil jako čtvercový se zaoblenými rohy proto aby mohli být žetony kladeny bezprostředně vedle sebe. Každý žeton má barevný střed s na něm umístěným bílým symbolem a bílým ohraničením, které slouží pro jasné oddělení středu žetonu od okolního obrazu. Pro první experimentování jsem zvolil font *Arial Rounded MT Bold*.

Výsledný model je navržen včetně „okraje“ (viz obrázel 3.7b). Barevný obsah žetonku se symbolem je natištěn na papír a následně nalepen na model žetonku.

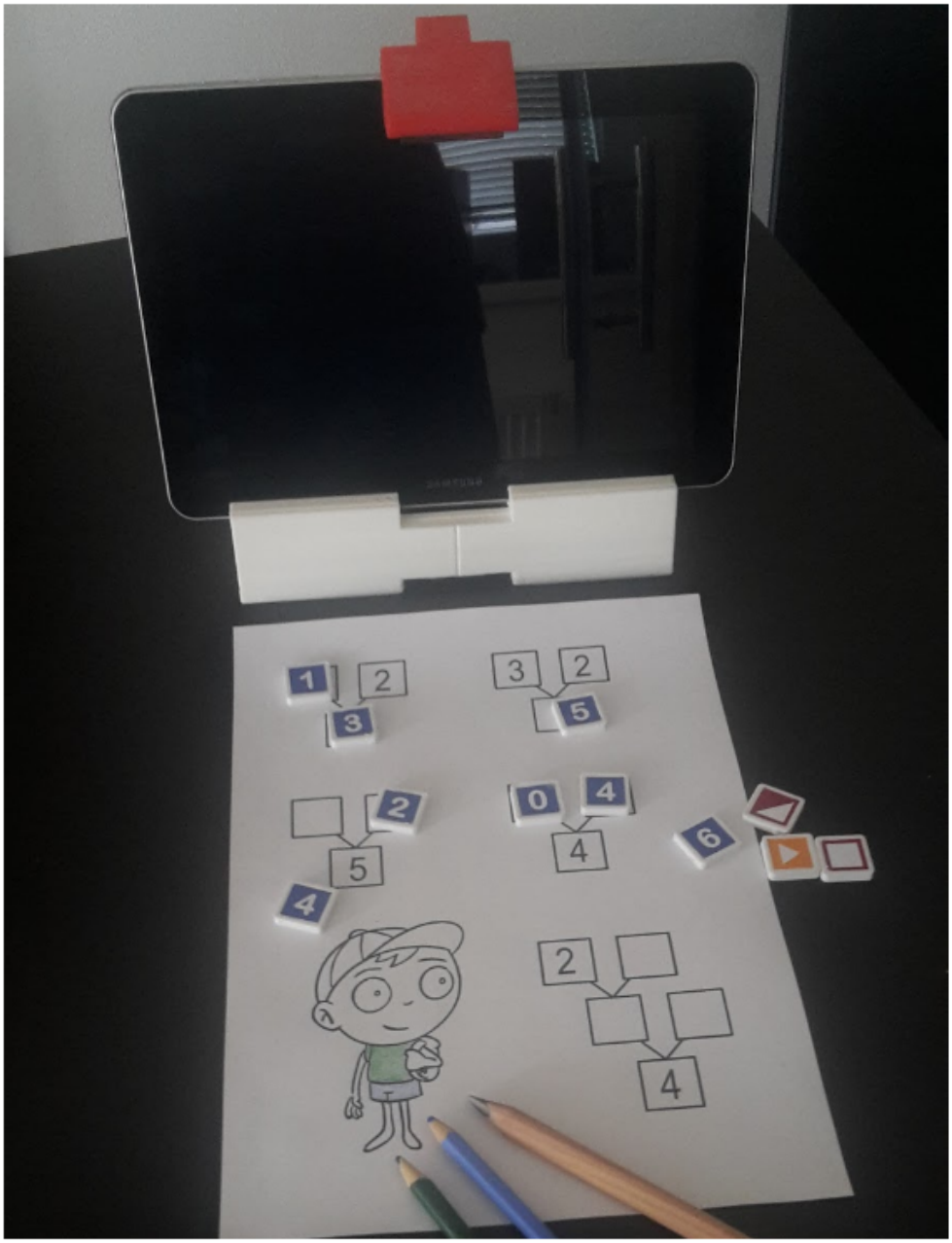
(a) Barevný návrh žetonu



(b) 3D model žetonu

3.2 Výroba komponent

Všechny komponenty (zrcátkový klip, stojan a vložka na tablet, žetony) byly vyrobeny na základě namodelovaných modelů v CAD nástrojích pomocí technologie 3D tisku na tiskárnách FIT ČVUT. Zrcátkový klip byl následně osazen skleněným zrcátkem o tloušťce 4mm, které je dostupné v každém sklenářství. Jednotlivé žetony byly polepeny fóliovým papírem, který byl potištěn příslušnými symboly. Celou sestavu stojan-tablet-zrcátkový klip s přiloženými žetony znázorňuje obrázek 3.8.



Obrázek 3.8: Realizovaná sestava tablet - stojan - zrcátkový klip, žetony

Kapitola 4

Vývoj SW dle CRISP-DM

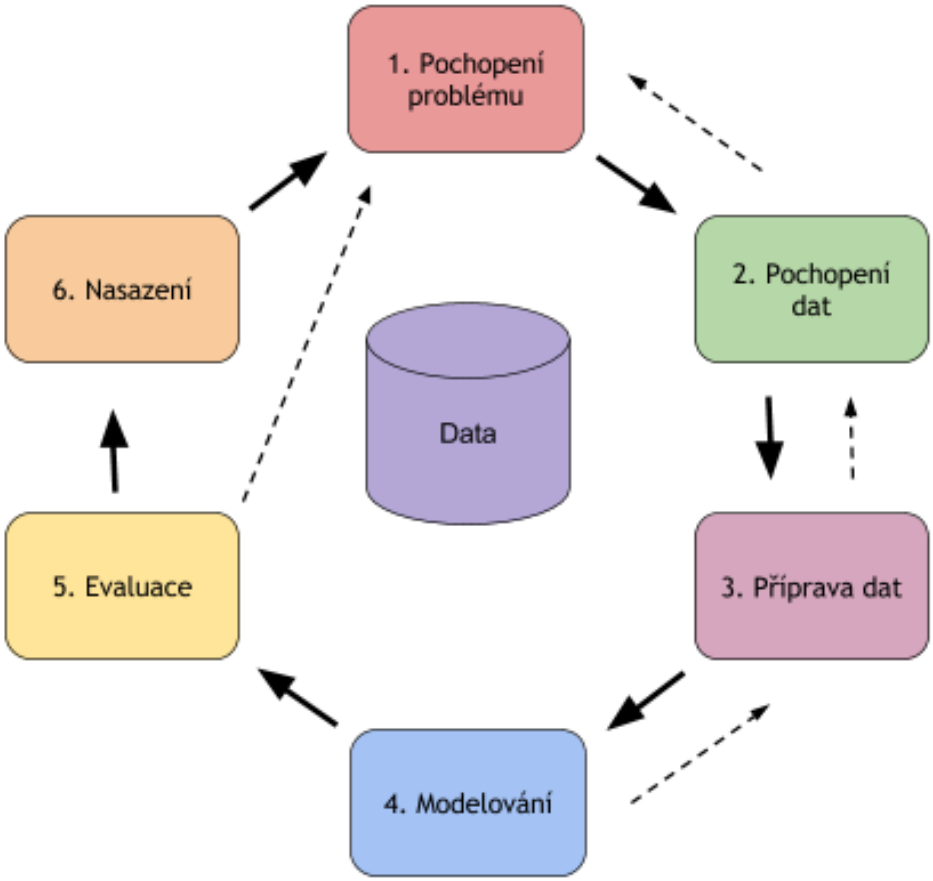
Následující kapitola je těžištěm celé práce. Zabývám se zde hledáním kýženého algoritmického workflow zpracování obrazu pro zamýšlenou aplikaci. Postupně popisuji jak jsem při vývoji posloupnosti algoritmů postupoval. V úvodní kapitole jsem při definování cílů práce stanovil jako jeden z cílů pojmout proces vývoje SW dle metodiky určené pro DM proces. Dnes existuje několik takových metodik, pro následnou práci jsem vybral metodiku CRISP-DM. Důvodem výběru této metodiky byla vlastní dobrá zkušenost a fakt, že je dle mnoha zdrojů CRISP-DM jedna z nejpoužívanějších metodik pro DM procesy.

Metodika CRISP-DM („Cross industry standard process for Data mining“) definuje posloupnost kroků, které je obvykle potřeba vykonat při průchodu DM procesem. Model metodiky je postaven na základě osvědčených postupů a umožňuje projít celým DM procesem efektivněji. Díky metodice proto bude práce vhodně strukturovaná a bude na ni možno případně snáze navázat. Základní kroky metodiky CRISP-DM znázorňuje obrázek 4.1.

V rámci DM procesu je potřeba často experimentovat. Jednotlivé experimenty přinášejí většinou nový pohled na skutečnost, často je na základě jejich výsledků potřeba upravit předpoklady (dochází k postupnému pochopení problému), pořídit nová data pro zkoumání nově objevených jevů, pochopit nová data, namodelovat je apod. CRISP-DM je proto navržen jako cyklický proces přičemž cykly v něm mohou probíhat na několika úrovních.

Krom hlavního cyklu (na obrázku 4.1 naznačen tlustými čárami) se v procesu typicky vyskytují i menší zpětné cykly (naznačeny čárkovanými čárami). Při modelování například dojde ke zjištění, že model nefunguje optimálně na určitých datech. Proto je třeba se vrátit zpět a připravit nová data lépe zahrnující „problémové jevy“. Díky tomu je často potřeba vrátit se až k bodu pochopení nových dat.

Poznámka: Stejným způsobem probíhal i vývoj SW v rámci této práce. Často bylo potřeba vrátit se a nasnímat nová data, pochopit je, předzpracovat a na základě toho vylepšit model a celý algoritmus. Cyklický proces se však obtížně převádí do lineární podoby, kterou má tato práce dánu ze zadání.



Obrázek 4.1: Realizovaná sestava tablet - stojan - zrcátkový klip, žetony

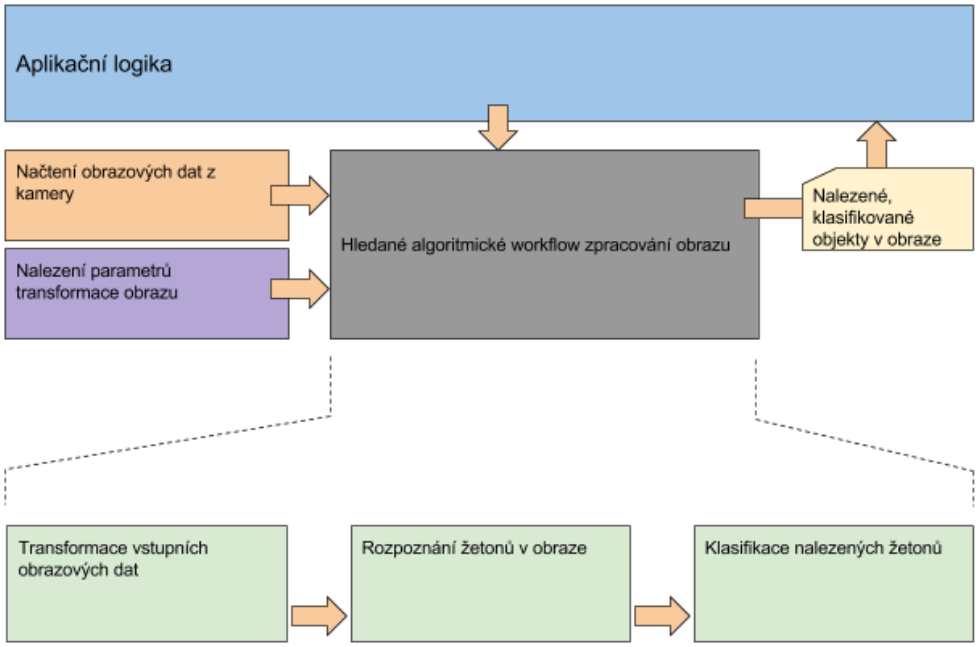
Rozčlenil jsem proto tuto kapitolu do podkapitol dle posloupnosti základních kroků CRISP-DM. Jednotlivé sekce podkapitol však během vývoje nenastaly nutně chronologicky za sebou. V textu zanechávám zmínku všude tam, kde bylo potřeba, se na základě nalezených zkušeností vrátit o krok či dva zpět a provést je opakovaně.

4.1 Pochopení problému

V rámci této kapitoly se zamýšlím nad navrženou aplikaci z hlediska problematiky DM procesu. Definuji a ohraničuji DM problém a stanovuji kritéria úspěchu řešení. Na závěr zhodnocuji aktuální situaci a zabývám se výběrem vhodných nástrojů pro realizaci práce.

4.1.1 Analýza DM problému v kontextu zamýšlené aplikace

Technickým problémem respektive cílem DM procesu je nalezení definovaných objektů v obraze a jejich následná klasifikace. V práci se na základě kapitoly „zaměření práce v kontextu Hejného metody“ dále omezuji na hledání žetonů potištěných čísly 0-9, jelikož jsou to nejdůležitější objekty pro zvolená didaktická prostředí HM. Na celý problém lze nahlížet detailněji v rámci kontextu fungování celé zamýšlené aplikace a rozložit jej do řady podproblémů, funkčních bloků. Tyto funkční bloky a jejich posloupnost v kontextu zamýšlené aplikace naznačuji v obrázku 4.2.



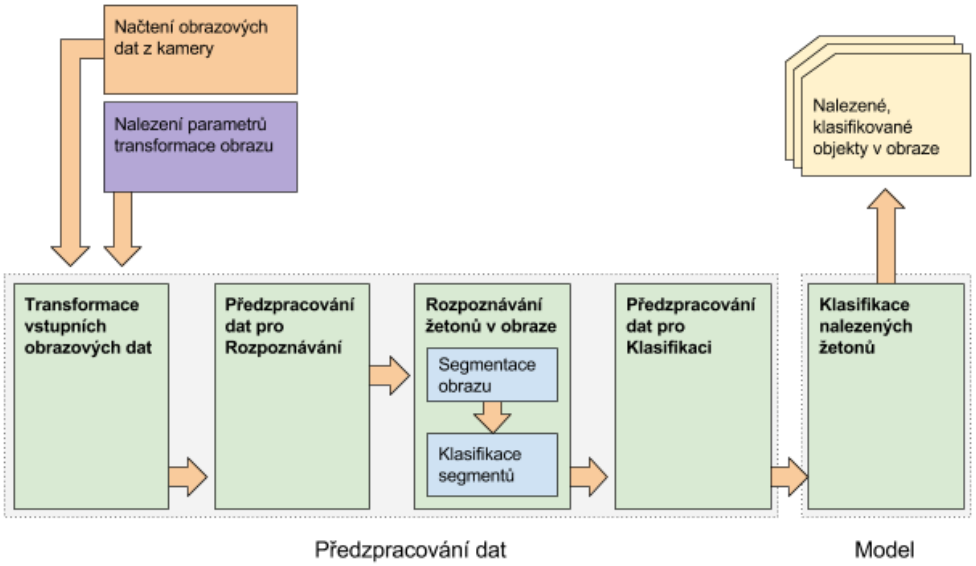
Obrázek 4.2: Hledané algoritmické workflow zpracování obrazu v kontextu celé aplikace

Na obrázku 4.2 lze vidět základní představu hledaného algoritmické workflow zpracování obrazu (cíl této práce) složené ze tří funkčních celků (podbarveny zeleně). Modře podbarvená část funkcionality se týká implementace logiky aplikace a samotné aplikace pro konkrétní platformu (OS), která není cílem této práce. Cílem této práce je připravit funkcionální základ (nižší vrstvu zamýšlené aplikace), který bude využíván vyšší vrstvou aplikační logiky.

4.1.2 Návrh algoritmického workflow

Na základě předchozí analýzy lze algoritmické workflow rozdělit podrobněji do základních funkčních bloků, které znázorňuji na obrázku 4.3. Jednotlivé funkční bloky zde tvoří posloupnost zpracování obrazových dat, výstup prv-

ního je vstupem druhého atd. až je výsledkem seznam nalezených a klasifikovaných žetonů. Celou posloupnost lze také pomyslně rozdělit na část předzpracování dat a část reprezentující model. Dále se zabývám popisem účelu jednotlivých funkčních bloků.



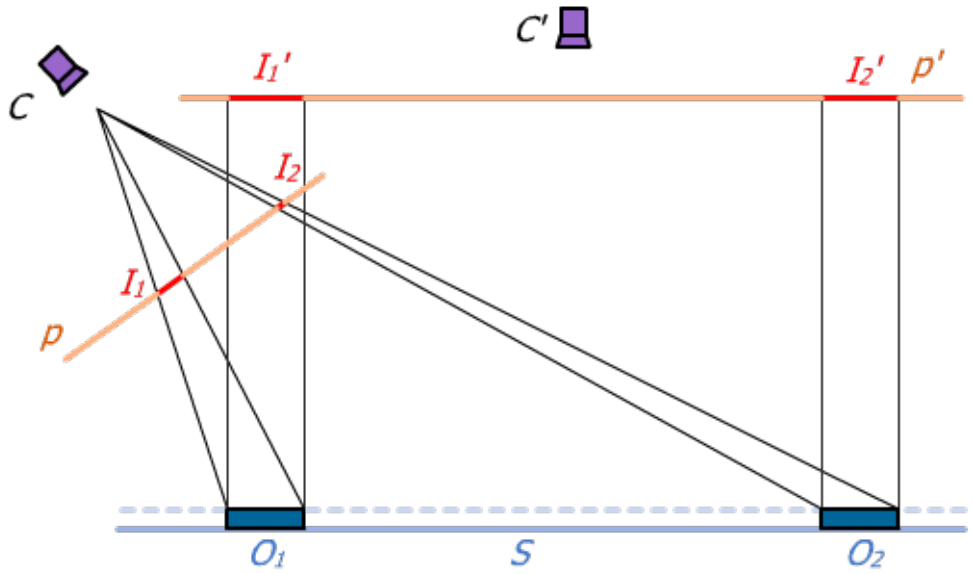
Obrázek 4.3: Funkční bloky hledaného algoritmického workflow

4.1.2.1 Transformace vstupních obrazových dat

V sekci 2.6 *Analýza problému snímání scény* jsem popsal středové neboli perspektivní promítání obrazu scény na chip kamery zařízení, který má za následek zmenšování obrazů objektů nacházející se na scéně dále od kamery. Aby byla následná úloha rozpoznání žetonů v obraze snazší, je vhodné snímaný obraz nejprve transformovat tak aby došlo k odstranění perspektivního zkreslení žetonů. Tím se velikost a tvar různě rozmístěných žetonů na scéně sjednotí a v rámci úlohy rozpoznávání žetonů v obraze se bude stačit zaměřit na objekty jednotné velikosti.

Zkreslování velikosti objektů na scéně vlivem perspektivního promítání kamery znázorňuje obrázek 4.4. Kamera C v modelu nahrazuje celou sestavu stojan - tablet - zrcátkový klip. Na scéně S jsou umístěny dva objekty stejné velikosti (reprezentující žetony) $O1$ a $O2$ jejichž obrazy $I1$ a $I2$ jsou promítnuty na promítací rovinu (chip) p kamery C. Na obrázku lze vidět různé zkreslení velikosti obrazu $I1$ a $I2$ na promítací rovině p v závislosti na pozici objektu na scéně. Tohoto zkreslení bychom se rádi z již popsaných důvodů zbavili. Jednou z možností jak toho docílit je transformovat pořízený obraz tak aby odpovídal pohledu „shora“ v ortogonálním promítání.

Při pohledu na scénu *S* (shora) při ortogonálním promítání (speciální případ rovnoběžného promítání), kdy je promítací rovina *p'* kamery *C'*, rovnoběžná s rovinou scény *S*, ke zkreslení velikosti objektu v závislosti na poloze objektu na scéně nedochází. Obrazy *I'1* a *I'2* budou mít vždy stejnou velikost ať se objekty *O1* a *O2* budou na scéně nacházet kdekoliv.



Obrázek 4.4: Model projektivního a ortogonálního promítání scény

Z obrazu scény *I* ovšem nelze zrekonstruovat celou scénu. Promítání z euklidovského 3D prostoru scény na 2D rovinu není prosté zobrazení, proto k němu ani neexistuje inverzní zobrazení. Z tohoto důvodu není možné získat přesný obraz scény při pohledu shora v ortogonálním zobrazení. To ovšem pro účely sjednocení velikosti žetonů nevadí. Nepotřebujeme totiž rekonstruovat scénu ve všech třech rozměrech.

O žetonech víme, že mají všechny stejné rozměry a že potisk/symbol, který nesou, se nachází na horní straně žetonu. Jde nám pouze o odstranění zkreslení žetonů. Z tohoto důvodu lze problém promítání zjednodušit pouze na zobrazení 2D roviny povrchu žetonů (v obrázku 4.4 naznačena čárkovaně) na promítací 2D rovinu chipu kamery. V tomto případě je již promítání zobrazení prosté a lze tedy teoreticky nalézt originální obraz scény.

Provedením správné inverzní transformace pořízeného obrazu kamerou dostaneme obraz scény v ortogonálním promítání. Ve zbytku práce se nebudu hledáním a odvozováním této transformace zabývat, jelikož tuto problematiku typicky lze řešit pomocí funkcionality frameworků pro implementaci rozšířené reality, ke kterým se ještě dostanu v této kapitole později.

Je nutno poznamenat, že transformace obrazu, kterou provádí zrcátkový

klip, se může (a pravděpodobně bude) drobně měnit za prvé při běhu aplikace ale především za druhé v rámci odlišných spuštění aplikace. Může k tomu dojít například následkem uživatelské manipulace s celou sestavou. Usazení tabletu ve stojanu a zároveň usazení zrcátkového klipu na tabletu je totiž provedeno s drobnou vůlí a proto není a ani nemůže být dokonale přesné. Po odlišných usazeních tabletu a spuštění aplikace bude proto typicky zrcátkový klip transformovat obraz scény lehce odlišně. Při vývoji algoritmických modelů je proto potřeba na tento jev pamatovat a vyvinout je dostatečně robustní.

4.1.2.2 Předzpracování dat pro rozpoznávání a klasifikaci

Před zpracováním obrazových dat funkčním blokem *Rozpoznávání* respektive *Klasifikace* je obvykle potřeba vstupní data nejprve vhodným způsobem předzpracovat. V případě zpracování obrazu jsou vstupní data typicky silně ovlivněna vlivy prostředí, ve kterém se nachází snímaná scéna. Jedná se například o různorodost osvětlení napříč různými scénami.

Funkční bloky Předzpracování dat pro *Rozpoznávání* resp. *Klasifikaci* hledaného algoritmického workflow mají za účel odstranit případné různé vlivy prostředí (jejichž analýze se věnuji v kapitole **??** *Pochopení dat*). Dalším účelem je vstupní data připravit do podoby, formátu vhodného pro následné funkční bloky *Rozpoznávání* a *Klasifikace*. Výstupem jsou tedy v obou případech vhodně předzpracovaná obrazová data.

4.1.2.3 Rozpoznávání žetonů v obraze

Cílem tohoto funkčního bloku je nalézt ve vstupním obraze místa na scéně, kde se nacházejí žetony. Rozpoznávání žetonu by mělo být dostatečně robustní a odolné vůči různému natočení žetonů.

Vstupem funkčního bloku *Rozpoznávání* je transformovaný a předzpracovaný obraz scény Itp. Výstupem tohoto bloku je pak seznam nalezených žetonů v podobě informací o „bounding boxech“ jednotlivých žetonů v obraze Itp.

Problém *Rrozpoznávání žetonů* v obraze lze rozdělit na dva podproblémy, které jsou naznačeny v obrázku 4.3. Nejprve je potřeba v obraze nalézt segmenty, které jsou svými rozměry a tvarem podobné žetonům. Tyto segmenty je poté potřeba filtrovat. Klasifikovat, které doopravdy žetony jsou a které nikoliv.

4.1.2.4 Klasifikace žetonů

Účelem funkčního bloku Klasifikace žetonů je přiřazení tříd k nalezeným žetonům dle hodnoty jejich symbolu. Žetony jsou klasifikovány dle hodnoty symbolu do deseti tříd *0,1,2..,9*.

Vstupem bloku je předzpracovaný obraz Itp a seznam nalezených žetonů, což je výstup předchozího bloku Rozpoznávání. Výstupem tohoto bloku je pak seznam nalezených žetonů obohacený o příslušnost žetonu k dané třídě.

4.1.3 Definice kritérií úspěchu

Hlavním kritériem úspěchu je nalezení takové posloupností technik zpracování obrazu v rámci jednotlivých funkčních bloků, které povede na získání požadovaného výstupu z definovaných vstupů. Nalezený algoritmus kromě toho musí splňovat následující podmínky:

1. **Robustnost vůči různým druhům osvětlení** - Předpokladem je dostatečně osvětlená scéna, což znamená, že hledané objekty budou bez potíží a zřetelně rozpoznatelné zdravým lidským okem.
2. **Úspěšnost klasifikace žetonů větší jak 95%** - Úspěšnost klasifikace záleží silně na vstupních datech. Lze předpokládat, že čím dále od kamery se žetony nachází a čím menší je snímací úhel kamery, tím slabší bude vlivem ztráty přesnosti úspěšnost klasifikace. Proto pro měření úspěšnosti se omezuji na objekty vyskytující se pouze na definované scéně.
3. **Úspěšnost nalezení žetonů v obraze větší jak 95%**

4.1.4 Zhodnocení aktuální situace

Před započetím hledání konkrétních technik pro navržené algoritmického workflow je potřeba nejprve zhodnotit aktuální situaci a vybrat pro následný vývoj vhodné nástroje či frameworky. V této podkapitole se proto zabývám dostupnými nástroji a frameworky, které jsou vhodné pro realizaci experimentů s daty a algoritmy v rámci DM procesu. V případě frameworků provádím základní průzkum funkcionalit a zhodnocení několika dostupných řešení.

4.1.4.1 Dostupné a vhodné nástroje

V dnešní době existuje již celá řada různých nástrojů a knihoven, které lze v rámci DM vhodně využít. Aby proces vývoje zamýšlené aplikace byl efektivní je vhodné oddělit proces experimentování a hledání vhodného algoritmu od samotné implementace pro určitý operační systém.

Ideální prostředí pro experimentování jsou takové, ve kterých lze rychle provádět experimenty, tzn. nabízejí skriptovací rozhraní, ve kterém je zároveň dostupno co nejvíce již předpřipravených algoritmů z oblasti zpracování a modelování dat. Pro účely práce jsem předvybral dnes dostupné a pravděpodobně vhodné nástroje:

1. Matlab + Computer Vision System toolbox, image processing toolbox, neural net toolbox
2. IPython + scikit-image + sciki-learn toolbox
3. Wolfram Mathematica

Při finálním výběru jsem posuzoval dostupnost, vybavenost a míru dokumentace konkrétního nástroje. Důležitou roli pak hrála také vlastní zkušenost. Na základě zmíněných kritérií jsem se proto rozhodnul pro experimentování v rámci DM procesu využít prostředí Matlab s příslušnými toolboxy. Matlab je pro mé účely dostupný díky univerzitní licenci zdarma, obsahuje mnoho potřebné funkcionality a jelikož je to jinak komerční software, vyniká také dobrou a přehlednou dokumentací, stejně tak širokou komunitní podporou.

Pro samotnou implementaci zamýšlené aplikace je vhodné využít jeden z frameworků určených pro rozšířenou realitu. Ikdyž implementace aplikace jde za rámec této práce, je vhodné mít základní přehled o možnostech dnešních knihoven a frameworků. Proto se v následujících odstavcích zabývám také, ikdyž okrajově, předvýběrem vhodných frameworků pro vývoj rozšířené, augmentované reality (AR).

4.1.4.2 Dostupné AR frameworky

Při vývoji aplikací s rozšířenou realitou, kdy digitální zařízení snímá prostor pomocí kamery, bývá největším problémem přesné určení pozice zařízení a směru, kterým kamera snímá okolní scénu. Jedná se o problém nalezení vstupní transformace z prostoru pohledu kamery do prostoru scény (viz obrázek 4.5a). V případě zamýšlené aplikace by se teoreticky mělo jednat o v čase konstantní transformaci, které se v průběhu běhu aplikace nemění. Prakticky to ovšem nelze předpokládat. Zároveň zcela jistě může dojít k drobné změně při jednotlivých spuštění aplikace např. vlivem nepřesného nasazení klipu na kameru tabletu, či lehce odlišnému usazení tabletu do staojanu. Lze také předpokládat, že se transformace bude mírně lišit v závyslosti na fyzických parametrech jednotlivých tabletů.

Z popsaných důvodů je proto dobré uvažovat v rámci finální aplikace o automatickém nalezení vstupního zobrazení při každém spuštění. Funkcionalita nalezení tohoto vstupního zobrazení je běžná napříč jednotlivými AR frameworky a často dobře funguje na základě tzv. „AR markerů“.

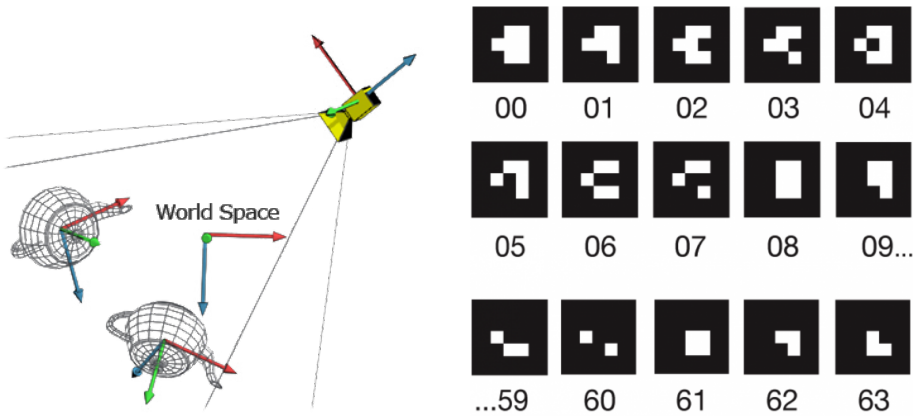
AR markery jsou jsou značky (viz obrázek 4.5b), díky kterým může framework na základě měření jejich pozice, velikosti a zkosení v obraze určit přesné natočení a pozici kamery vůči objektu s markery.

Jakmile je k dispozici informace o poloze kamery, lze do obrazu snadno vykreslit virtuální objekty. Pro vykreslování objektů lze pak vhodně využít hotových 3D enginů, kterých je dnes dostupná celá řada. Hlavním problémem, který zůstává vyřešit je rozpoznávání a klasifikace objektů na scéně.

Před budoucím výběrem konkrétního frameworku stanovuji základní požadavky na vlastnosti takového frameworku (v pořadí dle priorit):

1. Funkcionalita identifikovat AR markery

2. Multiplatformní (Android, iOS, Windows, Web, MAC, Linux)



(a) (b)

3. Zdarma dostupný, open source
4. Dobrá dokumentace
5. Pokročilé funkce

V tabulce 4.1 je seznam několika předvybraných, dostupných AR frameworků. Jsou zde ohodnoceny dle toho, jakým způsobem splňují požadavky. Každý požadavek je hodnocen na stupnici 0-5 (čím více, tím lépe).

Tabulka 4.1: Porovnání frameworků pro rozšířenou realitu

Název	Vlast. 1	Vlast. 2	Vlast. 3	Vlast. 4	Vlast. 5	Suma
AR Toolkit	**5**	**5**	**5**	**5**	**3**	**23**
Vuforia	**5**	**5**	**3**	**4**	**4**	**21**
Kudan	**5**	**5**	**3**	**5**	**5**	**23**
AR-media	**5**	**5**	**1**	**4**	**4**	**19**

4.2 Pochopení dat

Správné pochopení dat je pro úspěch DM procesu klíčové. V datech mohou obecně nastávat různé jevy. Data mohou být neúplná, nekonzistentní, zašuměná či zkreslená jakoukoliv neznámou transformací. Proto je před samotnou prací s daty důležité se nejprve zabývat pochopením těchto jevů, která se v datech vyskytují či mohou vyskytovat. Hledat jejich příčiny za účelem tyto jevy v budoucnu ošetřit.

V následující kapitole se proto zabývám pochopením obrazových dat, která přicházejí z kamery tabletového zařízení na vstup hledaného algoritmu. V kapitole nejprve definuji, co jsou to vstupní data, dále se zabývám analýzou

jevů vyskytujících se v experimentálně pořízených datech a na závěr stanovuji hlavní parametry kvality dat.

4.2.1 Definice vstupních dat

Vstupní data posyktující kamera referenčního tabletového zařízení jsou obvyklá obrazová data v RGB barevném modelu o rozlišení 1600x1200px. Formálně se tedy jedná o matici pixelů:

$$I[x, y] : N^2 \Rightarrow R^3$$

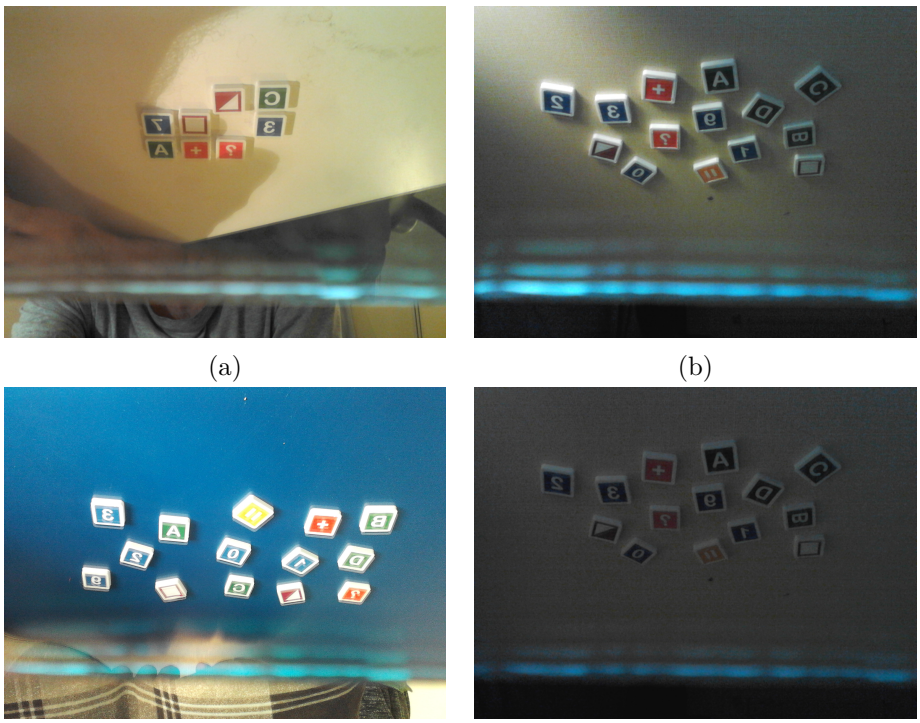
$$0 < x < 1600, 0 < y < 1200$$

4.2.2 Pořízení počátečních dat

Pro popis a analýzu hlavních jevů v datech je potřeba nejprve data pořídit. Počáteční data určená pro prvotní průzkum jevů byla sbírána pomocí vestavěné aplikace Kamera referenčního zařízení. Tablet byl umístěn ve stojanu a měl nasazen na frontální kameře zrcátkový klip, přičemž na scéně byly umístěny žetony s různými symboly. Data jsem pořizoval záměrně v různých prostředích. Výsledkem je set 5 obrázků z prostředí s rozdílnými světelnými podmínkami.

4.2.2.1 Ukázková počáteční data

Pro příklad a následný analytický popis dat zde uvádím několik z obrázků počátečních dat (viz obrázky fig:Scene-init).



(a) (b)

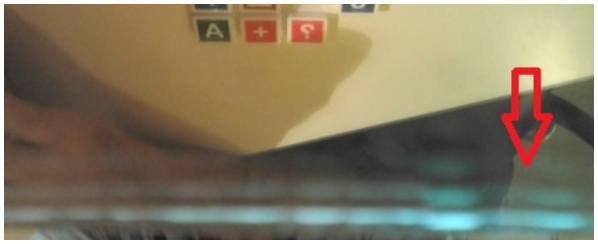
(c) (d)

Obrázek 4.6: Ukázková počáteční data

4.2.3 Analýza experimentálně pořízených dat

V této podkapitole se věnuji analýze nasbíraných počátečních dat. Popisuji zde jevy, které se nachází v datech na příkladech obrázků . Jevy v datech popisuji v následujícím seznamu:

1. **Odraz modrého světla na hraně zrcátka**
   V pořízených datech si lze všimnout jasného světle modrého efektu, který je způsoben odrazem přicházejícího světla na přední hraně zrcátka. Efekt je silnější, čím je ostřejší osvětlení scény. Naopak není tak znatelný na méně osvětlených scénách. Lze si také všimnout, že efekt není v obrazu přímo ohraničen nýbrž jeho intenzita postupně klesá se vzdáleností od okraje zrcátka. Tento nežádoucí jev lze odstranit několika způsoby: (1) oříznutí nasnímaného obrazu, (2) fyzické zaslepení přední hrany , (3) použitím zrcátka s jednou odrazovou vrstvou.

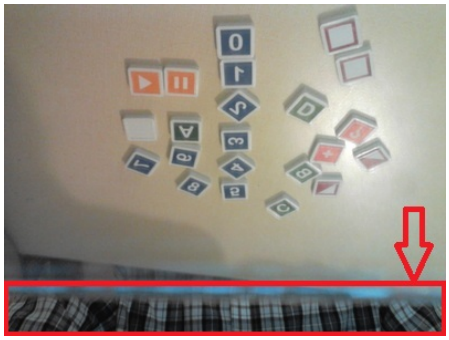


Obrázek 4.7: Odraz modrého světla na hraně zrcátka

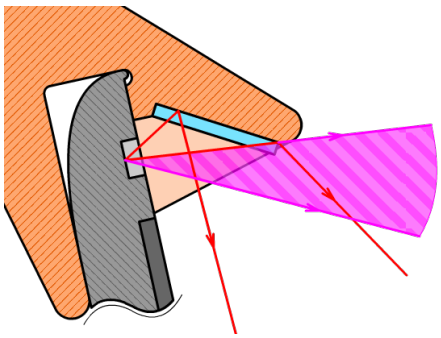
2. **Nevyužitý úhel pohledu kamery**
   V obrazových datech lze jasně pozorovat nevyužitý úhel pohledu kamery, který je popsaný v sekci 3.1 *Návrh potřebných HW komponent*. Jde o část vstupního světla, které vchází do kamery bez toho aniž by došlo k jeho odrazu v zrcátku. Nejedná se tedy o obraz scény nýbrž zpravidla o obraz uživatele aplikace.

3. **Rozostření vlivem dvojodrazu ve skle zrcátka**
   Toto rozostření ve skle zrcátka je další jev v datech, který byl předpokládán v rámci sekce **??** *Návrh potřebných HW komponent*. Jev je patrný především na barevných částech žetonů a sílí se vzdáleností žetonu od kamery a ostrostí přímého osvětlení. Matematicky se jedná o vážený součet obrazů žetonu na rozhraní *sklo-vzduch* a *sklo-kov*, které jsou od sebe vlivem tloušťky skla posunuty. Odpovídajícím transformačním modelem může být konvoluce s dvoubodovým konvolučním jádrem typu viz obrázek 4.9b, kde vzdálenost obou bodů je úměrná tloušťce zrcátka.

4. **Perspektivní vidění kamery a ztráta rozlišení**
   Tak jak jsem již popsal v sekci 2.6 *Analýza problému snímání scény*, se

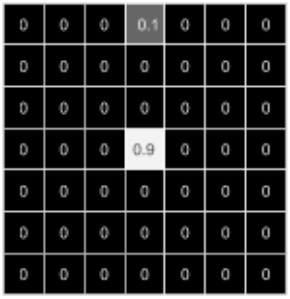


(a) Nevyužitý pohled kamery v obraze

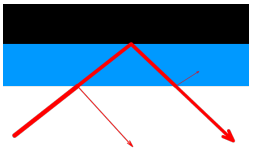


(b) Nevyužitý pohled kamery (fialově) schématicky v modelu zrcátkového klipu

(a) Rozostření žetonu ve vertikálním směru



(b) Modelové konvoluční jádro



(c) Model odrazu světla od skleněného zrcadla

vzdáleností od kamery dochází vlivem perspektivy ke zmenšování obrazů objektů na scéně. To má za následek snižování rozlišení vzdálenějších žetonů.

5. **Odraz modrého světla na hraně zrcátka** - Obrazy scén, které jsou hůře osvětleny jsou typicky poznamenané viditelně zvýšenou mírou šumu. Šum bývá nejvíce patrný na podkladu scény.

Obrázek 4.10: Výrazný šum ve snímcích špatně osvětlených scén scén

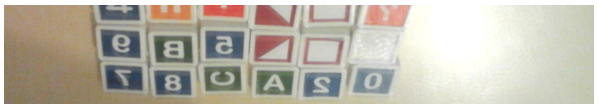
6. **Obraz scény je zrcadlově převrácen**

7. **Obraz scény je zrcadlově převrácen** - Na scénách se světlým barevným podkladem občas dochází ke splynutí barvy bílého okraje žetonu

s okolím. V obraze žetonu tak zůstává jedna výrazná hrana daná přechodem výplň žetonu - okraj žetonu dle které lze určit pozici a natočení žetonu.

Obrázek 4.11: Splynutí okraje žetonu s pozadím scény

8. **Rozdílné osvětlení napříč různými scénami**
   V pořízených obrazech si lze jednoduše všimnout různých intenzit osvětlení na různých místech v rámci jedné scény. Identifikovat lze také několik druhů stínů. Můžeme je rozdělit na stíny objektů na scéně (stíny vznikající v okolí žetonů při horším či ostrém nasvícení) a stíny objektů mimo scénu, které mohou být ostré či s postupným přechodem.



Obrázek 4.12: Nehomogenního osvětlení v obraze

9. **Rozdílné osvětlení napříč různými scénami**
   Napříč různými scénami si lze také všimnout rozdílnosti v osvětlení. Na jedné straně extrému se jedná o tmavé, špatně osvětlené scény. Na druhé straně extrému pak o scény nasvícené ostrým světlem. U tmavých scén se markantně projevuje šum v obraze. U scén nasvícených ostrým světlem je pak umocněn problém rozostření obrazu ve skle zrcátka.

   Z analýzy vyplývají jevy v datech, se kterými je třeba počítat v následném experimentování. Při úvahách o odstraňování těchto jevů je třeba míti na paměti závažnost daného jevu, výpočetní náročnost jeho odstranění a dopad na výslednou kvalitu. Pakliže bude některý z popsaných jevů omezovat úspěšnost algoritmu, je třeba přistoupit k jeho odstranění.

4.2.4 Kvalita dat

Kvalita dat je důležitým faktorem predikující úspěšnost celého algoritmu. Výhodou této práce je fakt, že data pro účely práce je třeba generovat samostatně, díky tomu se nestane, že by data byla nekonzistentní či nekompletní, jak tomu v případě jiných DM procesů může nastat. Největší možné problémy kvality dat týkající se této problematiky jsem identifikoval jako:

1. **Zašumněnost**
   Zašuměnost je častým problémem při zpracování obrazu. Je dána především světelnými podmínkami na konkrétní scéně a kvalitou hardwaru kamery. Je tedy potřeba se vyrovnat s rozdílností šumu na jednotlivých scénách

2. **Nedostatečné pokrytí datasetu, chybějící data**
   Špatné pokrytí datasetu by mohlo způsobit problémy při tvorbě klasifikačního modelu. Jak již bylo zmíněno, se vzdáleností od kamery se snižuje rozlišení snímaných objektů. Zároveň je možné, že žetony budou na scéně různě pootočeny. Je tedy potřeba aby se obrazy žetonů v trénovacím a testovacím datasetu vyskytovaly rovnoměrně ve všech oblastech snímané scény a s různými natočeními.

4.3 Příprava dat

Před modelováním problému, je třeba nejprve připravit data, na základě kterých bude prováděno experimentování a trénování konrétních modelů. V případě této práce se přípravou dat rozumí pořizování kamerových snímků scén s hledanými objekty (žetony) a jejich následné předzpracování. Aby výsledný algoritmus byl schopný co nejlépe rozpoznat a klasifikovat žetony na obecné scéně, je třeba nasnímat ideálně co nejvíce snímků scén s různými parametry. Cílem této podkapitoly je připravit dostatečně obsáhlý datový set pro trénování rozpoznávacího a klasifikačního modelu. Dále se nejprve zaměřuji na přípravu trénovacích a testovacích dat pro rozpoznávání a až následně pro klasifikaci.

4.3.1 Výběr - generování dat

Specifikem této práce z pohledu DM procesu je fakt, že oproti jiným DM procesům na počátku nejsou k dispozici žádná data, tudíž je potřeba zdrojová data (obrazy scén) generovat, nasnímat kamerou tabletového zařízení umístěného v sestavě stojan-zařízení-zrcátkový klip. Při snímání zdrojových dat je třeba pamatovat na definovanou, předpokládanou kvalitu výsledných dat.

Výsledný dataset by se měl skládat z různých scén s různými vlastnostmi osvětlení. Měl by obsahovat rovnoměrně zastoupené žetony dle jednotlivých tříd (číselných hodnot). Žetony jedné třídy by zároveň měli být rovnoměrně rozmístěny napříč snímané scény. Ideální dataset je takový, který obsahuje všechny možné záběry kamery (z různých úhlů, vzdáleností, za různého osvětlení,...) konkrétní třídy žetonu. Takový dataset by byl ovšem nekonečně veliký, nereálný a proto je nutno se omezit.

Před generováním dat proto nejprve definuji parametry scén dle kterých bude sestavován respektive generován výsledný dataset. Při pořizování snímku

budou tyto parametry postupně měněny, což by mělo být zárukou rovnoměrného a snad dostatečného pokrytí výsledného datasetu. V případě selhání klasifikátoru na některých datech je potřeba se vrátit k tomuto kroku a vygenerovat další doplňující data. Pro přípravu dat uvažuji následující parametry scén:

1. **Scény s různými světelnými podmínkami**
   Osvětlení scény mívá na výsledky algoritmů zpracování obrazu často zásadní vliv. Možností jak nasvítit scénu je nespočet. Scény mohou být osvětleny různou intenzitou, typem (bodové,..) a barvou světla. Těchto variant může být velmi mnoho. V rámci práce se omezím pouze na několik případů a scény budu snímat v 6 různých prostředích. V těchto 6 prostředích pak budou scény měněny dle následujících parametrů...

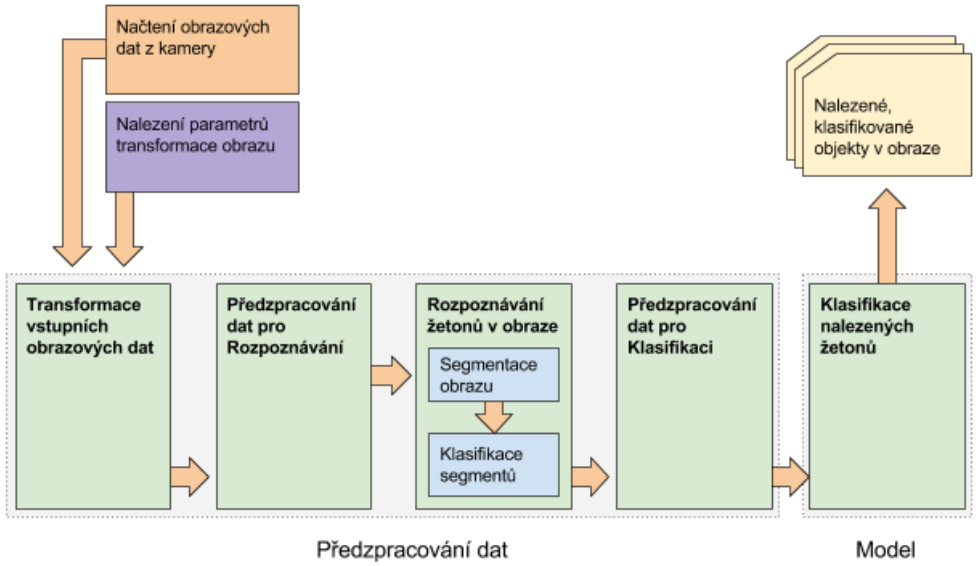
2. **Pozice žetonů**
   Pozice žetonů na scéně má vliv na jejich následné rozlišení v transformovaném obraze. Místa vzdálenější od kamery ztrácejí vlivem perspektivního vidění kamery rozlišení. Proto jsou žetony snímány z 10 pozic rovnoměrně rozmístěných na scéně. Žetony pak při pořizování snímků dané scény budou rotovat po daných pozicích. Tímto způsobem jsou získány obrazy jednotlivých tříd žetonů s různým rozlišením (dle pozice na scéně).

3. **Natočení žetonů**
   Krom parametru pozice je také třeba uvažovat natočení žetonu na konkrétní pozici na scéně. Při pořizování snímků budou na každé pozici žetony rotovány kolem své osy postupně v pěti úrovních od -45 do 45 stupňů (například -45, 22, 0, 28, 40).

4.3.2 Konstrukce dat

Konstrukce dat se skládá za prvé z generování/snímání obrazových dat jednotlivých scén dle popsaných parametrů v předešlé seki a za druhé z úpravy těchto dat do podoby vhodné pro experimentování a vývoj jednotlivých funkčních bloků hledaného algoritmického workflow. Konstrukcí dat se zabývám pouze pro bloky, které to vyžadují. Například v případě bloku předzpracování dat pro Rozpoznávání jsou vstupní data dána výstupem bloku Transformace vstupních dat. V tomto případě tedy konstrukci dat znovu neřeším, jelikož jsou výstupem předešlého bloku.

Obrazová data scén (zdrojová data celého DM procesu) jsou snímaná vestavěnou frontální kamerou referenčního zařízení umístěného v sestavě stojan-tablet-zrcátkový klip. Tato data je následně potřeba vhodným způsobem upravit pro práci na vývoji jednotlivých funkčních bloků hledaného algoritmu. Dále proto řeším konstrukci dat (tam kde je potřeba) pro jednotlivé funkční bloky (zelené viz obrázek 4.13) zvláště.



Obrázek 4.13: Funkční bloky hledaného algoritmického workflow

4.3.2.1 Konstrukce dat pro funkční blok Transformace vstupních dat

Vstupem funkčního bloku Transformace vstupních dat respektive celého hledaného algoritmu jsou surová obrazová data definovaná v sekci 4.2.1 *Definice vstupních dat* a parametry transformace ze souřadnicového systému kamery (nasnímaného obrazu) do souřadnicového systému scény popsané v sekci 4.1.2 *Návrh algoritmické workflow*. Data respektive snímky scén postupně naaranžovaných dle definovaných parametrů v sekci 4.3.1 *Výběr - generování dat* jsou snímána frontální kamerou tabletového zařízení umístěného v sestavě *stojan-tablet-zrcátkový klip*.

Takovýmto způsobem bylo pořízeno celkem 238 snímků na 7 různých scénách lišících se prostředím, typem a mírou osvětlení. Na většině z těchto snímků se nachází vždy 2x10 žetonů definovaných tříd 0-9. Pořízené snímky jsou umístěny v rámci přiloženého datového uložiště v adresářích pojmenovaných dle jednotlivých scén následovně: */Data/BasicSceneDatasets/SceneA,B,..,D-1,2,../1-source/*. Obrázek 4.14 ukazuje jeden ze 238 snímků scén.

4.3.2.2 Konstrukce dat pro tvorbu Klasifikačního modelu

Pro tvorbu klasifikačního modelu je potřeba nejprve připravit dostatečně veliký dataset trénovacích a testovacích dat. Vstupem klasifikačního modelu jsou výstupní data celé posloupnosti funkčních bloků *Předzpracování dat* (viz obrázek 4.13), nebo také výstup funkčního bloku Předzpracování pro Klasifikaci. Vývoj a fungování tohoto a předchozích funkčních bloků popisuji později v pří-

Obrázek 4.14: Ukázka jednoho snímku v připraveném datovém setu pro blok Transformace vstupních dat

slušných sekcích v rámci této kapitoly. Vstupem kalsifikačního modelu je matice o velikosti 14x14 pixelů s hodnotou pixelů v rozmezí 0-255. Výstupem je pak hodnota v rozmezí 0-9 reprezentující příslušnost obrazu žetonu na vstupu k třídě odpovídající hodnotě symbolu, kterým je žeton potištěn.

Pro generování trénovacího a testovacího datasetu pro model klasifikace bylo zapotřebí nejprve implementovat celou část algoritmické workflow zaměřenou na předzpracování dat. Za použití celého algoritmu předzpracování byl vygenerován potřebný dataset z pořízených snímků scén, které se nacházejí v adresáři */Data/BasicSceneDatasets/** v rámci přiloženého DVD disku. Pro generování byl použit skript */Implementation/DataPreparation/prepareDataFor-*

TokenClassification.m, který ukládá výstupní data v rámci datasetu každé scény do podadresáře *5-tokenspreprocessed/*. Takto vygenerovaná, předzpracovaná data bylo následně potřeba ručně roztřídit dle hodnoty žetonu. Rozříděná data jsou umístěna v rámci adresáře: */Data/TokenClassificationDataset/2-handlabeled* rozdělená do podadresářu dle tříd. Každá z tříd obsahuje téměř 400 vzorků dat. Některé třídy jsou zastoupeny méně. To je způsobeno tím, že funkční blok rozpoznávání žetonů nemá 100% úspěšnost.

Jelikož množství dat získané ze základního setu snímků scén není příliš

veliké (350-400 vzorků pro každou třídu), vygeneroval jsem programově nový rozsáhlejší dataset pro trénování a testování klasifikačního modelu. Tento dataset je vygenerován na základě ručně tříděných datech. Nejprve byl originální dataset rozdělen na trénovací a testovací množinu v přibližném poměru 2:1. Na každý datový vzorek takto rozdělených datasetů byla následně aplikována transformace rotace ve 4 náhodou ovlivněných úrovních. Tímto vzniknul trénovací resp. testovací dataset o velikosti přibližně 1200 resp. 400 vzorků pro každou třídu.

Vygenerovaná data jsou uložena v rámci adresáře:
/Data/TokenClassificationDataset v podadresářích *3-generated-test/* a *3-generated-train/*. Pro generování byl použit skript:
/Implementation/DataPrepration/prepareDataForTrainTestCNNet.m.

4.3.2.3 Konstrukce dat pro testování celého algoritmu

Pro testování celého algoritmu je vhodné nasnímat speciální data, která nebudou využita při vývoji jednotlivých funkčních bloků. Pro tyto účely jsem nasnímal 9 testovacích scén, které jsou uloženy v rámci přiloženého disku v adresáři: */Data/TestSceneDataset*. V datasetu se nachází scény s různým osvětlením. Nachází se v něm dále jedna velmi tmavá scéna a dva snímky scény, která nemá rovný povrch nýbrž vypouklý (jedná se o povrch gauče). Tuto scénu jsem přidal záměrně pro ozkoušení robustnosti algoritmu vůči odchylce od definice scény.

4.3.3 Funkční blok Transformace vstupních dat

Vstupem následného funkčního bloku *Předzpracování pro Rozpoznávání* jsou obrazová data transformovaná do ortogonálního „pohledu shora“ a ořezaná na přibližnou velikost scény. Jedná se o výstup z předchozího funkčního bloku. Díky této transformaci je ze vstupního obrazu odstraněno zmenšení objektů nacházejících se dále od kamery oproti objektům nacházejících se blízko kamery, které má na svědomí perspektivní vidění kamery.

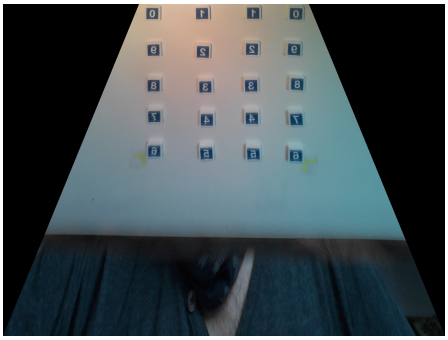
Nalezení parametrů této transformace předpokládám jako vstup celého algoritmu, jelikož tuto funkcionalitu typicky nabízí dnes dostupné frameworky pro rozšířenou realitu, které jsem zmiňoval v sekci 4.1.4 *Zhodnocení aktuální situace*. Pro účely této práce jsem použil parametry zobrazení, které byly nalezeny experimentálně za pomoci vestavěné funkce *maketform* a *imtransform* prostředí Matlab.

Obrázky 4.15a, 4.15b 4.15c ukazuje porovnání vstupních a výstupních obrazových dat projektivní transformace. Výsledný, transformovaný obrázek je ořezán tak aby zabíral přibližně prostor o něco málo větší než je definovaná scéna. Výstupem je čtvercový obraz, jelikož scéna je definována také ve tvaru čtverce. Výsledné rozlišení výstupního obrazu se může lehce měnit na základě

parametrů transformace, proto neuvádím jeho konkrétní hodnotu. V následujících krocích algoritmu velikost zpracovávaného obrazu sjednocuji.

Transformované snímky scén jsou umístěny v rámci přiloženého datového uložiště v adresářích pojmenovaných dle jednotlivých scén následovně: /Data/BasicSceneDatasets/SceneA,B,..,D-1,2,../2-transformed/.

(a) Zdrojová obrazová data z kamery



(b) Zdrojová obrazová data po aplikaci projektivní transformace

(c) Transformovaná data ořezaná na přibližnou velikost scény

4.3.4 Funkční blok Předzpracování pro Rozpoznávání

Účelem tohoto bloku je příprava dat pro funkční blok *Rozpoznávání*. V této sekci se věnuji analýze problému předzpracování dat, návrhu řešení a implementaci. Vstupem tohoto funkčního bloku jsou transformovaná data, jejichž přípravě se věnovala předchozí sekce.

Cílem bloku *Předzpracování dat* je data za prvé připravit do podoby vhodné pro další zpracování v blocích *Rozpoznávání a Klasifikace* a za druhé v obraze potlačit či odstranit jevy, které jsou způsobeny vlivy prostředí, ve kterém se nachází daná scéna. V rámci sekce 4.2.3 *Analýza experimentálně pořízených dat* jsem již tyto jevy popsal. Pro připomenutí se jedná o následující jevy:

1. Odraz modrého světla na hraně zrcátka
2. Nevyužitý úhel pohledu kamery
3. **Rozostření vlivem dvojodrazu ve skle zrcátka**
4. Perspektivní vidění kamery a ztráta rozlišení
5. **Viditelný šum při horším osvětlení**
6. Obraz scény je zrcadlově převrácen
7. Splynutí okraje žetonu s pozadím scény
8. **Nehomogenní osvětlení napříč scénou, přítomnost jemných i ostrých stínů**
9. **Rozdílné osvětlení napříč různými scénami**

Jevy 2 a 4 byly vyřešeny blokem Vstupní transformace obrazu, který odstraňuje perspektivní zkreslení žetonů a zanechává pouze výřez obrazu scény. Touto transformací je také potlačen jev 1, který se vyskytuje především v oříznutých částech obrazu. Jinde bývá jeho vliv minimální. Jevy 6 a 7 následné zpracování obrazu nikterak neovlivňují. Zbývají jevy 3, 5, 8, 9, které mohou a typicky budou mít vliv na následné zpracování. Jevy 8 a 9 pak můžeme shrnout do problematiky různorodosti osvětlení. Krom vypořádání se s popsanými jevy je cílem předzpracování příprava obrazových dat pro blok Rozpoznávání a Klasifikace.

Zpracování obrazu je často výpočetně náročný úkon. Ideální je proto zpracovávat obraz, jehož datová velikost je co nejmenší ale zároveň dostatečná pro splnění účelu zpracování. V případě této práce se jedná o provedení nalezení žetonů a klasifikace s dostatečnou úspěšností. V rámci předzpracování dat je tedy třeba uvažovat a provést následující úkony:

- Snížení datové velikosti obrazu

- Nevyužitý úhel pohledu kamery
- Potlačení viditelného šumu při horším osvětlení
- Rozostření vlivem dvojodrazu ve skle zrcátka
- Nehomogenní osvětlení napříč konkrétní scénou i různými scénami, přítomnost jemných i ostrých stínů

4.3.4.1 Snížení datové velikosti obrazu

Snížení datové velikosti zpracovávaného obrazu lze docílit (1) redukce rozlišení obrazu podvzorkováním a (2) redukcí reprezentace barvy jednotlivých pixelů. V obou případech se jedná o ztrátovou redukci dat. V obou případech je potřeba určit správné parametry redukce, tak aby ve výsledném obraze zůstalo dostatečné množství informace.

V případě rozlišení jsem se rozhodnul vstupní obraz podvzorkovat na velikost 256x256 pixelů. Pro stanovení této hodnoty rozlišení jsem přepočítal velikost žetonů a scény na velikost v pixelech. Scéna má reálně rozměry 21x21cm, žeton pak 2x2cm, přičemž symbol na něm potištěný zabírá přibližně (v závislosti na konkrétním symbolu) prostor o rozměrech 1x1cm. Pokud bude scéna převzorkována na rozlišení 256x256 pixelů pak bude jednomu centimetru odpovídat přibližně 12 pixelů. Problém rozpoznávání bude tedy o hledání objektů velikosti 24x24pixelů v obraze o celkovém rozlišení 256x256 pixelů. Model klasifikace žetonů pak bude zpracovávat data o nejmenší velikosti 12x12 pixelů.

V případě redukce reprezentace barev jsem se rozhodnul vstupní barevný obraz převést do šedotónu. RGB barevnost jednotlivých pixelů totiž nenese jednoznačnou informaci o příslušnosti daného pixelu k žetonu či příslušnosti žetonu k definovaným třídám.

Jistota toho, že zvolené parametry redukce datové velikosti obrazu, jsou správné v tomto bodě práce není. Při prvotním nastavování parametrů jsem vycházel ze zkušeností jiných s klasifikací textu v obraze. Následné experimenty ovšem ukázaly, že zvolené parametry jsou pro účely algoritmu minimálně dostačující.

4.3.4.2 Odstranění šumu v obraze

V celé práci uvažuji o šumu v obraze jako o aditivním gaussovském modelu šumu. Tento model popisuje šum tak, že každý pixel obrazu I je výsledkem součtu hodnoty pixelu skutečného obrazu Ir s náhodnou hodnotou N danou gaussovskou náhodnou distribucí, tedy:

$$I = I_r + N$$

Pro odstranění šumu v obraze existuje mnoho metod. Základně tyto metody lze rozdělit na lineární a nelineární. Lineární metodou je například i

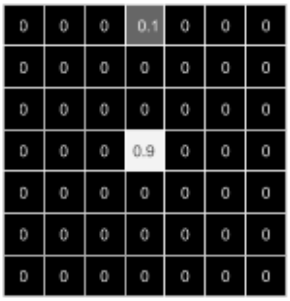
operace konvoluce, která je použita i pro podvzorkování obrazu, kterému jsem se se věnovall v předchozí sekci.

Je známo, že šum v obraze je ve frekvenčním spektru obrazu vysokofrekvenční složka. Aplikací lineární konvoluce dochází k podvzorkování obrazu, což má za následek ztrátu detailu. Jelikož operace konvoluce potlačuje vysokofrekvenční složky obrazu, podvzorkováním vstupního obrazu již došlo k jistému potlačení šumu.

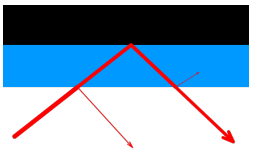
4.3.4.3 Rozostření vlivem dvojodrazu ve skle zrcátka

Samotný jev a důvod jeho vzniku jsem popsal v sekci 4.2.3 *Analýza experimentálně pořízených dat*. Transformace vstupního obrazu na rozhraní zrcátka odpovídá operaci konvoluce s vhodným jádrem, jehož odhadovanou podobu zobrazuje obrázek 4.16b.

(a) Rozostření žetonu ve vertikálním směru



(b) Modelové konvoluční jádro



(c) Model odrazu světla od skleněného zrcadla

Jelikož je tento jev způsoben operací konvoluce, lze teoreticky k ní inverzní operací získat originální obraz. Inverzní operace konvoluce se nazývá dekonvoluce a je možná, pokud je známo přesné konvoluční jádro. Pokud přesné konvoluční jádro známo není, existují i metody pro jeho odhad. Nicméně dekonvoluce samotná je poměrně náročná operace. Vyžaduje převod obrazu do frekvenční domény jejíž složitost je $O(|I|log|I|)$. Proto v případě malé úspěšnosti výsledného algoritmu je v budoucnu lépe řešit problém výměnou aktuálního zrcátka za zrcátko s jednou odrazivou vrstvou.

4.3.4.4 Nehomogenní osvětlení a stíny

Na různých scénách a na různých místech daných scén jsou jednotlivé body scény nasvíceny obecně různou intenzitou světla. Ideálně bychom ale chtěli mít pro další zpracování obraz, na kterém je každý bod scény nasvícen stejnou intenzitou. Obraz scény je tvořen skutečným obrazem I_r, který je nasvícen homogenně a osvětlením $L(s)$, které je různé pro každou scénou s. Výsledný obraz je pak dán následovně:

$$I_s = L(s) * I$$

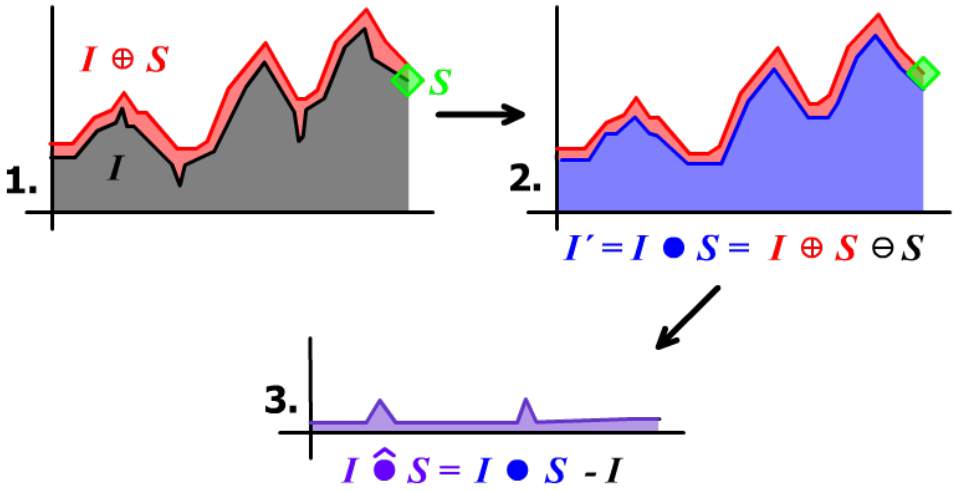
kde operace $*$ je daná použitým modelem osvětlení. V ideálním případě bychom potřebovali nalézt transformaci t, která bude mít následující vlastnost:

$$t(I = L(s_i) * I) \approx t(I = L(s_j) * I); s_i, s_j \in S$$

kde S je prostor všech možných scén. Taková transformace se nazývá světelně invariantní, jelikož na její výsledek nemá vliv osvětlení vstupního obrazu. K ideální transformaci t se můžeme pouze přiblížit. Vlivem šumu je nemožné ideální světelně invariantní transformaci nalézt. Reálně nám jde pak o to nalézt takovou transformaci t, která minimalizuje rozdíl různě nasvětlených, transformovaných obrazů totožné scény:

$$\sum_{x,y} |t(I = L(s_i) * I) - t(I = L(s_j) * I)| \to 0; s_i, s_j \in S$$

Jako vhodná transformace, která potlačuje rozdílnosti v osvětlení napříč scénou, se ukázala tzv. „Bottom-hat“ transformace. Tato transformace využívá efektu dilatace a eroze, při kterém dochází k zaoblení hran v obraze. Jednotlivé kroky transformace názorně popisuje obrázek 4.17.



Obrázek 4.17: Princip Bottom-hat transformace

Původní signál I je dilatován a posléze erodován pomocí strukturovacího elementu S. Od dilatovaného a následně erodovaného obrazu I je následně odečten původní obraz I.

Bottom-hat transformace má několik výhodných vlastností. Za prvé zvýrazňuje tmavé regiony obrazu, které jsou menší jak strukturovací element S.

Tato vlastnost lze využít pro zvýraznění barevné výplně žetonu, která je typicky jedno z nejtmavších míst v pořízených snímcích scén. Za druhé pak tato transformace vyrovnává jas těch regionů obrazu, které jsou větší jak strukturovací element. Což má za následek potlačení rozdílné intenzity osvětlení napříč scénou. Tento jev lze hezky pozorovat na obrázku 4.17, kde ačkoliv vstupní signál (černý) má na ose x silně rozdílnou intenzitu, výstupní (fialový) signál je až na vrcholy téměř konstantní. Tato vlastnost je dána díky odečtení originálního obrazu od obrazu, který byl postupně dilatován a erodován.

Při experimentech se ukázalo, že pro následnou klasifikaci žetonů je vhodné výstupnímu obrazu Bottom-hat transformace zvýšit kontrast. Bottom-hat transformace totiž na špatně osvětlených scénách zvýrazňuje tmavé regiony méně než na dobře osvětlených scénách, což mělo za následek nedobré výsledky klasifikátoru během experimentování. Pro následnou segmentaci se ale ukázalo jako lepší využít původní výstup transformace bez aplikace zvýraznění kontrastu. Zvýšení kontrastu totiž posiluje šum v obraze, který má na následnou segmentaci negativní dopady.

Pro zvýšení kontrastu lze využít metody ekvalizace histogramu. Při ekvalizaci histogramu je cílem přemapovat odstíny šedi v obraze tak aby pokrývaly rovnoměrně celou škálu šedotónu. Jako nejvhodnější se při experimentování ukázala metoda tzv. „contrast-limited adaptive histogram equalization“ [10], která provádí ekvalizaci histogramů v rámci definovaných n x n subregionů původního obrazu. Tato metoda, jak z názvu vyplývá, zároveň limituje maximální velikost kontrastu. To se ukázalo jako vyhovující vlastnost jelikož při ekvalizaci histogramu docházelo k výraznému zesílení šumu. Výsledek zvýšení kontrastu obrazu hůře nasvětlené scény lze vidět na obrázku 4.18.

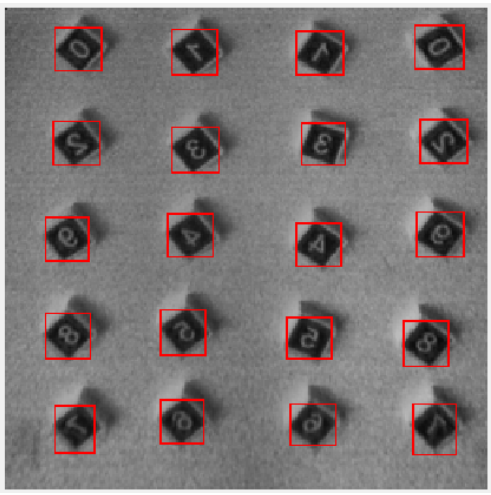
Obrázek 4.18: Výsledek aplikování metody adaptivního histogramu s limitovaným kontrastem

Předzpracované snímky scén jsou umístěny v rámci přiloženého datového uložiště v adresářích pojmenovaných dle jednotlivých scén následovně: */Data/BasicSceneDatasets/SceneA,B,..,D-1,2,../3-preprocessed/*.

4.3.4.5 Funkční blok Rozpoznávání

Účelem bloku Rozpoznávání je připravit data pro funkční blok *Klasifikace*. Cílem je nalézt žetony v obraze scény. Vstupem bloku je šedotónový obraz, který je výstupem bloku *Předzpracování*. Výstupem je seznam tzv. „bounding boxů“, rámců ohraničujících žetony v obraze. Výstupní datová struktura obsahuje pro každý takový rámec informaci o jeho pozici na scéně a velikosti

šířky a výšky v pixelech (znázornění bounding boxů nalezených žetonů viz obrázek 4.19).



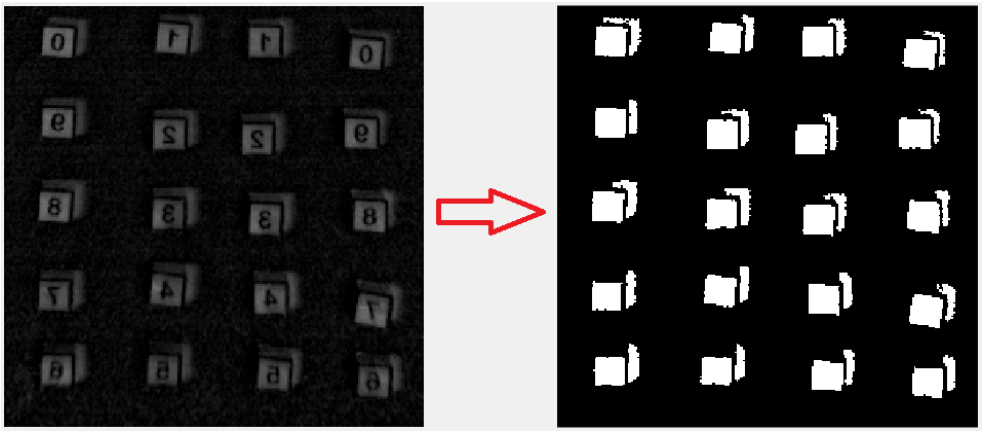
Obrázek 4.19: Grafické znázornění výstupu rozpoznávání („bounding boxů“)

Funkční blok Rozpoznávání lze rozdělit na dva podproblémy. Prvním podproblémem je segmentace, jejímž cílem je vhodně rozdělit obraz na segmenty, které pokud možnost odpovídají objektům přítomným na scéně. Výstupem segmenteru jsou segmenty obrazu, které jsou definovany shluky pixelů. Druhým podproblémem je pak klasifikace těchto segmentů. Je potřeba rozlišit mezi segmenty, které odpovídají žetonům v obraze a které nikoliv.

4.3.4.6 Segmentace obrazu

Problém segmentace se ukázal jako poměrně jednoduchý. Vstupní data jsou totiž dobře segmentovatelná jednoduchou metodou prahování. Díky Bottom-hat transformaci jsou v obraze zvýrazněna původně tmavá místa menší než strukturovací element, která zpravidla odpovídají barevné výplni žetonů. Původně světlá místa (mj. ta která odpovídají okrajům žetonů) byla transformována do černé barvy, díky čemuž jsou v obraze žetony od okolí pěkně odděleny (viz obrázek 4.20).

Při prahování dochází k mapování původní šedotónové škály do čistě černobílé reprezentace (viz obrázek 4.20). Zásadní je pro tuto metodu nalezení správného hodnoty prahu, dle které jsou pixely obrazu namapovány na hodnotu 0 respektive 1. Pro nalezení optimálního prahu jsem použil hotovou funkci graythresh v prostředí matlab fungující na principu metody *Otsu* [11] . Po aplikaci prahování byly dále odstraněny segmenty menší jak 30 pixelů a byly vyplněny „díry“ všech nalezených segmentů. Ukázka výsledku segmentace je vidět na obrázku 4.20.



Obrázek 4.20: Segmentace obrazu předzpracovaného Bottom-hat transformací

4.3.4.7 Klasifikace segmentů

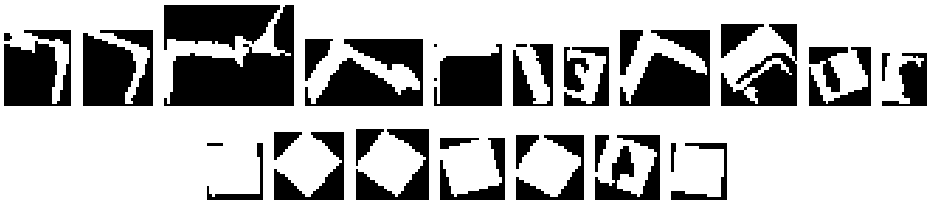
Jak je vidět na obrázku 4.20 nalezené segmenty neodpovídají ve všech případech žetonům. Segmentovací funkce má na výstupu také shluky pixelů, které odpovídají obecně tmavým místům na scéně, které jsou menší jak strukturovací element Bottom-hat transformace. Typicky se jedná o stíny, které vrhají žetony vlivem silného směrového osvětlení. Krom stínů se ale může jednat o jakýkoliv tmavý objekt na scéně. Je proto potřeba segmenty klasifikovat do tříd 0 resp. 1 dle toho zdali se může jednat o žetony či nikoliv.

Lidským okem jsou segmenty snadno rozpoznatelné na základě jejich tvarových charakteristik. Segmenty odpovídající žetonům mají podobnou velikost a přibližně čtvercový tvar. Jako vhodná cesta se proto jeví pro jednotlivé segmenty měřit tvarové charakteristiky a na základě těch natrénovat klasifikátor. Pro měření tvarových chrakateristik jsem vybral následující:

- Šířka „bounding boxu“ segmentu
- Výška „bounding boxu“ segmentu
- Poměř šířky a výšky
- Obvod segmentu
- Obsah segmentu
- Délka hlavní osy elipsy, která má stejný normalizovaný druhý centrální moment jako daný segment
- Délka vedlejší osy elipsy
- Poměr délky hlavní a vedlejší osy elipsy

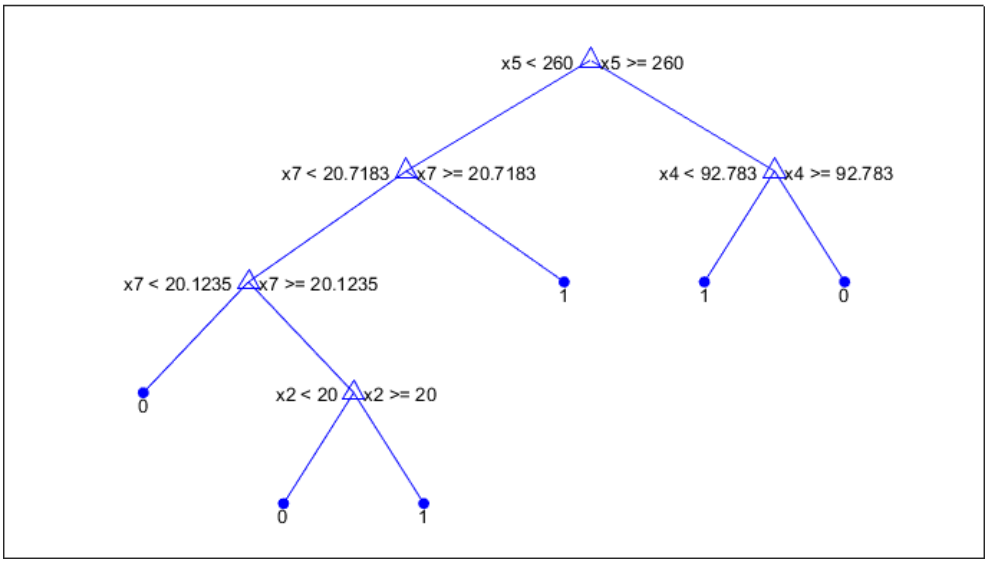
Jelikož se jedná o poměrně snadnou klasifikační úlohu, která by šla vyřešit poměrně efektivně s použitím několika *if-else* větví, rozhodl jsem se pro řešení využít model rozhodovacího stromu pro jeho jednoduchost a rychlost vybavovací fáze.

Pro účely trénování modelu jsem připravil dataset přibližně o 6 stech různých segmentů, rovnoměrně rozdělený na segmenty žetonů a ostatní (ukázka viz obrázek 4.21).



Obrázek 4.21: Ukázka datasetu pro trénování rozhodovacího stromu

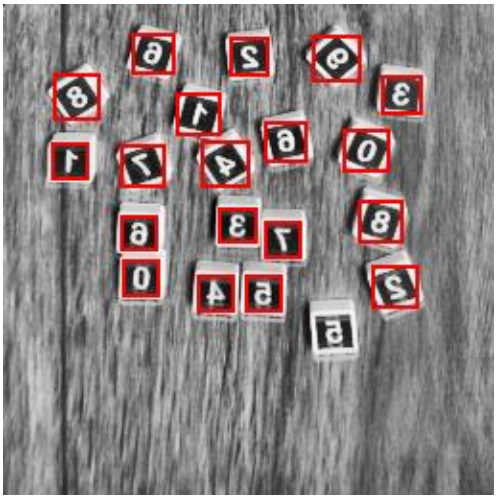
Experimentování a trénování rozhodovacích stromů jsem prováděl v rámci skriptu */Implementation/Scripts/trainCTree.m*, který je umístěn na přiloženém DVD. Nalezený model dosahuje 100% úspěšnosti na testovacích datech a znázorňuje ho obrázek 4.22 kde parametry Xi odpovídají popsaným měřeným tvarovým charakteristikám segmentů.



Obrázek 4.22: Model rozhodovacího stromu

4.3.4.8 Testování úspěšnosti Rozpoznávání

Pro testování úspěšnosti algoritmu byl nasnímán speciální testovací set snímků scén viz sekce Konstrukce dat pro testování celého algoritmu. Tento set byl následně podroben rozpoznávání žetonů. Zpracované obrazy scén s vyznačenými rozpoznanými žetony se nacházejí v adresáři */Results/TokenRecognitionTest*. V testovacím setu se nepodařilo rozpoznat pouze jeden žeton (viz obrázek 4.23). Celkem se v setu nachází 179 žetonů. Celková úspěšnost Rozpoznávání na testovacích datech je tedy 99,4%. Testování je implementováno v rámci skriptu: */Implementation/Scripts/testTokenRecognition.m*.

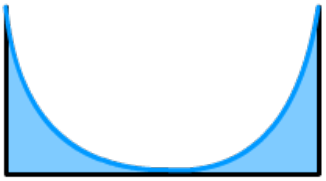


Obrázek 4.23: Nerozpoznaný žeton v testovacích datech

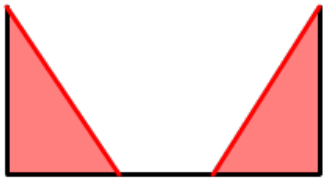
4.3.5 Funkční blok Předzpracování pro klasifikaci

Účelem tohoto funkčního bloku je připravit data pro Klasifikační model. Vstupem bloku je předzpracovaný šedotónový obrázek (výstup bloku Předzpracování pro Rozpoznávání) a informace („bounding box“) o pozici žetonu. Výstupem je pak matice pixelu o rozměrech 14x14 pixelů.

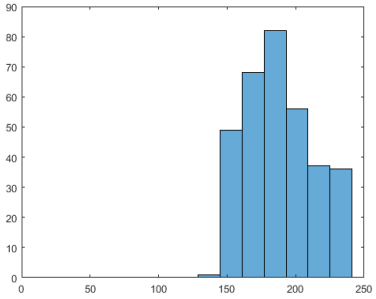
V ideálním případě bychom na vstupu klasifikátoru chtěli mít černo bílý obraz, ve kterém je tvar číselného symbolu ohraničen jasnými, plynulými hranami. Histogram takového ideálního obrazu by mohl vypadat například jako na obrázku 4.24a. Histogram vstupního obrazu je ovšem typicky nevyvážen viz příklad na obrázku 4.24c. Z toho důvodu v rámci tohoto bloku provádím ekvalizaci histogramu dle funkce viz obrázek 4.24b. Touto funkcí aproximuji ideální tvar histogramu. Výstup ekvalizace histogramu na lze vidět na obrázku 4.24d.



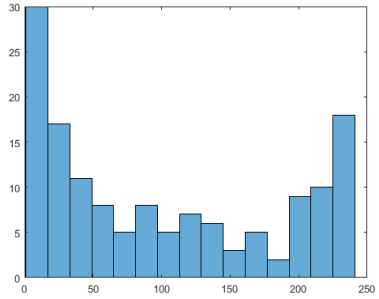
(a) Ideální histogram vstupního obrazu



(b) Aproximace ideálního histogramu



(c) Histogram typického obrazu na vstupu bloku Předzpracování pro klasifikaci



(d) Histogram vstupního obrazu po aplikaci ekvalizace dle definované funkce

4.4 Modelování

V rámci této kapitoly se věnuji hledání vhodného modelu pro problém klasifikace žetonů do deseti tříd 0,1,2,..,9 dle hodnoty jejich symbolu. V kapitole se nejprve věnuji výběru modelovací techniky. Následuje popis fungování vybrané techniky, tvorba a trénování modelu. Závěrem kapitoly se pak věnuji optimalizaci parametrů modelu za účelem zvýšení jeho úspěšnosti klasifikace a testování výsledného modelu.

4.4.1 Výběr modelovací techniky

Úloha klasifikace číslic spadá obecně do problematiky rozpoznávání vzorů. Rozpoznávání číslic, zejména pak těch ručně psaných, je problematika již poměrně probádaná. První úspěšné klasifikátory ručně psaných číslic byly vyvinuty okolo počátku devadesátých let minulého století. Rozpoznávání ručně psaných číslic byla tehdy jedna z prvních úspěšných aplikací neuronových sítí.

Tuto úspěšnou neuronovou síť pro rozpoznávání ručně psaných číslic v roce 1990 představili Y. LeCunn, B.Boser, J. S. Denker, D. Henderson, R. E Howard, W. Hubbard a L.D. Jackel [4]. Ukázalo se, že pro učení neuronových sítí se nejlépe hodí algoritmy založené na tzv. „gradient based learning“, které byly popsány již v padesátých letech minulého století. Na tomto principu funguje algoritmus zpětného šíření chyby „Gradient back propagation“, který se

pro trénování neuronových sítí dnes běžně používá. Jako nejlepší neuronové sítě pro úlohy zpracování obrazu se ukázaly tzv. konvoluční neuronové sítě[1].

Konvoluční sítě jsou dnes využívány na mnohem složitější úlohy než je klasifikace číslic a písmen psaného textu. Tyto neuronové sítě zaznamenávají silný úspěch obecně v oblasti rozpoznávání a klasifikace objektů v obraze [2]. Ve své práci jsem se proto rozhodnul dále využít jako klasifikační model právě konvoluční neuronovou síť, jelikož tyto sítě byly již mnohokrát s úspěchem využity na řešení problému klasifikace číslic.

4.4.2 Princip fungování konvoluční neuronové sítě

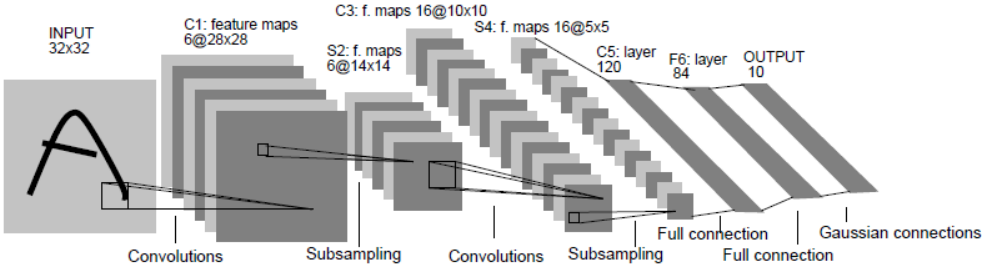
Obvyklý model rozpoznávání vzorů v obraze se skládá ze dvou částí. První částí je tzv. extraktor, který realizuje extrakci příznaků ze vstupního obrazu. Extrahuje tak z obrazu zajímavé informace a eliminuje ty irelevantní. Extraktor je v tomto modelu navržen ručně, člověkem. Druhou částí je pak natrénovatelný klasifikátor, který kategorizuje extrahované příznaky do tříd. Problémem tohoto modelu je fakt, že je třeba vždy předpřipravit ručně vhodné extraktory příznaků. Tímto způsobem ovšem můžeme některé potenciálně vhodné příznaky snadno opomenout nebo zvolit nevhodně jiné či jejich špatnou kombinaci. Zajímavější přístupem je nechat extraktor se sám naučit, jaké příznaky jsou zajímavé a jaké jsou irelevantní a přizpůsobovat se během učení trénovacím datům.

Tento přístup lze teoreticky realizovat pomocí klasické dopředné neuronové sítě, nicméně v případě aplikace na problém rozpoznávání v obrazu nastávají problémy. Prvním problémem je velikost vstupu. Obrazová data jsou typicky veliká. Obsahují minimálně řádově stovky pixelů. V případě plného propojení takového vstupu s první vrstvou neuronů se problém stává velmi výpočetně náročným. Další problém je to, že díky množství vstupních dat v celé síti naroste výrazně počet vah, které je potřeba naučit během procesu trénování. Velmi tak rostou nároky na velikost trénovacího datasetu. Zásadní nevýhodou takové sítě je pak to (pokud není skutečně obrovská a výpočetně nezvládnutelná), že nerealizuje invariantní transformace vůči posuvu či lokálnímu zkreslení. Výsledek je tedy silně závislí na tom, kde se rozpoznávaný vzor ve vstupních obrazových datech nachází. Proto by v takovém případě bylo typicky třeba vstupní obraz neuronové sítě prvně vhodně předzpracovat, vycentrovat ho a normalizovat jeho velikost.

Takto plně propojená neuronová síť ale mj. také ignoruje topologické vlastnosti vstupních dat. V rozpoznávaných obrazových datech se často vyskytují silné lokální struktury (hrany, rohy,..). To je koncekonců důvod, proč jsou tvořeny a kombinovány extraktory jednotlivých příznaků v prvně zmíněném, obvyklém modelu rozpoznávání vzorů, před tím než jsou data zpracována natrénovaným klasifikátorem. Konvoluční neuronové sítě vynucují extrakci lokálních příznaků omezením receptivní oblasti neuronů skryté vrstvy. Neurony

skrytých sítí jsou na rozdíl od plně propojených neurnových sítí ovlivňovány pouze malou okolní oblastí neuronů předchozí vrstvy nikoliv všemi neurony.

Architektura konvoluční sítě kombinuje několik prvků, kterými je zajištěna určitá invariance posunu, škálování, a zkreslení. Jsou jimi již zmíněné lokální receptivní oblasti, sdílení vah mezi neurony a podvzorkování. Známou konvoluční neuronovou sítí, která byla konstruována pro problém rozpoznávání psaných číslic je síť LeNet-5 viz obrázku 4.25.



Obrázek 4.25: Architektura sítě LeNet-5

Vstupní vrstva (INPUT) přijímá obrazová data číslic o daném rozlišení. Každý neuron skryté vrstvy vždy přijímá vstupy pouze od neuronů předchozí vrstvy v definovaném malém okolí. Díky lokální percepci mohou neurony extrahovat základní vizuální příznaky jako jsou hrany, rohy, apod. Tyto příznaky jsou poté kombinovány v následujících vrstvách za účelem detekování více komplexních příznaků. Lokální percepce je definována maticí vah. Tato matice odpovídá defacto konvolučnímu jádru, které je aplikováno na předchozí vrstvu. Z tohoto důvodu se těmto sítím říká konvoluční. Realizují totiž operaci konvoluce. Neurony jsou v jednotlivých vrstvách uspořádány do rovin ve který sdílejí vektor vah. Výstupy neuronů takových rovin se nazývají mapy příznaků. Celá konvoluční vrstva se skládá z několika takových map příznaků s odlišnými vektory vah. Díky tomu je možné v každém místě vstupního obrazu nebo předchozí vrstvy identifikovat vícero různých příznaků.

Příkladem je první skrytá vrstva *C1* sítě LeNet-5, která je organizovaná do 6 map příznaků. Každý z neuronů této vrstvy je propojen s 5x5 neurony v jeho perceptivním okolí předchozí vrstvy, přičemž perceptivní okolí sousedících neuronů v mapě příznaků jsou nastavena tak, že se překrývají. Každý neuron má 25+1 trénovatelných vah (+1 je pro bias).

Jakmile je v rámci konvoluční vrstvy detekován konkrétní příznak, informace o jeho přesné pozici se stává méně důležitou. Relevantní je totiž pouze jeho přibližná pozice vůči ostatním příznakům. Přesná informace o pozici příznaku je dokonce pro generalizaci sítě škodlivá, jelikož pozice příznaků vstupních obrazových dat patřící do stejné třídy se může měnit (například vlivem posuvu). Redukce přesnosti polohy příznaku je řešena podvzorkováním mapy příznaků. Tuto operaci lze provést několika způsoby. Často používanou operací

je tzv. max-pooling (výběr maximální hodnoty) nebo průměrování nxn bodů (obvykle se používá n=2) mapy příznaků. V rámci celé sítě tuto operaci řeší vrstvy následující ihned po konvolučních vrstvách (značeny *S2* a *S4* na obrázku 4.25). Každý neuron provede operaci průměrování (nebo max-pooling) na svém perceptivním okolí (v případě LeNet-5 jde o okolí 2x2) předchozí vrstvy, následně hodnotu multiplikuje svojí vahou, přičte bias a aplikuje sigmoidí transformaci, stejně jako v běžných neuoronových sítích. Perceptivní okolí jednotlivých neuronů ve vrstvách podvzorkování se nepřekrývají.

Postupnou aplikací konvolučních vrstev následovaných vrstvami podvzorkování se snižuje rozlišení jednotlivých map příznaků. Na úkor toho je ovšem v dalších vrstvách rozšiřován počet map příznaků. Díky této redukcí rozlišení map příznaků a následné kompenzaci navýšením počtu těchto map příznaků mohou konvoluční sítě dosahovat vysokého stupně invariance vůči geometrickým transformacím. Neurony v konvoluční vrstvě *C3* na obrázku 4.25 mohou být propojen s několika mapami příznaků předchozí vrstvy najednou. Na závěru celé sítě LeNet-5 se pak nacházejí dvě plně propojené vrstvy neuronů a vrstva výstupní.

Váhy celé síťe jsou trénovány pomocí algoritmu zpětného propagování chyby. Díky tomu na konvoluční sítě můžeme pohlížet tak, že si trénují vlastní extraktory příznaků. Extrahované příznaky ve vrstvě *S4* (viz obrázek 4.25) jsou následně klasifikovány plně propojenou vícevrstvou neuronovou sítí.

4.4.3 Tvorba modelu

Při návrhu modelu je potřeba uvažovat jednotlivé parametry architektury konvoluční sítě. Hlavními parametry jsou:

- Počet dvojic vrstev konvoluční vrstva-podvzorkovací vrstva
- Pro každou konvoluční vrstvu je třeba určit parametry:
  - velikost map příznaků
  - Velikost překryvu oblastí percepce sousedících neuronů
  - počet map příznaků
- Způsob a míra podvzorkování v podvzorkovacích vrstvách
- Počet plně propojených vrstev

Pro účely mojí práce a prvotní experimentování s konvolučními sítěmi jsem se rozhodnul vytvořit jednodušší model sítě, než jaký je popsán v předchozí sekci. LeNet-5. Vedlo mě k tomu několik důvodů. Prvním důvodem je fakt, že trénování hlubších konvolučních sítí je velmi výpočetně náročné a trvá poměrně dlouhou dobu. Podobným důvodem je pak také náročnost vybavovací fáze, která může být u hlubších sítí typicky také časově náročná. Pro zjednodušení sítě je ale hlavním důvodem to, že klasifikovaná obrazová data v rámci

této práce nejsou ručně psaná jako v případě LeNet-5, nýbrž každé číslo má na vstupu algoritmu stejný tvar. To mě vede k domněnce, že pro účely hledaného algoritmu bude stačit síť s menší sílou extrakce příznaků.

Z popsaných důvodů jsem se rozhodnul experimentovat hlavně s architekturou sítě, která má pouze jednu konvoluční resp. podvzorkovací vrstvu. Pro podvzorkování uvažuji operaci „max-pooling“ oblasti 2x2. Díky tomu vrstva podvzorkování bude mít vždy čtyřikrát menší rozlišení. Parametry počtu (P1) a velikosti (P2) konvolučního filtru konvoluční vrstvy podléhaly po prvotním experimentování optimalizaci. Přesah perceptivních oblastí sousedních neuronů konvoluční vrstvy jsem na základě experimentování nastavil tak aby byl vždy roven alespoň 1 pixelu nebo více. Na základě provedených experimentů lze říci, že čím menší přesah byl nastaven, tím horší byly výsledky sítě.

4.4.3.1 Tvorba modelu

Pro trénování modelu bylo nejprve potřeba připravit trénovací a testovací data. Postup přípravy těchto dat popisuji v předchozí kapitole v rámci sekce Konstrukce dat pro tvorbu Klasifikačního modelu. Experimenty s trénováním modelu byly prováděny v rámci skriptu: */Implementation/Scripts/trainCNNet.m*. V prvních experimentech měl navržený model konvoluční neuronové sítě úspěšnost v rozmezí 90-95%. Proto jsem dále přistoupil k hledání optimálních parametrů *P1* a *P2*.

4.4.3.2 Optimalizace parametrů modelu

Optimalizaci jsem podrobil parametr *P1* počet map příznaků v konvoluční vrstvě a parametr *P2* velikost filtru první konvoluční vrstvy. Pro každou kombinaci dále určených hodnot parametrů bylo natrénováno několik různých konvolučních sítí, které byly následně testovány na validačním datasetu. Validační dataset byl vytvořen odebráním náhodně výbraných vzorků dat z trénovací množiny (při zachování rovnoměrného zastoupení tříd). Takovýmto způsobem jsem proměřil všechny vygenerované konvoluční sítě, z čehož vyplynuly obecně suboptimální hodnoty těchto parametrů. Parametr *P1* nabýval postupně hodnot: *20, 15, 10, 5*. Parametr *P2* nabýval hodnot: *7, 5*.

4.4.3.3 Testování modelu

Pro testování modelu jsem použil konvoluční síť, která měla nejlepší výsledky během procesu optimalizace parametrů popsaného v předchozí sekci. Nalezený model konvoluční sítě je uložen v adresáři: */Results/Models/CNN/BestConvnet*. Danou síť jsem podrobil měření úspěšnosti na testovacích datech připravených v rámci sekce Konstrukce dat pro tvorbu klasifikačního modelu. Výsledky testování prezentuje následující matice záměn viz obrázek 4.26.

4.5 Evaluace procesu

V této kapitole se zabývám zhodnocení výsledků DM procesu. Aby bylo možné provést úplnou evaluaci, provedl jsem nejprve testování celého algoritmického workflow. Tento test je implementován v rámci skriptu: */Implementation/Scripts/testAlgorithmWorkflow/*. Výstupem tohoto testu je sada zpracovaných testovacích obrazových dat (definovaných v sekci Konstrukce dat pro testování celého algoritmu) s vyznačenými nalezenými a klasifikovanými žetony viz obrázek 4.27.

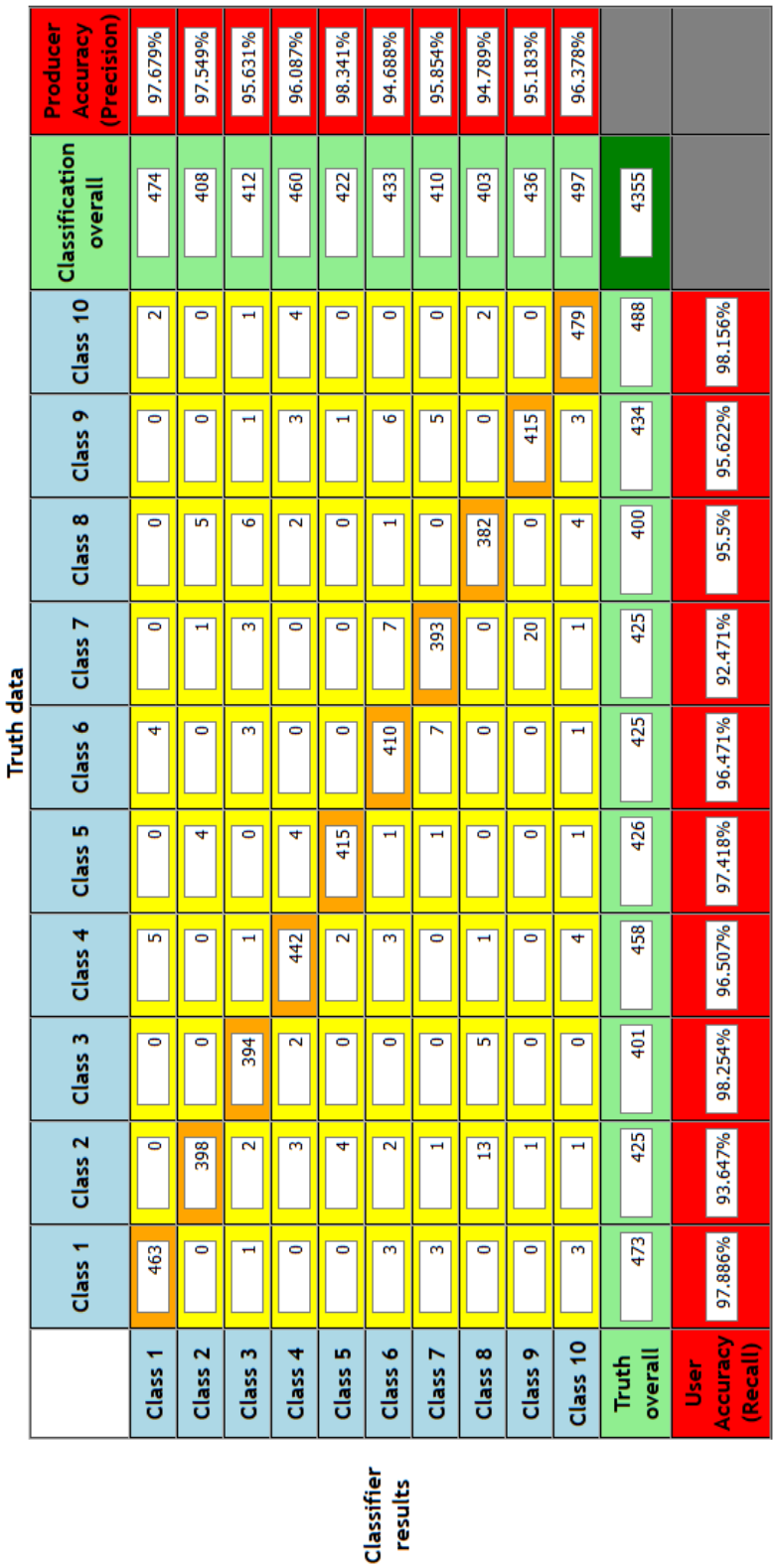
Výsledky algoritmu jsou optimistické. Nalézt se podařilo v obraze až na jeden všechny žetony. Klasifikační model se zmýlil také jen v jediném případě (viz obrazek 4.27), kdy klasifikoval velmi natočený žeton s hodnotou 5 do třídy 6. Naopak případ kdy algoritmus klasifikace naprosto selhal nastal ve chvíli, kdy povrch scény nebyl (tak jak bylo zadefinováno) rovný, nýbrž „vypouklý“ (viz obrázek 4.28). V tomto případě došlo k viditelnému zkreslení žetonů v obraze v závislosti od polohy na scéně.

Algoritmus také poměrně často selhává v případě rozpoznání žetonů, umístěných v odlehlejších částech scény. Často se stává, že segmenty žetonů jsou v těchto místech zdeformované (viz obrázek 4.29). Tento jev je způsoben sníženým rozlišením obrazu vzdálenějších částí scény. Umocněn je však jevem dvojodrazu světla na rozhraní zrcátka.

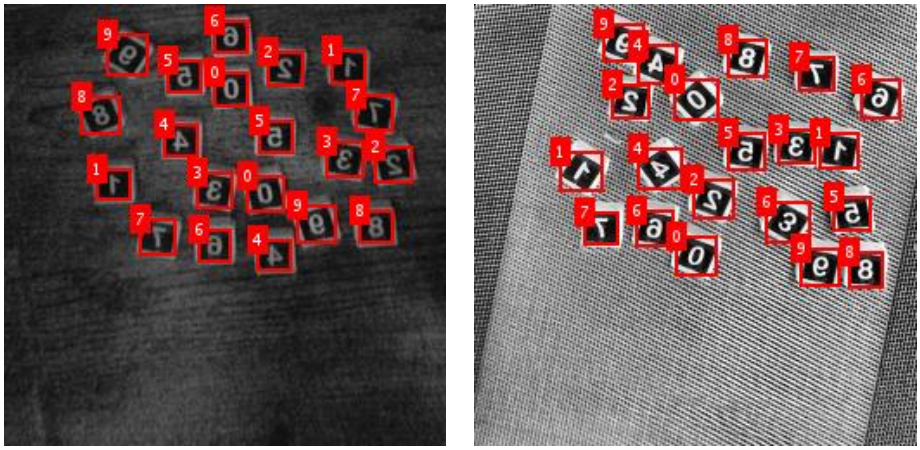
Jako zásadní proto pro další pokračování práce považuji osazení zrcátkového klipu vhodnějším typem zrcátka, ve kterém k dvojodrazu docházet nebude. Ideálně chromovanou ocelí.

Z výstupu testování klasifikátoru v sekci Testování modelu předchozí kapitoly vyplývá, že u některých tříd dochází k záměnám mnohem častěji než u jiných. Tento jev lze vypozorovat z prezentované matice záměn klasifikátoru (obrázek 4.26). Jedná se především o záměnu čísla 7 za číslo 1 a obráceně. Dále pak záměny čísla 6 za číslo 8. K zaměnám jiných čísel dochází řádově méně častěji. Řešením problému může být za prvé použití robustnějšího klasifikačního modelu. Druhou možností je provést úpravu fontu číslic takovým způsobem aby problémové dvojice symbolů od sebe byly více rozlišeny. Na základě této evaluace považuji za vhodné pro případ pokračování ve vývoji provedení následujících kroků:

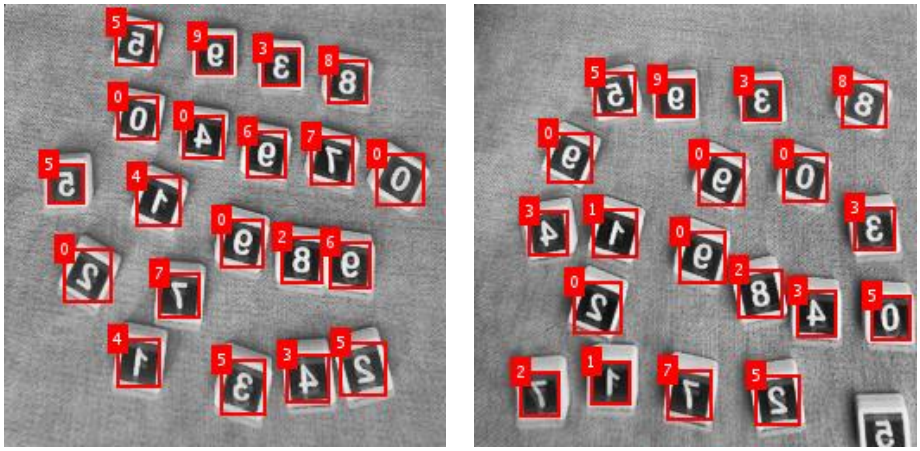
- Osadit klip novým zrcátkem s jedním odrazovým povrchem
- Vhodným způsobem upravit font číslic žetonů
- S novými žetony a zrcátkovým klipem provést nasnímání nových dat
- Otestovat dopad změn na aktuální algoritmus
- Zaměřit se na tvorbu robustnějšího klasifikačního modelu - vícevrstvé konvoluční neuronové sítě



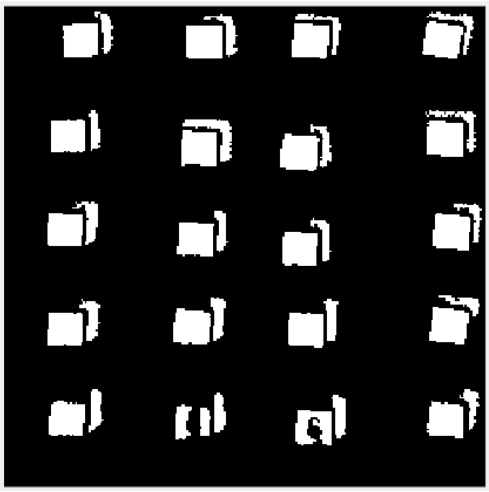
Obrázek 4.26: Metice záměn



Obrázek 4.27: Výstup celého algoritmu



Obrázek 4.28: Selhání klasifikátoru na scéně s vypouklým povrchem



Obrázek 4.29: Nepřesná segmentace žetonu ve vzdálenějších polohách od kamery

Závěr

V úvodní kapitole jsem nastínil jeden z problému dnešního základního vzdělávání. Zmínil jsem Hejného metodu výuky matematiky jako jeden z inovativních a slibných směrů, kterým se dnešní vzdělávání ubírá. Dále jsem se zabýval tím, jak by při výuce matematiky Hejného metodou mohli pomoci moderní technologie. V tomto kontextu jsem nastínil vlastní představu využití rozšířené reality ve výuce a popsal vizi vzdělávací tabletové aplikace poskytující rozšířenou realitu, ke které jsem touto prací chtěl pomalu směřovat. Cílem práce pak bylo základně prozkoumat možnosti řešení takové tabletové aplikace, navrhnout hrubě její podobu a nalézt vhodné algoritmické workflow technik zpracování obrazu, které dokáže v obrazu nalézt definované objekty. Taková funkcionalita by totiž umožnila ovládat aplikaci skrze definované fyzické objekty.

Během práce byly nejprve analyzovány základní potřebné funkce zamýšlené aplikace a možnosti jejího ovládání. Na základě této analýzy bylo vybráno řešení aplikace, kdy je tabletové zařízení usazeno ve speciálním stojanu s nasazeným zrcátkovým klipem na frontální kameře zařízení, díky čemuž je aplikaci umožněno skrze kameru zařízení snímat prostor před tabletem ve kterém bude docházet k interakci s fyzickými žetony potištěnými číselnými symboly. Vybrané řešení bylo podrobeno analýze, která se zabývala především problematikou snímání scény (prostoru před tabletem). Na základě této analýzy byly navrženy 3D modely potřebných HW komponent stojanu, zrcátkového klipu a žetonů, které byly posléze vyrobeny pomocí technologie 3D tisku.

Těžiště celé práce tkvělo ve vývoji SW. Jednalo se o proces hledání vhodného algoritmického workflow zpracování obrazu, které by mohlo být v budoucnu použito pro implementaci zamýšlené aplikace. Proces hledání kýženého algoritmu se ukázal v průběhu práce jako cyklický. Často bylo potřeba experimentovat s různými algoritmy, na základě výsledků pořizovat nová data, ty analyzovat, vhodně zpracovávat a experimenty provádět opětovně s novými parametry a vstupními daty. Jako vhodné se proto ukázalo v rámci procesu vývoje SW postupovat dle metodiky CRISP-DM, která pomohla práci uspo-

řádat a do jisté míry standardizovat.

Výsledkem tohoto procesu je nalezené algoritmické workflow, které splňuje definovaná základní kritéria úspěchu. Workflow algoritmů je implementované ve skriptovacím prostředí nástroje Matlab. V neposlední řadě jsou výstupem hotové hardwarové komponenty potřebné pro realizaci aplikace. Součástí výsledků práce je pak také dataset pořízených obrazových dat, sada skriptů generujících různé další trénovací a testovací datasety, které byly zapotřebí během implementace, a v podobě skriptů připravené prostředí pro další experimentování a případné rozvíjení této práce.

Příloha **A**

Seznam použitých zkratek

HM Hejného metoda

DM Data mining

CRISP-DM Cross industry standard process for data mining

AR Augmented reality

Literatura

[1] LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. *Gradient-based Learning Applied to Document Recognition.* Proceedings of the IEEE. Vol 86, pp. 2278–2324, 1998.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. *"Imagenet classification with deep convolutional neural networks."* Advances in neural information processing systems. 2012.

[3] SONKA, Milan, Vaclav HLAVAC a Roger. Image Processing, Analysis, and Machine vision. Třetí. Spojené státy americké: Thomson learning, 2008. ISBN 0-495-25538-4..

[4] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., et al. *Handwritten Digit Recognition with a Back-propagation Network.* In Advances of Neural Information Processing Systems, 1990.

[5] HLAVNÍ ZJIŠTĚNÍ PISA 2012: *MATEMATICKÁ GRAMOTNOST PATNÁCTILETÝCH* ŽÁKŮ. Praha: Česká školní inspekce, 2013. ISBN 978-80-905632-0-9.

[6] *Vývoj školství na našem území a současný školní* [online]. [cit. 2017-04-05]. Dostupné z: `http://zameklomnice.cz/pdf/implementace/docs/1_rocnik/SOS/PED/vyvoj.pdf`

[7] Svoboda učení: *O co jde* [online]. [cit. 2017-05-08]. Dostupné z: `http://www.svobodauceni.cz/clanek/o-co-jde`

[8] *Tři problémy českého školství* [online]. [cit. 2017-05-08]. Dostupné z: `http://www.ceskaskola.cz/2016/08/daniel-munich-pro-hn-tri-problemy.html`

[9] L.N.Tolstoj a jeho pedagogický odkaz. SPN Praha 1982. s. 191. In CIPRO, M. Galerie světových pedagogů: Encyklopedie prameny výchovy, Druhý svazek, Devatenácté století, s. 339-348.

[10] K. Zuiderveld: *Contrast Limited Adaptive Histogram Equalization.* In: P. Heckbert: Graphics Gems IV, Academic Press 1994

[11] Otsu, N., *"A Threshold Selection Method from Gray-Level Histograms,"* IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66

[12] KOMPRDOVÁ, PH.D., Mgr. Klára. *Rozhodovací stromy a lesy.* Brno: AKADEMICKÉ NAKLADATELSTVÍ CERM, 2012. ISBN ISBN 978-80-7204-785-7.

Příloha **B**

Obsah přiloženého CD

- `Data/` ..... obsahuje veškeré datasety, které byly v práci použity
- `Implementation/` obsahuje veškeré implementované skripty a experimenty
- `Results` ..... výsledky experimentů a testování celého algoritmu