Diplomová práce

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Detekce barev a bankovek pro nevidomé uživatele v prostředí Android

Jan Pechan
Kybernetika a robotika – Systémy a řízení

Leden 2015
Vedoucí práce: Ing. Daniel Novák, Ph.D.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jan Pechan**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Detekce barev a bankovek pro nevidomé uživatele v prostředí Android**

Pokyny pro vypracování:

1. Navrhněte algoritmus pro rozpoznávání barev a bankovek (česká, evropská a dolarová měna).
2. Algoritmus implementujte pro OS Android 4.x.
3. Vyhodnoťte přesnot algoritmu.

Seznam odborné literatury:

[1] A.Hinwood, Bank note recognition for the vision impaired,Australasian Physical & Engineering Sciences in Medicine Volume 29 Number 2, 2006
[2] Hasanuzzaman, F.M, Robust and effective component-based banknote recognition by SURF features, IEEE Wireless and Optical Communications Conference (WOCC), 2011

Vedoucí: Ing. Daniel Novák, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

L.S.

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 20. 1. 2014

Poděkování / Prohlášení

Chtěl bych poděkovat své přítelkyni Yong—An za psychickou podporu, své rodině za finanční a morální podporu, svému vedoucímu práce Danovi Novákovi za trpělivost a cenné rady a v neposlední řadě svému duchovnímu vůdci Aštaru Šeranovi, nechť je jeho kosmická flotila lásky a míru vždy na obzoru.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 5. 1. 2015

.....

Abstrakt / Abstract

Tato práce se zabývá vývojem aplikací na rozpoznávání barev a bankovek pro nevidomé a těžce zrakově postižené uživatele. Obě aplikace jsou vyvíjeny pod mobilní platformou Android a pracují s obrazovou informací získanou z kamery příslušného přístroje. Rozpoznávání bankovek zpracovává nepřetržitý vstup videa a výstupem je měna a hodnota předložené bankovky. Detekce barev pracuje s daty z jediného snímku a vrací název dominantní barvy ve scéně. Obě řešení je možné použít jako samostatné aplikace.

Klíčová slova: detekce barev; rozpoznávání bankovek; zrakově postižení; nevidomí; Android.

This thesis is aimed at developing color and banknote recognition applications for blind and visualy impaired people. Both applications were developed under a platform Android and work with data acquired from device camera. Banknote recognition works with continuous video stream and outputs currency and value of presented banknote. In case of color detection single image is processed and closest name of dominant color in the picture is returned. It is possible to use both solutions as standalone applications.

Keywords: color recognition; banknote recognition; visually impaired; blind; Android.

Title translation: Color and banknote detection for blind users in Android

Obsah /

1 Úvod ... 1
1.1 Rozpoznávání barev ... 1
1.2 Rozpoznávání bankovek ... 2
2 Analýza a návrh řešení ... 5
2.1 Rozpoznávání barev ... 5
2.1.1 Analýza a požadavky na aplikaci ... 5
2.1.2 Technická omezení ... 5
2.1.3 Návrh řešení ... 6
2.2 Rozpoznávání bankovek ... 7
2.2.1 Analýza a požadavky na aplikaci ... 7
2.2.2 Technická omezení ... 7
2.2.3 Návrh řešení ... 7
3 Použité metody a algoritmy ... 9
3.1 Vnímání barev lidským okem ... 9
3.2 Barevný prostor ... 9
3.2.1 RGB barevný prostor ... 10
3.2.2 XYZ barevný prostor ... 10
3.2.3 Lab barevný prostor ... 10
3.2.4 HSL barevný prostor ... 11
3.2.5 Konverze mezi barevnými prostory ... 12
3.3 K-means ... 12
3.4 Nalezení nejbližších K sousedů ... 13
3.5 Detekce příznaků ... 13
3.5.1 FAST ... 14
3.5.2 ORB ... 14
3.6 Deskriptor příznaků ... 14
3.7 Párování příznaků ... 14
3.8 EURion konstelace ... 15
4 Implementace ... 17
4.1 OS Android ... 17
4.1.1 Přístup ke kameře ... 17
4.1.2 AsyncTask ... 17
4.2 OpenCV pro Android ... 17
4.2.1 Mat ... 18
4.3 Rozpoznávání barev ... 18
4.3.1 Třída MainActivity ... 18
4.3.2 Třída LabColor ... 19
4.3.3 Třída KMeans ... 20
4.3.4 Třída ColorProcess ... 20
4.3.5 Třída ColorName ... 20
4.3.6 Trénovací množina dat ... 21
4.3.7 Uživatelské rozhraní ... 21
4.4 Rozpoznávání bankovek ... 21
4.4.1 Pořízení a úprava vzorů bankovek ... 22
4.4.2 Uložení příznaků trénovací množiny v datovém úložišti ... 23
4.4.3 Předzpracování obrazu ... 23
4.4.4 Reprezentace trénovací množiny v běhu programu ... 24
4.4.5 Uživatelské rozhraní ... 24
4.4.6 Algoritmus rozpoznání bankovky ... 24
4.4.7 Použité třídy ... 25
4.4.8 Volba prahů pro určení kvality shod ... 26
5 Testování ... 29
5.1 Rozpoznávání barev ... 29
5.1.1 Test přesnosti rozpoznání ... 29
5.1.2 Osvětlení přisvětlovací diodou ... 29
5.1.3 Umělé osvětlení ... 30
5.1.4 Denní světlo ... 30
5.1.5 Vyhodnocení testu přesnosti rozpoznávání ... 30
5.1.6 Test segmentace dominantní barvy ... 31
5.2 Rozpoznávání bankovek ... 32
5.2.1 Osvětlení velké intenzity ... 32
5.2.2 Osvětlení malé intenzity ... 33
6 Diskuze ... 35
6.1 Rozpoznávání barev ... 35
6.1.1 Project Ara ... 35
6.1.2 Modul pro rozpoznávání barev ... 35
6.1.3 Vylepšení implementované aplikace ... 36
6.2 Rozpoznávání bankovek ... 36
6.2.1 Využití EURion konstelace ... 37
6.2.2 Vylepšení implementované aplikace ... 37
6.2.3 Doporučení pro užívání aplikace ... 37

7 Závěr 39
Literatura 41
A Použité zkratky 43
B Obsah přiloženého CD 45

Tabulky / Obrázky

5.1. Rozpoznání barev – LED přisvětlení. 29
5.2. Rozpoznání barev – umělé osvětlení. 30
5.3. Rozpoznání barev – denní světlo. 31
5.4. Rozpoznání bankovek – osvětlení velké intenzity. 32
5.5. Rozpoznání bankovek – osvětlení malé intenzity. 33

1.1. Elektronické přístroje na rozpoznávání barev. 1
1.2. Hmatové značky na českých bankovkách. 2
1.3. Elektronické přístroje na identifikaci bankovek. 3
1.4. Android aplikace BlindDroid Wallet. 3
2.1. Srovnání obrázků pořízených pro různé hodnoty vyvážení bílé. 6
3.1. Vnímání barev lidským okem. . . . 9
3.2. RGB barevný prostor. 10
3.3. Lab barevný prostor. 11
3.4. HSL barevný prostor. 11
3.5. Příklad algoritmu kNN. 13
3.6. FAST nalezení příznaku. 14
3.7. EURion konstelace. 15
4.1. Graficke schéma aplikace rozpoznávání barev. 19
4.2. Určení achromatických barev z HSL. 20
4.3. Uživatelské rozhraní rozpoznání barev. 21
4.4. Grafické schéma aplikace rozpoznávání bankovek. 22
4.5. Znázornění descriptor-MatMap objektu. 24
4.6. Předzpracování bankovky detekce FAST. 25
4.7. Předzpracování bankovky detekce ORB. 25
4.8. ROC křivky pro jednotlivé bankovky. 26
4.9. ROC křivka pro všechny bankovky. 27
5.1. Vyhodnocení algoritmu segmentace barev. 31
6.1. Project Ara prototyp. 36

Kapitola 1
Úvod

Dle posledních odhadů je na světě 285 milionů lidí se zrakovým postižením, z toho 39 milionů trpí úplnou slepotou [1]. Každodenní život nevidomých je mnohdy odkázán na pomoc třetích osob, případně na pomoc zprostředkovanou skrze specializované pomůcky. Cílem této práce je nabídnout mobilní alternativu v podobě aplikace ke dvěma běžně používaným specializovaným pomůckám (detekce barev a rozpoznávání bankovek).

1.1 Rozpoznávání barev

Rozpoznávání barev je nevidomým užitečné např. při oblékání či praní prádla. Dále může sloužit k primitivnímu rozlišní produktů dle barvy obalu. Na českém trhu je k dostání několik ručních přístrojů na rozpoznávání barev s hlasovým výstupem, např. Colorino[1]) či ColorTest[2]). Takové pomůcky vynikají svou přesností a jednoduchostí použití, avšak vzhledem ke své vysoké ceně jsou běžnému uživateli často nedostupné. Principem získání informace o barvě snímaného povrchu je jeho postupné umělé osvícení několika LED vyzařující světlo na různých vlnových délkách. Po každém osvícení je pomocí fotoprvku zaznamenána intenzita odraženého světla od povrchu a následně jsou sebraná data vyhodnocena. Nevýhodou takového přístupu je, že přístroj musí být pokaždé přiložen těsně ke snímanému povrchu.

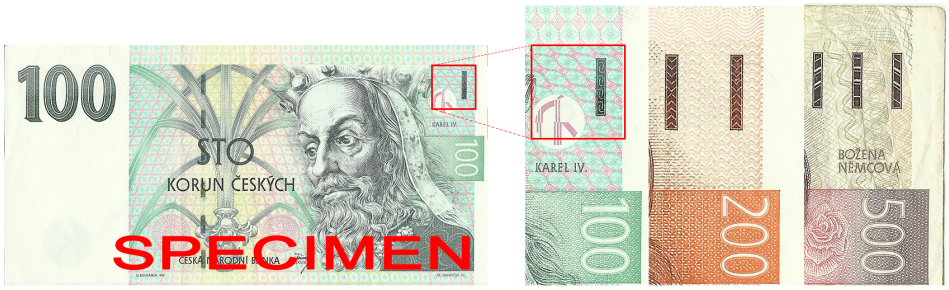
Obrázek 1.1. Elektronické přístroje na rozpoznávání barev. Vlevo přístroj Colorino, vpravo ColorTest.

Ve sféře mobilních aplikací v prostředí Android existuje mnoho aplikací, které umožňují zjistit barvu ze snímané scény. Většina těchto aplikací však slouží jako nástroj pro designéry. V několika případech se jedná o aplikace, které pomáhají lidem trpící daltonismem. Žádná z nabízených aplikací není přímo určena nevidomým, což je dáno všeobecnou nedůvěrou této skupiny k dotykovým telefonům.

[1]) `http://is.braillnet.cz/pomucky_vypis.php?name=Colorino`
[2]) `http://is.braillnet.cz/pomucky_vypis.php?name=ColorTest`

1.2 Rozpoznávání bankovek

Rozpoznávání bankovek je problém, se kterým se nevidomí dostávají do styku každý den. Design českých bankovek je navrhnut tak, aby rozpoznání bankovek nevidomým z části ulehčil. Bankovky o hodnotách 100, 200 a 500 KČ, mají jednotnou šířku, a to 69mm. Jejich délka se od sebe liší o 6mm. Obdobné rozlišení mají i bankovky hodnot 1000, 2000 a 5000 Kč, které sdílejí šířku bankovky 74mm a délka je odstupňována opět po 6mm [2]. Nevidomý tak může určit hodnotu bankovky porovnáním s jinou bankovkou. Samotné určení bankovky bez jiné reference je obtížné. Bankovky sice obsahují hmatové značky, hmatový vjem však není nikterak výrazný a u starších, opotřebovaných bankovek zcela mizí[1]). Osoby s částečným zrakem mohou bankovky rozlišit dle barvy. Nominální hodnoty jsou navíc na bankovkách dostatečně výrazné a kontrastní.



Obrázek 1.2. Hmatové značky na českých bankovách.

V zahraničí jsou návrhy bankovek vystaveny obdobné praxi jako v České republice. Vyjímku tvoří USA, kde bankovky neobsahují žádné hmatové značky, všechny bankovky mají stejnou barvu a navíc mají i identické rozměry. Takové bankovky není nevidomý schopen rozpoznat bez použití specializované pomůcky, nebo druhé osoby. Ačkoliv přepracování designu amerických bankovek bylo odsouhlaseno již v roce 2008, nové bankovky s hmatovými značkamy nemají přijít do oběhu před rokem 2020 [3].

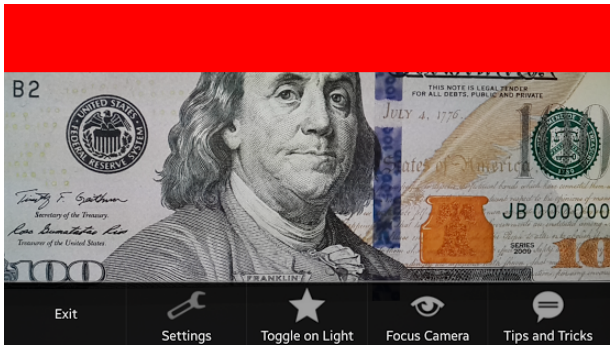
V České republice a Evropské unii patří k nejběžnějším pomůckám při rozpoznávání bankovek speciální šablona, do které se bankovka zasune a přehne. Díky rozdílným délkám bankovek lze v závislosti na tom, kam až bankovka dosáhla odečíst příslušnou hodnotu. V USA využití stejného principu není možné, jelikož mají všechny bankovky stejné rozměry. Oficiální doporučení Americé nadace pro nevidomé je využívat různé přehnutí bankovek k jejich identifikaci [4]. Tato metoda však neumožňuje určit bankovku, o které není apriorně známo, jaké je hodnoty. Další možností identifikace jsou elektronické rozpoznávače bankovek s hlasovým výstupem jako např. přístroj Money Talks[2]), či NoteTeller2[3]).

[1]) `http://bit.ly/1xIBEMO`
[2]) `http://www.maxiaids.com/products/7870/Tell-Money-Talking-Money-Identifier.html`
[3]) `http://www.brytech.com/noteteller/`

Obrázek 1.3. Elektronické přístroje na identifikaci bankovek. Vlevo Money Talks, vpravo NoteTeller2.

Rozpoznávání bankovek na platformě Android je většinou integrováno v některé z aplikací, které se zabývají rozpoznáváním obrazu. Příkladem takové aplikace je Google Goggles[1]). Tato aplikace odešle obraz zachycený kamerou, případně obraz z úložiště, na vzdálený server kde se provede rozpoznání a posléze vrátí výsledek zpět do mobilního telefonu. Rozpoznávání bankovek však není primárním určením aplikace. Nevýhodou podobných aplikací je nutnost internetového připojení.

Jediná plně funkční aplikace sloužící k rozpoznávání bankovek určená nevidomým je v Obchodě Play Blind-Droid Wallet[2]). Aplikace pracuje bez nutnosti připojení k internetu a podporuje řadu měn včetně USD a EUR. Rozpoznávání CZK nicméně chybí.



Obrázek 1.4. Ukázka aplikace Blind-Droid Wallet.

[1]) https://play.google.com/store/apps/details?id=com.google.android.apps.unveil
[2]) https://play.google.com/store/apps/details?id=com.mirwebsistem.currencyeyes

Kapitola 2
Analýza a návrh řešení

V této kapitole jsou formulovány požadavky na aplikace a základní úvahy, ze kterých byl vytvořen návrh řešení obou aplikací. Je zde formulováno i samotné řešení se základními myšlenkami a postupy, které jsou rozvedeny v následujících kapitolách. Dále jsou zde uvedena omezení, daná vlivem prostředí a platformou, na které jsou aplikace vyvíjeny.

2.1 Rozpoznávání barev

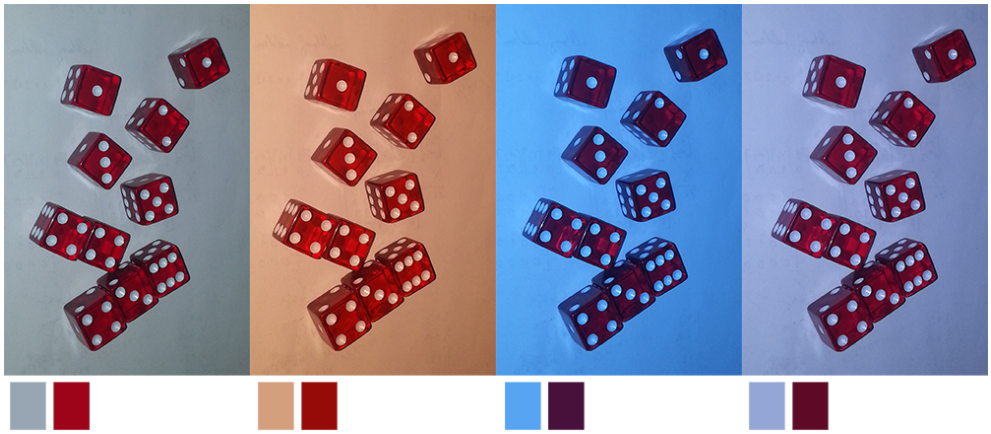
2.1.1 Analýza a požadavky na aplikaci

Cílem aplikace rozpoznávání barev je zprostředkovat nevidomému uživateli informaci o barvě objektu či scény, která se nacházejí v jeho blízkosti. Jediným vstupem, který dokáže zprostředkovat obrazovou informaci o okolní scéně, je u běžně dostupných chytrých telefonů kamera, se kterou tedy aplikace pracuje. Ze snímaného obrazu scény se vyhodnocuje pouze informace obsažená uprostřed obrazu, jelikož nejsou předem známé rozměry objektu či vzdálenost od daného objektu. Z vybrané oblasti se zjistí dominantní barva, která se následně prezentuje uživateli, a to jak prostřednictvím zvukového výstupu, tak zároveň obrazově (pro uživatele, kteří trpí zrakovým postižením, ale nejsou nevidomí).

Aplikace by měla dokázat rozeznat základní barvy – černá, bílá, modrá, zelená, červená, žlutá, hnědá, oranžová, fialová, růžová, šedá, a to i jejich intenzity (světlá či tmavá). Výstup by měl reflektovat snímanou skutečnost a snažit se omezit záporné vlivy zapříčiněné kvalitou snímače použitého přístroje.

2.1.2 Technická omezení

Výsledek rozpoznání barvy je silně závislý na typu a intenzitě osvětlení scény a schopnosti telefonu dané podmínky kompenzovat. Klíčové je správné nastavení vyvážení bílé, čehož však nelze vždy dosáhnout, a to zejména v aplikaci, kdy není předem známé nic o prostředí, ve kterém se bude využívat. Částečně lze závislost na vyvážení bílé potlačit použitím přisvětlovací diody, pro kterou je v telefonu uložena příslušná korekce. Přisvětlovací diodou však nedisponují všechna zařízení, navíc pokud je ambientní osvětlení intezivnější než světlo diody, může mít použití diody negativní efekt, ve smyslu použití špatného vyvážení bíle. Dalším neduhem, který užití přisvětlovací diody vnáší do řešení, jsou odlesky od lesklých předmětů. Při zachycení odlesku kamerou může dojít ke zkreslení či absolutní záměně rozpoznané barvy.



Obrázek 2.1. Srovnání obrázků pořízených pro různé hodnoty vyvážení bílé.

Použitá kamera, obzvláště u levnějších chytrých telefonů, ovlivňuje parametry nasnímaného obrazu — šum, barevnou věrnost a ostrost. Ostrost v případě rozpoznávání barev nemá na výsledek žádný vliv, šum lze potlačit do mezí, kdy kvalitu určení barvy neovlivňuje, nicméně barevná věrnost obrazu je parametr, který se ovlivňuje ztěží. Celé řešení je komplikováno rozmanitostí zařízení, které využívají OS Android, a tedy nelze aplikaci přizpůsobit jednomu zařízení, či třídě zařízení.

2.1.3 Návrh řešení

Ze sejmuté scény se nejprve uprostřed určí čtvercová oblast (šířka čtverce je 25% kratšího strany snímané scény). Vybraná oblast se rozdělí do několika klusterů pomocí algoritmu K-means, ze kterých se vybere ten, který má nejvíce prvků. Pomocí tohoto rozdělení lze vybrat průměrnou barvu dominantní barvy v dané oblasti. Tento krok je důležitý pro odstranění odlesků, šumu, vedlejších barev (např. potisk na oblečení, text na etiketě), či jiných nechtěných rušení, které se mohou ve scéně objevit.

Barevný prostor RGB, ve kterém jsou zpracovány bitmapy v OS Android, není vhodný pro porovnávání podobnosti dvou barev z pohledu precepce barev lidským okem, proto je pro výpočet vzdáleností mezi jednotlivými barvami využit barevný prostor Lab, který reflektuje vnímání barev lépe. Pro rychlejší průběh výpočtu dominantní barvy nejsou vzdálenosti počítány mezi všemi pixely z daného čtvercového výběru, ale pouze mezi náhodně vybranými pixely, které tvoří reprezentativní vzorek.

Po vybrání dominantní oblasti, a tedy i dominantní barvy, která je průměrem (spočítaným v barevném prostoru Lab) všech barev, které oblast obsahuje, se provede krok, ve kterém je dané barvě přiřazen slovní název. Toto přiřazení probíhá ve dvou barevných prostorech. Černá, bílá a šedá barva se určuje ze saturace a světlosti v prostoru HSL. Postup pro určení těchto barev je v jistém smyslu heuristikou, která byla odvozena z testování na různých zařízení. Bílá barva je často zaznamenána jako světle šedá, či jinak světlá barva s nízkou hodnotou saturace. Černá barva bývá často na snímaném obrázku zaznamenána jako tmavě hnědá, či jinak tmavá barva s nízkou saturací. Pokud není barva vyhodnocena z prostoru HSL, proběhne druhá část přiřazení barvy - algoritmus kNN (K nejbližších sousedů) oproti uložené databázi barev. Tento algoritmus najde v databázi K barev, které jsou nejvíce podobné zkoumané barvě. Výsledná barva je pak ta, která má majoritní zastoupení v množině vybraných barev. Tento algoritmus opět pracuje v barevném prostoru Lab.

2.2 Rozpoznávání bankovek

2.2.1 Analýza a požadavky na aplikaci

Cílem aplikace rozpoznávání bankovek je určit hodnotu předložené bankovky. Rozhraním pro získání informace o bankovce, stejně jako v případě určení barvy, je kamera telefonu. Rozpoznávání by mělo pracovat v nepřetržitém režimu, tedy scéna je snímána kontinuálně a vyhodnocován je ideálně každý snímek. Pokud je snímek vyhodnocen ve smyslu, že obsahuje známou bankovku, prezentuje se výsledek uživateli prostřednictvím hlasového výstupu. Zároveň se detekce zastaví do té doby, než vydá uživatel pokyn pro detekci další bankovky.

Aplikace by měla být robustní vůči natočení či pomuchlání bankovky a měla by dokázat rozpoznat bankovku ze snímku i v případě, že není ve scéně celá (což může být způsobeno např. částečným zakrytím bankovky rukou, ve které je držena). Důležitým požadavkem je také minimum nesprávně rozpoznaných (špatně označených) bankovek.

2.2.2 Technická omezení

V aplikaci rozpoznávání bankovek se uplatňují stejná omezení daná přístrojem jako v aplikaci na rozpoznávání barev. Tato omezení však zaujímají jinou roli. Vyvážení bílé a barevná přesnost mají na výsledek rozpoznání bankovky mnohem menší vliv, naopak šum a ostrost mohou v závislosti na použitém algoritmu výsledek značně ovlivnit. Některé přístroje s OS Android mají pevně nastavenou ohniskovou vzdálenost, takže vzdálenost, ve které je scéna správně zaostřena, může být menší než u jiných přístrojů. Takové přístroje mají také definovanou větší vzdálenost od snímače, kde je scéna správně zaostřena. Nízká intenzita osvětlení může rozpoznání ovlivnit negativně v případě, že přístroj nedisponuje přisvětlovací diodou a kamera nedokáže v takových podmínkách zachytit jemné detaily obrazu.

2.2.3 Návrh řešení

Před samotnou detekcí bankovek pomocí aplikace je nutné shromáždit trénovací data bankovek, podle kterých se bude určovat, zda snímaná scéna obsahuje bankovku nějaké měny a její hodnotu. Jako tato data poslouží ofocené bankovky různých měn, které budou dále upraveny (například zbaveny sériového čísla) tak, aby reprezentovali co největší množinu bankovek stejné měny a stejné hodnoty. Z těchto připravených obrazů se na základě výsledného algoritmu detekce a popisu příznaků extrahují příznaky, které jedoznačně reprezentují jednotlivé bankovky. Podobné řešení bylo navrhnuto v pracích [5] a [6]. V aplikaci tedy budou uloženy pouze data extrahovaná z jednotlivých bankovek v podobě příznaků a nikoliv celé bankovky, což by mohlo vést k porušení zákonu o padělání a pozměňování peněz.

Vstupní obraz v závislosti na jeho kvalitě projde předzpracováním. Toto předzpracování může obsahovat algoritmy na úpravu histogramu, například jeho ekvalizaci. Dále může být obraz zpracováván pouze v jednom barevném kanálu, či rozmazán gausovským rozostřením, v závislosti na potřebách dalších operací se sejmutým obrazem. Z obrazu jsou extrahovány příznaky algoritmem FAST nebo ORB, které jsou následně porovnány oproti příznakům získané ze vzorových bankovek. Porovnávání se provádí vždy v rámci jedné měny a to v zájmu rychlejší detekce bankovek, tak i kvůli snížení počtu falešně pozitivních shod. Výsledek je po úspěšném rozpoznání prezentován uživateli hlasovým výstupem.

Nastavený práh pro počet podobností příznaků indikující pozitivní shodu mezi bankovkou ve scéně a některou vzorovou bankovkou je určen na základě ROC křivky s kritériem přiklánějícím se k méně falešně pozitivním výsledkům.

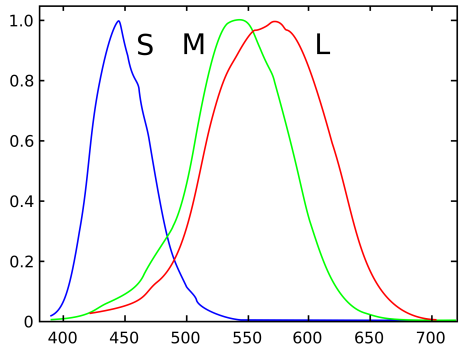
Druhým navrhovaným řešením je využití EURion konstelace, která je přítomná na většině současných bankovek. Pokud je z obrazu využit pouze modrý kanál, jsou více zřetelná jednotlivá mezikruží, které jsou součástí této konstelace. Bankovky kromě několika EURion konstelací obsahují i větší množství mezikruží, která jsou rozesety v jejich okolí. Vhodnou detekcí pak lze z bankovek dostat příznaky reprezentující právě tyto konstelace. Nevýhodou této metody je, že potřebuje kvalitní obraz snímané bankovky, což zahrnuje dobré zaostření, dostatečné osvětlení a kvalitní kameru.

Kapitola 3
Použité metody a algoritmy

V této kapitole jsou popsány metody a prinicipy, které byly použity při konstrukci řešení zadaných úloh.

3.1 Vnímání barev lidským okem

Lidské oko obsahuje dva typy receptorů – čípky a tyčinky, pomocí kterých mozek interpretuje vnímané barvy. Čípky umožňují vnímání barev. Lidské oko obsahuje tři druhy – krátký S, střední M a dlouhý L. Každý z čípků je citlivý na různé vlnové délky světla se špičkami intenzit 420nm – 440nm, 534nm – 545nm a 564nm – 580nm, které se dají hrubě interpretovat jako modrá, zelená a červená barva. Při malé intenzitě světla je schopnost rozeznávat barvy pozbyta a hlavní roli při vnímání obrazu přebírají tyčinky, které jsou citlivé na intenzitu světla a nedokáží vnímat barvy.



Obrázek 3.1. Vnímání barev skrze čípky lidského oka.

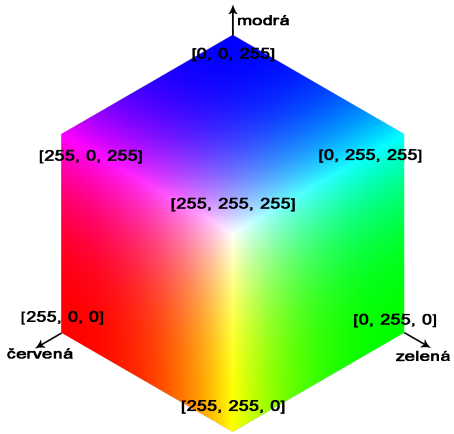
Ze znalosti tří základních barev, na které reaguje lidské oko byly sestaveny základní barevné prostory RGB a XYZ, kde každá z tří barev reprezentuje jednu osu a všechny ostatní barvy jsou dány jejich funkcí.

3.2 Barevný prostor

Barevný prostor je n-dimenzionální geometrický prostor který obsahuje popis vymezené množiny barev (gamutu), většinou takových, které jsou vnímatelné lidským okem. Každá barva v takovém prostoru je reprezentována bodem, který je popsán barevným modelem. Barevný model je matematickým modelem, který kombinuje základní barevné komponenty příslušné použitému barevnému prostoru, kde je výsledkem hledaná barva. Příkladem barevného modelu může být aditivní model, který kombinuje tři barvy, běžně používaný v počítačové grafice. Kombinací všech barev v maximální intenzitě vznikne bílá, míšením základních barev s rozdílnými intenzitami vznikají barvy daného prostoru a absence základních barev produkuje černou. Opačným přístupem je subtraktivní model používaný pro tisk, kde smícháním všech základních barev vznikne černá.

3.2.1 RGB barevný prostor

RGB barevný prostor je aditivní barevný prostor založený na třech základních barvách - červené **R**, zelené **G** a modré **B**. V počítačové grafice je nejrozšířenějším prostorem RGB ten, který reprezentuje každou z třech základních barev diskrétně číslem o rozsahu 8 bitů. Gamut RGB tedy pokrývá všechny barvy smíchatelné ze tří základních barev na prostoru o velikosti 24 bitů.



Obrázek 3.2. Znázornění RGB barevného prostoru.

Nevýhodou barevného prostoru RGB je, že barvy, které jsou si vizuálně podobné se nemusejí v prostoru nalézat v menší vzdálenosti, než barvy které si podobné nejsou. To představuje problém v navrhovaném algoritmu pro nalezení dominantní barvy ve snímaném obrazu. Dalším problémem tohoto prostoru je komplikované určení černé a bílé barvy, pokud přistupujeme k identifikaci barvy s premisou, že šedá barva v určité intenzitě reprezentuje černou a bílou, případně že černá i bílá mohou být ve scéně zobrazeny jako chromatické barvy (například kvůli špatnému vyvážení bílé, či nevhodnému osvětlení).

3.2.2 XYZ barevný prostor

Barevný prostor XYZ, nebo také CIE 1931 [7], byl zaveden pro věrnou reprezentaci barev na základě struktury lidského oka a reflektuje všechny barevné vjemy, které je lidské oko schopno pojmout. Ve své podstatě je analogií k S, M a L čípkům, které se nacházejí v lidském oku. Z obrázku 3.1 je patrné, že rozsahy vlnových délek, na kterých jsou čípky reaktivní, se vzájemně překrývají. Pokud se v tomto smyslu v barevném prostoru reprezentují barvy jako kombinace modré, zelené a červené (jako v případě RGB prostoru), musí být přihlednuto k onomu překrytí vnímaných spekter, neboť některé barevné kombinace pak nejsou možné.

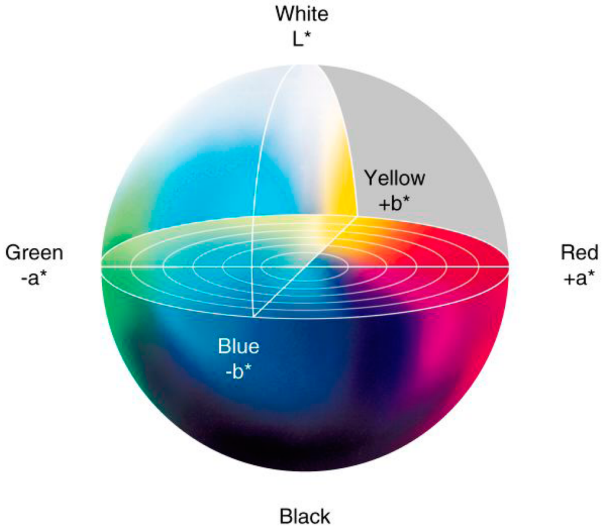
XYZ model klade důraz na to, že lidské oko vnímá zelenou barvu (M čípek) ve scéně jako nejintenzivnější. Z tohoto důvodu je této barvě přiřazen význam jasu a je reprezentován **Y** složkou. **Z** složka je úměrná vjemu S čípku a **X** složka je lineární kombinací zbylých. Barevný tón může být tedy vyjádřen pouze za pomocí **X** a **Z**.

Tento barevný prostor v implementaci aplikace rozpoznávání barev slouží jako meziprostor, skrze který se provádějí konverze do ostatních barevných prostorů.

3.2.3 Lab barevný prostor

Ačkoliv je barevný prostor XYZ věrnou reprezentací všech barev vnímatelných lidským zrakem, pro aplikaci ve které je nutné porovnávat barvy mezi sebou a zjišťovat jejich

vzdálenosti na základě psychologického vnímání barev je ekvivalentní RGB barevnému prostoru. Z tohoto důvodu byl zaveden nový barevný prostor Lab [7].



Obrázek 3.3. Znázornění Lab barevného prostoru.

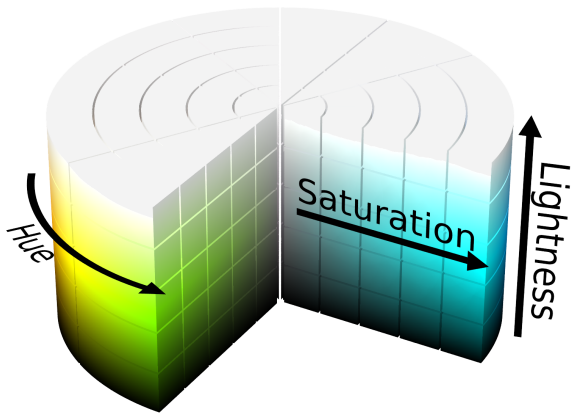
Gamut Lab, stejně jako XYZ, pokrývá všechny barvy viditelné lidským okem. Navíc lze v tomto reprezentovat rozdíl mezi dvěma barvami D_{12} jako jejich euklidovskou vzdálenost

$$D_{12} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}. \tag{3.1}$$

V Lab barevném prostoru **L** reprezentuje světlost, **a** reprezentuje zelenou/červenou barvu a **b** žlutou/modrou, v obou případech se jedná o navzájem komplementární barvy.

3.2.4 HSL barevný prostor

HSL barevný je reprezentaci RGB barevného prostoru ve válcové souřadnicové soustavě. Úhel kolem středové vertikální osy odpovídá odstínu barvy **H**. Tento úhel je získán natočením RGB krychle a její následnou projekcí na rovinu. Vznikne tak hexagon, jehož deformací získáme kružnici, která slouží jako obvod základny válce HSL prostoru. Vzdálenost bodu od středové vertikální osy je reprezentací saturace **S** a poslední souřadnice ve směru vertikální osy udává světlost **L**.



Obrázek 3.4. Znázornění HSL barevného prostoru.

3.2.5 Konverze mezi barevnými prostory

V aplikaci pro rozpoznávání barev jsou prováděny převody barev [1]) z RGB do Lab a zpět a převod z RGB do HSL. Převod mezi RGB a Lab nelze provést přímo a je nutné využít barevného prostoru XYZ. Pro převod z RGB do XYZ je nutné znát převodní matici **M** odpovídající referenční bílé barvě pro RGB prostor, v tomto případě pro sRGB prostor. Dále je nutné převést kompandované kanály (**R**, **G**, **B**) na lineární (**r**, **g**, **b**). Výsledná převodní rovnice je potom

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \tag{3.2}$$

Převod z XYZ do RGB je analogický s opačným pořadím operací. Nejdříve se získají lineární kanály **r**, **g**, **b** pomocí inverzní matice **M** a následně se zkomapndují.

Pro konverzi z XYZ do Lab je nutné znát reference bílé v barevném prostoru XYZ. Převodní rovnice jsou potom dány jako

$$L = 116 f_y - 16, \tag{3.3}$$

$$a = 500(f_x - f_y), \tag{3.4}$$

$$b = 200(f_y - f_z), \tag{3.5}$$

kde f_x, f_y a f_z jsou dány poměrem hodnot **X**, **Y** a **Z** vůči jejich referencím bílé.

Převod z Lab do XYZ je opět anaglogický, jako tomu bylo v případě konverze z RGB do XYZ.

Konverze RGB do HSL vychází z konstrukce HSL barevného prostoru. **R'**, **G'**, **B'** jsou normalizované hodnoty RGB na rozsah $< 0, 1 >$. C_{max} a C_{min} jsou maximální a minimální hodnoty škálovaného RGB a $\Delta = C_{max} - C_{min}$. Barevný odstín H vychází z promítnutého hexagonu a je dán jako

$$H = \begin{cases} \frac{\pi}{3}(\frac{G'-B'}{\Delta} mod 6); C_{max} = R' \\ \frac{\pi}{3}(\frac{B'-R'}{\Delta} + 2); C_{max} = G' \\ \frac{\pi}{3}(\frac{R'-G'}{\Delta} + 4); C_{max} = B' \end{cases} \tag{3.6}$$

Zbylé hodnoty pro saturaci a světlost jsou dány jako

$$S = \begin{cases} 0; \Delta = 0 \\ \frac{\Delta}{1-|2L-1|}; \Delta \neq 0 \end{cases} \tag{3.7}$$

$$L = \frac{(C_{max} + C_{min})}{2}. \tag{3.8}$$

3.3 K-means

Algoritmus K-means slouží k rozdělení množiny dat M do předem specifikovaného počtu shluků K. Přiřazení data do příslušného shluku se řídí optimalizací stanoveného kritéria. Pro rozdělení množiny barev do shluků je kritériem minimalizace euklidovské vzdálenosti zkoumané barvy v barevném prostoru Lab od průměrné barvy shluku, která se počítá ze všech barev aktuálně přiřazených příslušnému shluku. Algoritmus se dá popsat ve čtyřech krocích:

[1]) `http://www.brucelindbloom.com/index.html?Equations.html`

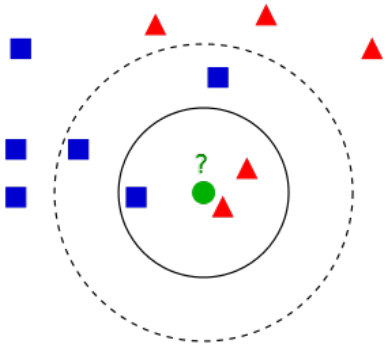
1. Určí se výchozí hodnoty průměrů barev K shluků. Tyto hodnoty lze určit například náhodným výběrem K barev z množiny dat, případně lze vycházet z apriorní známosti rozložení barev v množině dat a průměry určit na základě nějaké heuristiky.
2. Všechna data se umístí do jednoho z K shluků, a to na základě minimalizace euklidovské vzdálenosti průměru barev shluku a testované barvy.
3. Pro každý shluk se vypočítá nová hodnota průměru barev ze všech barev v něm obsažených.
4. Opakují se kroky 2 a 3 do té doby, než je nalezeno optimální rozložení.

Algoritmus je náchylný na volbu počátečních hodnot průměru jednotlivých shluků a je jím zajištěno nalezení pouze lokálního minima optimalizovaného kritéria.

3.4 Nalezení nejbližších K sousedů

Nalezení K nejbližších sousedů, zkráceně kNN, je metoda strojového učení s učitelem, která přiřadí prvek z neznámé třídy do některé z předem definovaných tříd na základě trénovací množiny dat. Trénovací množna obsahuje n-rozměrné vektory, (například barvy v některém z barevných prostorů), kde každý z vektorů je členem jedné z předem definovaných tříd (jméno barvy).

V samotné klasifikaci je identifikovaný prvek vložen do prostoru trénovacích dat a hledá se K jeho nejbližších sousedů dle stanovené metriky (euklidovská vzdálenost pro barevný prostor Lab). Po nalezení jeho K sousedů je zařazen do té třídy, která se nejčastěji vyskytovala mezi K nalezenými sousedy.



Obrázek 3.5. Příklad nalezení K nejbližších sousedů ve dvourozměrném prostoru.

Na obrázku 3.5 je příklad klasifikace prvku ve dvojrozměrném prostoru s euklidovskou metrikou. Pro parametr K=3 je zelený, identifikovaný, prvek přiřazen do třídy červených a pro K=5 je přiřazen do třídy modrých prvků.

Parametr K je zvolen ze znalosti trénovací množiny. Pro velké parametry K je potlačen šum (nesprávně zařazené prvky) trénovací množiny, ale hranice mezi jednotlivými třídami jsou méně výrazné. Pro hodnotu parametru K=1 je algoritmus zredukován na algoritmus hledání nejbližšího souseda.

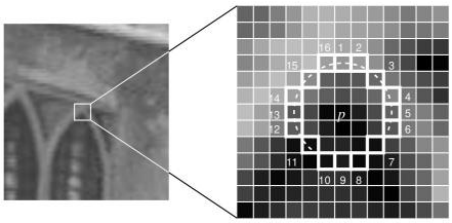
3.5 Detekce příznaků

Detekce příznaků je proces, ve kterém se v obrázku identifikují významné body - příznaky. Těmito významnými body mohou být hrany, rohy, bloby a jiné elementy, v závislosti na použitém detektoru.

3.5.1 FAST

FAST (Features from Accelerated Segment Test) [8] je hranový detektor, u kterého je prioritou rychlost detekce příznaků. Jednotlivé příznaky jsou detekovány na základě testu, tkerý je proveden nad všemi pixely zpracovávaného obrázku. Operace nad jedním pixelem vycházejí z následujícího postupu:

- Vybere se pixel, u kterého se chce určit zda je příznakem a určí se jeho intenzita I_p.
- V okolí zkoumaného pixelu se vybere 16 pixelů uspořádaných do kružnice 3.6.
- Pokud na dané kružnici existuje množina dvanácti dotýkajících se bodů, které jsou tmavší než $I_p - t$ nebo světlejší než $I_p + t$, kde t je předem vhodně zvolený práh, zkoumaný pixel je považován na příznak.



Obrázek 3.6. Výběr 16 pixelů a jejich porovnání oproti intenzitě středového pixelu s přihlédnutím k určenému prahu.

V FAST variantě se místo zkoumání všech 16 pixelů zkoumají pixely na pozicích 1, 9, 5 a 13. Pokud alespoň 3 z daných bodů jsou tmavší než $I_p - t$ nebo světlejší než $I_p + t$ je pixel podroben úplnému testu a následně je rozhodnuto, zda se jedná o příznak.

Nevýhodou tohoto detektoru je jeho náchylnost na otočení detekovaného obrazu a tedy nenalezení vždy stejných příznaků při natočení snímané scény či kamery.

3.5.2 ORB

ORB (Oriented FAST and Rotated BRIEF) [9] je detektorem, který byl navrhnut jako bezplatná alternativa k patentovaným detektorům SIFT [10] a SURF [11]. Je součástí volně šiřitelné knihovny OpenCV, která je využita v algoritmu rozpoznávání bankovek.

ORB vychází z detektoru FAST, který nenese informaci o orientaci příznaků a rozšiřuje ho právě o tuto vlastnost. Orientace se spočte jako vektor daný středem detekované hrany a váženým geometrickým středem intenzit pixelů v oblasti okolo detekované hrany. Další vlastností detektoru ORB je jeho nezávislost na měřítku detekovaného obrazu.

3.6 Deskriptor příznaků

Deskriptor příznaků je objekt, který obsahuje popis všech detekovaných příznaků s jejich vlastnostmi. V případě detektoru FAST se tedy jedná pouze o vektor nalezených hran bez jiných informací. Pro detektor ORB je informace o příznaku rozšířena o jeho orientaci. Pro záznam orientace je využito 12 úhlů odstupňovaných po $\frac{2\pi}{30}$.

3.7 Párování příznaků

Proces, ve kterém se porovnávají detekované příznaky ze vzorovéhoho obrazu oproti příznakům detekovaným ve zkoumané scéně, se nazývá párování příznaků. Při párování se počítají vzdálenosti mezi příznaky vzoru a zkoumaného obrazu a nejlepší shody se

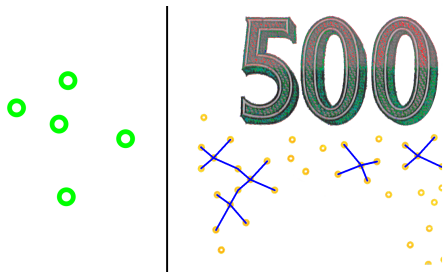
zaznamenávají. Rozhodnutí, zda se jedná o skutečnou shodu, je poté založeno na prahu, který odděluje dobré shody od špatných.

Jedním z přístupům pro párování příznaků je tzv. párování hrubou silou. Každý příznak z jedné množiny se porovnává postupně s každým příznakem z druhé množiny. Tento přístup dává dobré výsledky, jelikož jsou zohledněny všechny možnosti, ale pro velké množiny příznaků je pomalý.

Druhým přístupem je využití FLANN (Fast Library for Approximate Nearest Neighbors), která obsahuje algoritmy optimalizované pro rychlou detekci nejbližších sousedů pro velké množiny příznaků a pro příznaky s velkými dimenzemi. FLANN na velkých datasetech dává lepší časové výsledky v porovnání s párováním hrubou silou.

3.8 EURion konstelace

EURion konstelace [12] je seskupení pěti mezikruží dle předem stanoveného vzoru používané na většině moderních bankovkách napříč měnami. Záměrem tohoto vzoru je jednoduchá identifikace přítomnosti bankovky v obraze a využívá se výhradně pro zamezení reprodukce bankovek. Většina moderních barevných tiskáren a fotokopírek je schopna tento vzor odhalit a nedovolí bankovky reprodukovat. EURion konstelace je většinou vytištěna ve žluté barvě a je nejvíce viditelná v modrém barevném kanálu. Každá bankovka jich obsahuje několik, společně s jinými kombinacemi mezikruží, které nejsou součástí EURion konstelace.



Obrázek 3.7. Vlevo EURion konstelace, vpravo ukázka na části 500CZK.

Kapitola 4
Implementace

V této kapitole jsou popsány technické prostředky použité k implementování obou aplikací i samotná jejich implmentace.

4.1 OS Android

Obě aplikace jsou implementovány pro OS Android v minimální verzi 4.0 (API Level 14). Android je operační systém vyvinutý společností Google, který se používá především na mobilních dotykových zařízeních.

4.1.1 Přístup ke kameře

Přístup ke kameře zařízení a její ovládání zprostředkovává Android skrze framework Camera. Tento framework umožňuje nastavit parametry kamery, jako například rozlišení snímaného obrazu, typ kompenzace vyvážení bíle, vynucení zaostření kamery či zapnutí a vypnutí přisvětlovací diody. Aby bylo možné přistupovat ke kameře, je nutné v Android Manifestu (xml soubor prezentuje základní informace o aplikaci OS Android) povolit přístup k hardwaru kamery.

Přímo skrze framework Camera je v aplikaci na rozpoznávání barev zprostředkován kontinuální náhled na scénu a sejmutí scény v okamžiku detekce. Dále je použit pro nastavení vyvážení bílé a v případě, že zařízení disponuje přisvětlovací diodou, i k zapnutí přisvětlení.

4.1.2 AsyncTask

Android vykonává operace které modifikují UI na jednom vlákně (UI vlákno). Pokud je na tomto vlákně prováděn výpočetně náročný úkon, UI aplikace zamrzne do té doby, než je tento úkon ukončen. Proto Android nabízí pro implementaci asynchronního chování třídu AsyncTask, která umožňuje provádět operace na pozadí a publikovat výsledky do UI vlákna bez potřeby správy vláken či handlerů.

4.2 OpenCV pro Android

OpenCV (Open Source Computer Vision) [1]) pro Android je knihovna, která rozšiřuje Android SDK o prostředky umožňující zpracování snímaného obrazu v reálném čase nástroji implementující počítačové vidění. Tato knihovna je volná jak pro akademické účely, tak pro komerční využití.

OpenCV je použita při řešení úlohy rozpoznávání bankovek, a to zejména její implementované třídy pro detekci a porovnávání příznaků. Tato knihovna implementuje vlastní framework pro práci s kamerou zařízení. Po implementaci je po každém sejmutém snímku scény volána metoda onCameraFrame, ve které je argumentem snímek ze

[1]) `http://opencv.org/`

scény, který je připraven pro zpracování a návratovou hodnotou je obrázek ve formátu Mat, který je zobrazen v příslušném View.

Pro spuštění aplikací, které využívají knihovnu OpenCV je nutné, aby byla na zařízení nainstalována aplikace OpenCV manager [1]).

4.2.1 Mat

Třída Mat je základním prvkem zpracování obrázků v OpenCV. Skládá se ze dvou základních částí:

- Matice obsahující data o pixelech popisovaného obrazu, které mohou mít různé velikosti v závislosti na jejich typu.
- Hlavička matice, popisující vlastnosti matice, jako je její velikost či typ dat v ní obsažených.

4.3 Rozpoznávání barev

Návaznost jednotlivých tříd a průchod aplikací jsou vyobrazeny na obr. 4.1. V následujících několika podsekcích budou rozepsány funkce jednotlivých tříd a dalších komponent detailně.

4.3.1 Třída MainActivity

Třída **MainActivity** rozšiřuje třídu Activity z API Android, která je základním prvkem pro aplikace pod OS Android. Aktivita, jejíž vzhled reprezentuje XML soubor popisující rozložení prvků prezentovaných uživateli, reprezentuje jednu obrazovku v rámci aplikace. **MainActivity** obsahuje framework Camera, která spravuje kameru zařízení a její nastavení. Dále jsou v této třídě asynchronně prováděny výpočty rozpoznávání barev a následně jsou výsledky prezentovány uživateli.

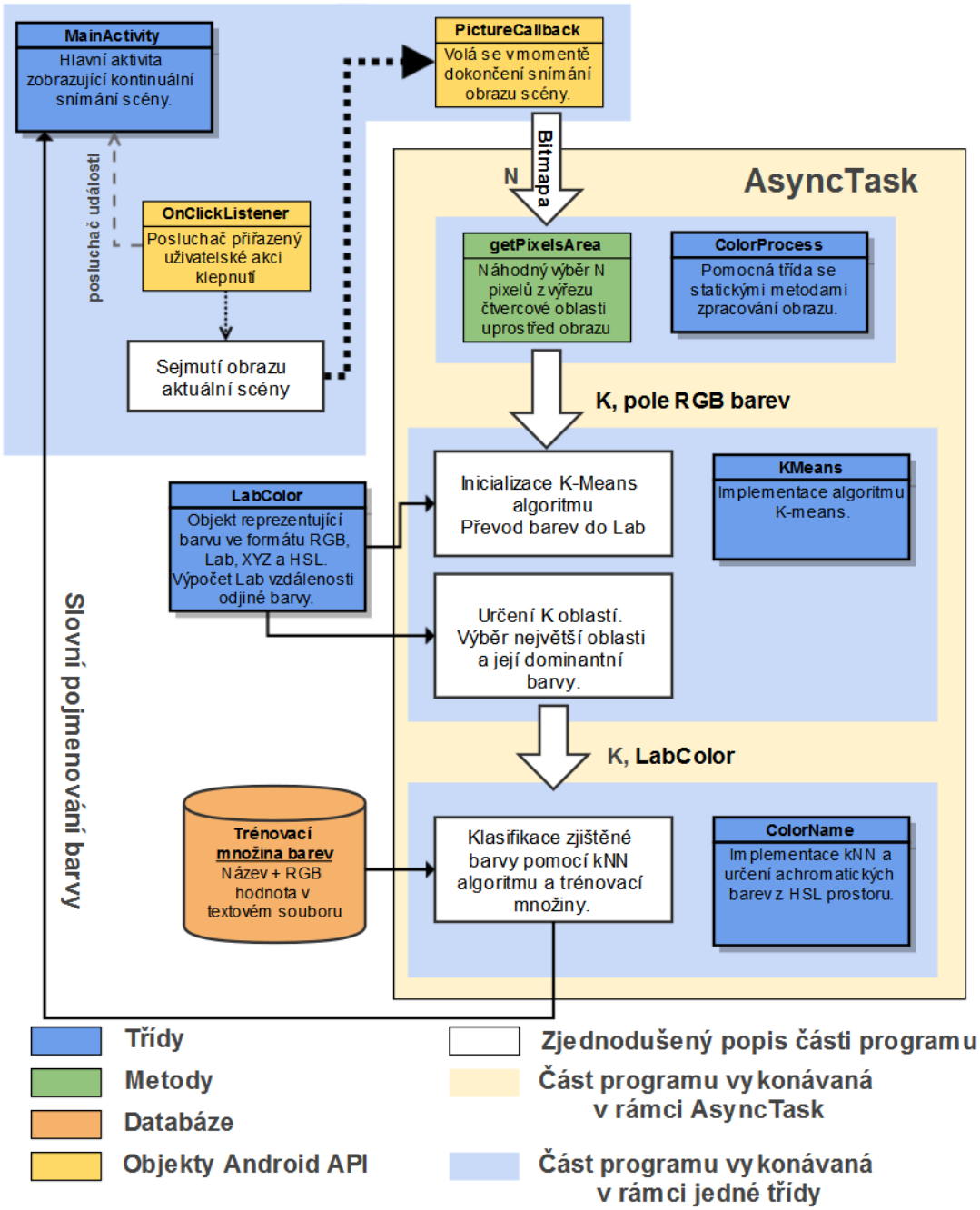
Pro počáteční nastavení kamery je nejdříve nutné vybrat rozlišení, ve kterém bude snímat scénu. Všechna přípustná rozlišení kamery jsou popsány ve třídě Camera.Parameters a výběr optimálního rozlišení se řídí dvěma kritérii:

1. Poměr šířky a výšky rozlišení kamery odpovídá poměru šířky a výšky displaye zařízení.
2. Plocha dána násobením šířky a výšky rozlišení kamery je maximální.

V závislosti na tom, zda zařízení obsahuje přisvětlovací diodu, se tato dioda rozsvítí a vyvážení bílé se nastaví na automatický režim.

Na zobrazený náhled snímané scény je připojen posluchač, který čeká na uživatelskou akci klepnutí na obrazovku. Po provedení této akce se zadá frameworku Camera příkaz k sejmutí obrazu z kamery. Tato akce je asynchronní a po dokončení snímání je zavolán PictureCallBack. V tomto momentě je k dispozici obraz sejmuté scény v okamžiku dotyku, který je asynchronně zpracován pomocnými třídami. Během zpracování přijatých dat je zamezeno pořizování dalších snímků, aby nedošlo k zahlcení paměti telefonu.

[1]) `https://play.google.com/store/apps/details?id=org.opencv.engine`



Obrázek 4.1. Grafické schéma aplikace rozpoznávání barev.

4.3.2 Třída LabColor

Třída LabColor je třídou, která v aplikaci reprezentuje barvu. Informace o barvě jsou přístupné v rozdílných barevných prostorech (RGB, Lab, HSL), které jsou používány v různých částech výpočtů. Součástí třídy jsou konverze mezi jednotlivými barevnými prostory, které byly detailně popsány v sekci 3.2.5, výpočet vzdálenosti v Lab barevném prostoru jedné barvy od druhé, který je používán v algoritmu K-Means je implementován jako euklidovská vzdálenost (kde jednotlivé složky Lab tvoří trojdimenzionální prostor). Další statickou metodou, která je používána v K-Means, je výpočet průměrné barvy z množiny barev.

4.3.3 Třída KMeans

Jak již název napovídá, tato třída je implementací algoritmu K-Means. Tato implementace používá náhodnou inicializaci, která se provede v případě, že zadaná množina barev pro rozdělení do jednotlivých shluků obsahuje více barev, než je hodnota parametru K. Pokud je hodnota K větší než počet barev, jsou jednotlivé barvy rozřazeny do tolika shluků, kolik je různých barev a jako dominantní barva je vybrána ta, která má nejvíce hodnot ve svém shluku. Pokud je při náhodné inicializaci přiřazena stejná barva jako výchozí barevný průměr dvou či více shluků, je inicializace opakována do té doby, než jsou všechny barvy rozdílné.

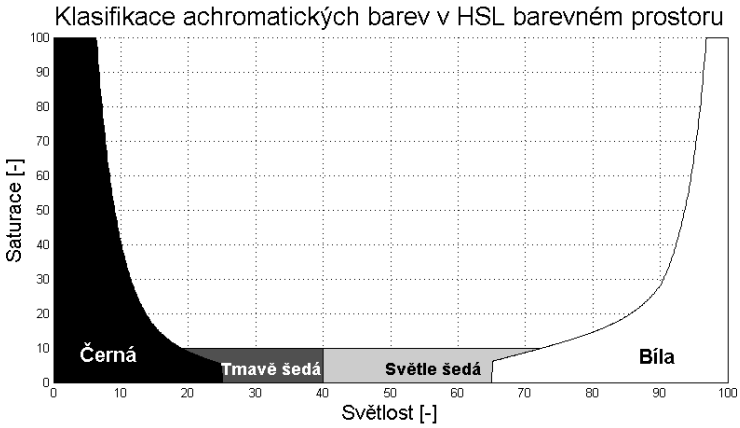
Po inicializaci je průchod algoritmem standardní, kdy se po každém určení nové průměrné barvy každého ze shluků všechny barvy přiřadí do nového shluku a je opakován do té doby než se obsah jednotlivých shluků neustálí na konečné množině přiřazených barev. Výsledná návratová hodnota je potom průměrná barva toho shluku, který obsahuje nejvíce prvků. Hodnota parametru K byla nastavena jako K=5.

4.3.4 Třída ColorProcess

Třída ColorProcess slouží jako pomocná třída obsahující statické metody. Jednou z těchto metod je getPixelsArea, která přijíma v argumentu Bitmapu, ze které je z prostřed vybrána čtvercová část o zadaných rozměrech. Tato oblast je posléze interpretována jako množina barev, kde jeden pixel reprezentuje jednu barvu. Z této množiny je náhodně vybrána podmnožina, ze které je následně algoritmem K-Means vybrána dominantní barva.

4.3.5 Třída ColorName

Ve třídě ColorName je implementován algoritmus kNN, který jako trénovací množinu používá textový soubor obsahující naučené barvy. Přiřazení názvu barvy podle kNN se provede pouze v případě, že barva v prostoru HSL nedosahuje určitých hodnot saturace a pohybuje se ve vymezeném prostoru světlosti. Tyto hodnoty byly určeny testováním aplikace na dvou přístrojích z různé výkonostní třídy a za jiných světelných podmínek. Klasifikace achromatických barev v HSL barevném prostoru vychází z předpokladu, že kamera snímá více a méně osvětlené objekty jinak a dochází ke zkreslení zejména achromatických barev, kdy černá obvykle bývá zachycena jako tmavě šedá a bílá jako světle šedá. Dalším problémem, která tato klasifikace potlačuje je nesprávné vyvážení bílé, kdy pro malé saturace jsou barvy klasifikovány jako achromatické.



Obrázek 4.2. Určení achromatických barev v HSL barevném prostoru.

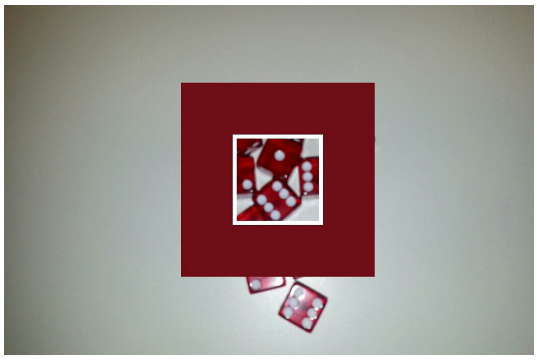
4.3.6 Trénovací množina dat

Trénovací množina pro určení vybrané barvy pomocí klasifikátoru kNN byla vytvořena z náhodně generovaných dat v barevném prostoru RGB, kterým byly manuálně přiřazeny jména z předem definovaného výčtu podporovaných barev. Tento seznam byl generován pomocí skriptu v programovacím jazyce Python a interpretu CodeSkulptor, což je volně přístupny webový interpret jazyka Python[1]).

Generovaná data tímto způsobem v sobě mohou nést chybu způsobenou špatným přiřazením jména k zobrazené barvě. To platí zejména u barev, které se vyskytují na hranici dvou různých barev. Dalším problémem je různé zobrazení barev na různých platformách. Barva může vyvolat jiný psychologický vjem na jiných zařízeních, např. různé počítačové monitory. Tyto chyby se snaží redukovat klasifikační algoritmus, který využívá více nejbližších barev trénovací množiny pro určení výsledné barvy. Větší množství dat obsažených v trénovací množině může také pomoci přesnější klasifikaci.

4.3.7 Uživatelské rozhraní

Uživatelské rozhraní bylo navrhnuto s přihlednutím k faktu, že aplikace může sloužit i lidem, kteří nemají praktickou nebo totální nevidomost, ale trpí slabozrakostí. Proto je ve tříde MainActivity periodicky vykreslován okolo středu, ze kterého se vyhodnocuje dominantní barva, čtvercový rámeček, který má barvu poslední uživatelem detekované barvy.



Obrázek 4.3. Návrh uživatelského rozhraní pro aplikaci rozpoznávání barev.

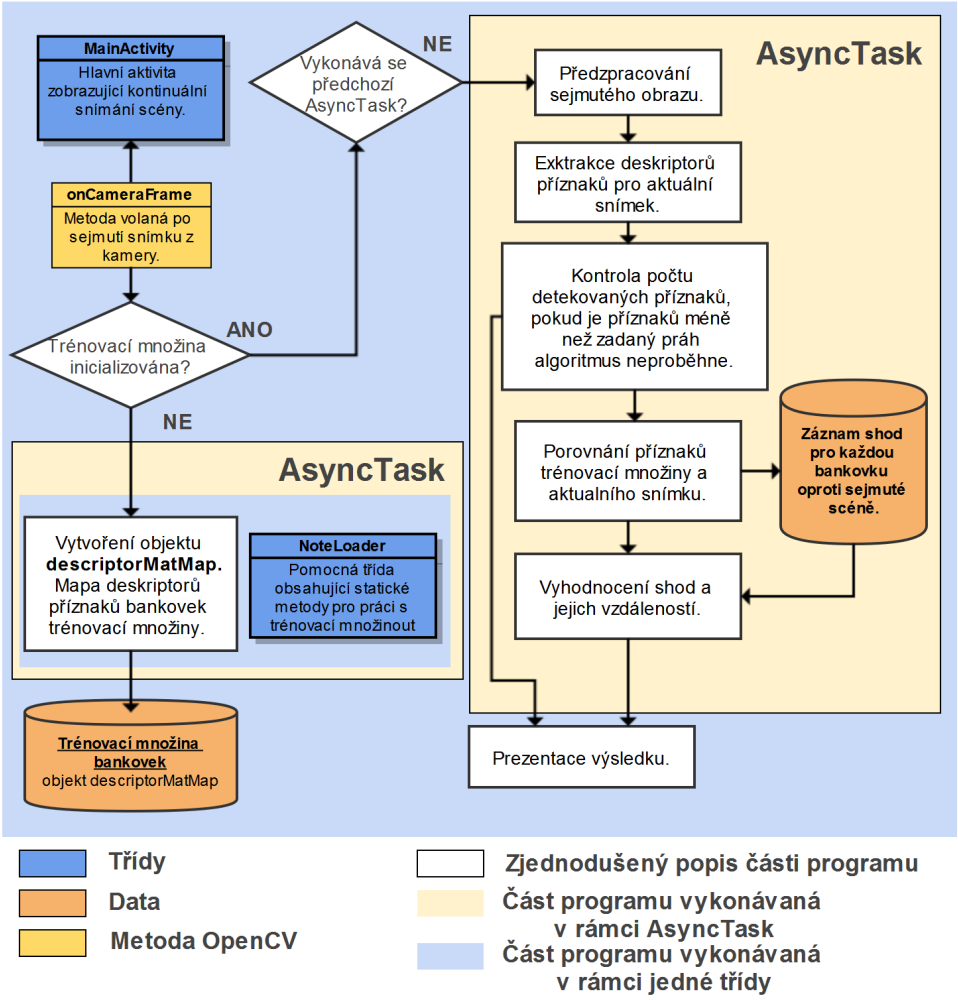
4.4 Rozpoznávání bankovek

Aplikace rozpoznávání bankovek popsaná v následujících několika podsekcích je implementace optimalizovaná pro zařízení Samsung Young S6310 [2]) disponující kamerou, která dokáže snímat video v maximálním rozlišení 640x480 pixelů. Toto zařízení bylo vybráno tak, aby reprezentovalo nízkou třídu telefonů, které jsou cenově dostupné a bez funkcí, které jsou nevidomým uživatelům nepřístupné, kvůli jejich handikepu. Samotná implementace se skládá ze tří celků:

- Příprava trénovací množiny, sběr dat, extrakce příznaků a jejich uložení.
- Předzpracování snímaných dat (obrazu), algoritmus rozpoznávání bankovek implementace aplikace.
- Nastavení apliakce pomocí metod, které hodnotí přesnost rozpoznání

[1]) `http://www.codeskulptor.org/`
[2]) `http://www.gsmarena.com/samsung_galaxy_young_s6310-5280.php`

V následujících několika podsekcích jsou jednotlivé části detailně popsány. Algoritmus pro nastavení prahu detekce je ilustrován na české koruně, jelikož pro ostatní měny byla přístupná pouze trénovací množina bankovek. Na obr. 4.4 je znázorněné schéma průběhu aplikace rozpoznávání bankovek.



Obrázek 4.4. Grafické schéma aplikace rozpoznávání bankovek.

4.4.1 Pořízení a úprava vzorů bankovek

Pro každou podporovanou měnu byl vybrán vzorek reprezentující všechny bankovky v dané měně, tj. jedna bankovka od každé nominální hodnoty. Vybrané bankovky byly takové, které jevily nejmenší známky opotřebování, minimální počet viditelných přehybů, co nejmenší znečištění a pomačkání. Pro každu bankovku byly pořízeny dva fotografické snímky (rub a líc) ve stejných světelných podmínkách se známým vyvážením bílé. Z pořízených snímku byly vytvořeny obdélníkové výřezy bankovek (pro odstranění pozadí snímku) v grafickém editoru Gimp. Takto upravené snímky byly uloženy v ruzných velikostech. V maximální velikosti, kdy se šířky bankovek v pixelech liší v závislosti na skutečné velikosti bankovek a poté ve velikostech s šířkou snímku 1200px, 1000px, 800px, 700px a 600px. Tyto různé rozměry poté byly testovány jako výchozí snímky pro extrakci příznaků na různých přístrojích.

Druhou sadou obrázků bankovek pro extrakci příznaků tvoří bankovky, které jsou digitálně upraveny tak, aby obsahovaly pouze data, která jsou společná pro všechny

bankovky dané nominální hodnoty v oběhu. Z oříznutých bankovek byly vyretušovány prvky jako číselné označení bankovky nebo rok vydání bankovky. Další úpravou bylo odstranění prvků, které se shodují pro všechny hodnoty bankovek v dané měně, například vlajka Evropské unie na Euro bankovkách. Takto upravené snímky bankovek byly opět uloženy ve stejných velikostech jako snímky, které byly pouze oříznuty bez dalších úprav.

Upravené snímky připravené k extrakci příznaků jsou v příslušné velikosti a provedení (s či bez přídavných retuší bankovek) uloženy jako bitmapa s názvem ve formátu, který jednoznačně identifikuje bankovku v rámci měny – **H–X–Y**, kde **H** označuje číselnou hodnotu bankovky, **X** označuje zda se jedná o snímek líce či rubu bankovky (A pro líc a B pro rub) a **Y** označuje verzi bankovky. Tento formát je později využit při sestavování souborů s deskriptory příznaků bankovek.

4.4.2 Uložení příznaků trénovací množiny v datovém úložišti

Z uložených obrázku bankovek každé měny se extrahují deskriptory příznaků, ke kterým se doplní metadata – hodnota bankovky, údaj o tom, zda se jedná o rub či líc a verze bankovky (měna může mít více typů bankovek jedné hodnoty). Data o jedné bankovce jsou uloženy ve třídě **BankNote**. Deskriptory příznaků (v OpenCV reprezentovány formátem Mat) jsou v této třídě uloženy jako textové řetězce JSON. Tento způsob byl zvolen, aby bylo možné uložit extrahované deskriptory jako serializovaný objekt do souboru. Pro uchování objektu v souboru musí objekt, který chceme uložit, implementovat Java rozhraní Serializable a všechny jeho vnitřní proměnné musí být primitivními typy nebo také implementovat toto rozhraní.

Konverze objektu Mat na textový řetězec JSON se provádí statickou metodou matToJson a zpětná konverze metodou matFromJson. V textovém řetězci jsou zaznamenány informace o počtu řádků a sloupců objektu Mat, o velikosti jednoho elementu objektu Mat a také samotná binární data obsažená v objektu Mat, která jsou zakódována Base64 kódováním. Pro samotnou konverzi objektu Json na textový řetězec je využito knihovny Gson [1]).

4.4.3 Předzpracování obrazu

Sejmutý obraz je před detekcí příznaků upraven tak, aby byly vylepšeny jeho vlastnosti a bylo získáno více či méně, v závislosti na kritériu, příznaků. Vychází z předpokladu, že při rozpoznávání bankovky bude ve scéně bankovka zabírat podstatnou část a pozadí bude zabírat minimální část. V takovém případě je aplikace např. algoritmu ekvalizace histogramu zaměřena přímo na bankovku a není ovlivňována pozadím, které není známé.

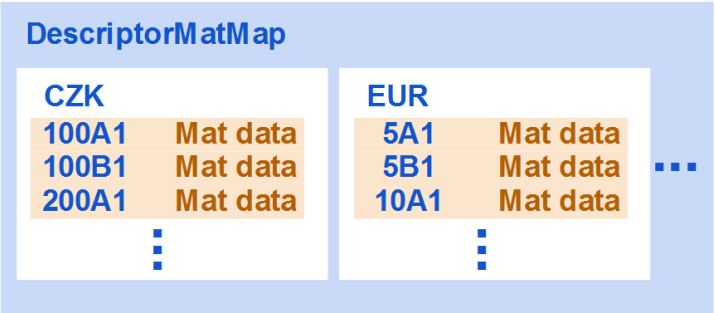
První provedenou úpravou je gaussovské rozmazání s malým konvolučním jádrem, které částečně potlačí šum v sejmutém obrazu. Tato úprava většinou sníží počet detekovaných klíčových bodů, což může pro použití detekčního algoritmu FAST být výhodné, pokud detekuje více příznaků v malém okolí obrazu. Po rozmazání obrazu je aplikován algoritmus ekvalizace histogramu, který v případě použití detektoru ORB umožní získání více příznaků. Obě metody předzpracování vycházejí ze zkoumání výstupní sekvence obrazů při snímání videa kamerou přístroje Samsung Young.

Výsledné předzpracování, použité na snímku scény pro rozpoznání bankovky je aplikováno také na trénovací množině před extrakcí příznaků.

[1]) `https://code.google.com/p/google-gson`

4.4.4 Reprezentace trénovací množiny v běhu programu

Při vyhodnocování snímané scény a porovnávání příznaku oproti trénovací množině je v programu využíván objekt descriptorMatMap. Tento objekt je Mapou, ve které je klíčem kód měny a obsahuje další mapu s klíčem reprezentující bankovku dle formátu hodnota, rub/líc, verze a objekt Mat s deskriptory příznaků. Naznačení struktury je na obr. 4.5.



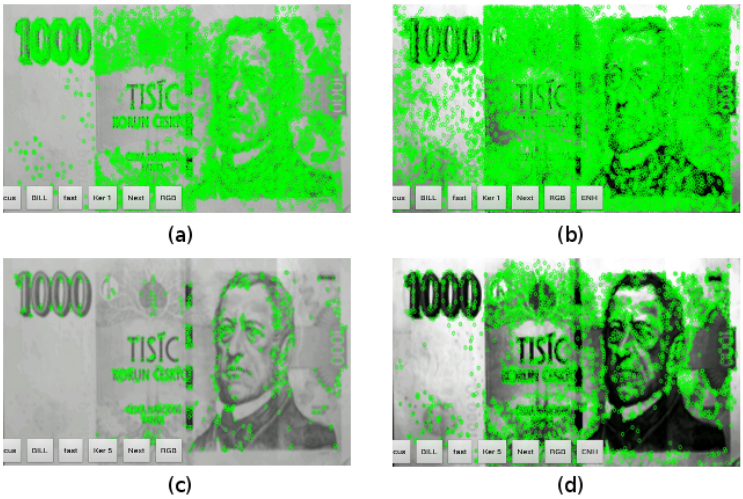
Obrázek 4.5. Příklad struktury objektu descriptorMatMap obsahující informace o trénovací množině.

4.4.5 Uživatelské rozhraní

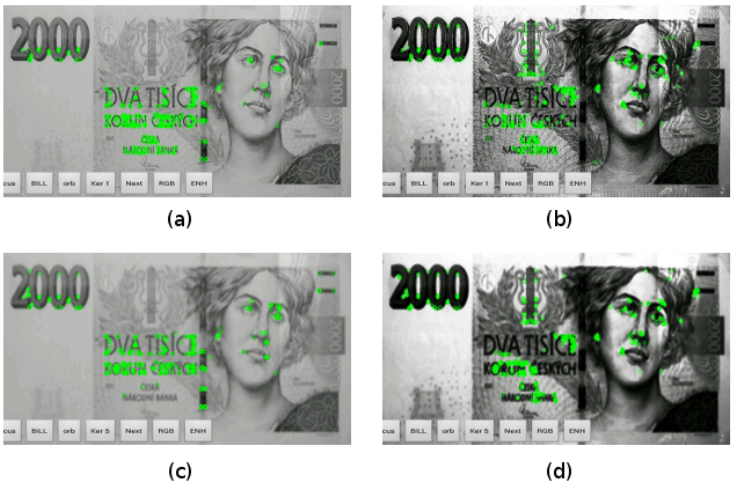
Uživatelské rozhraní se skládá z náhledu právě snímané scény a zvukové zpětné vazby. Po proběhnutí jednoho cyklu porovnání příznaků trénovací množiny jedné měny oproti sejmutému obrazu se ozve kliknutí, které nevidomému uživateli zprostředkuje informaci o tom, že detekce probíhá. Pokud je ve scéně rozpoznána bankovka, obraz zamrzne a uživateli se přehraje hlasový výstup oznamující hodnotu sejmuté bankovky. Pokud uživatel klepne kdekoli na displayi, detekce se opět spustí se zvukem kliknutí v každém cyklu provedeného AsyncTask.

4.4.6 Algoritmus rozpoznání bankovky

Algoritmus detekce v programu probíhá v objektu GetNote, který je instancí AsyncTask. Z předzpracovaného sejmutého obrazu scény se extrahují příznaky metodou ORB. Tyto příznaky jsou posléze porovnávány oproti příznakům získaným z trénovací množiny, které jsou v programu zpřístupněny v objektu descriptorMatMap. Z tohoto objektu je vybrána mapa bankovek pro nastavenou měnu a iterací přes všechny v ní uložené bankovky jsou párovány jejich deskriptory příznaků oproti těm, které byly sejmuty z aktuální scény. Všechny zaznamenané shody, které mají vzdálenost menší než definovaný práh th_1 se uloží do seznamu. Z tohoto seznamu se nakonec zjistí, pro kterou bankovku bylo nalezeno nejvíce shod s nejmenší celkovou vzdáleností mezi spárovanými příznaky. Pokud je těchto shod více než definovaný práh th_2 je příslušná bankovka určena jako výsledná. Postup nastavení prahů th_1 a th_2 je popsán v sekci 4.4.8



Obrázek 4.6. Příklady předzpracování bankovky s ekvalizací histogramu a rozmazáním a následnou detekcí příznaků FAST. (a) Bez rozmazání, bez ekvalizace histogramu. (b) Bez rozmazání, s ekvalizací histogramu. (c) Rozmazání s jádrem o rozměru 5x5, bez ekvalizace histogramu. (d) Rozmazání s jádrem o rozměru 5x5, s ekvalizací histogramu.



Obrázek 4.7. Příklady předzpracování bankovky s ekvalizací histogramu a rozmazáním a následnou detekcí příznaků ORB. (a) Bez rozmazání, bez ekvalizace histogramu. (b) Bez rozmazání, s ekvalizací histogramu. (c) Rozmazání s jádrem o rozměru 5x5, bez ekvalizace histogramu. (d) Rozmazání s jádrem o rozměru 5x5, s ekvalizací histogramu.

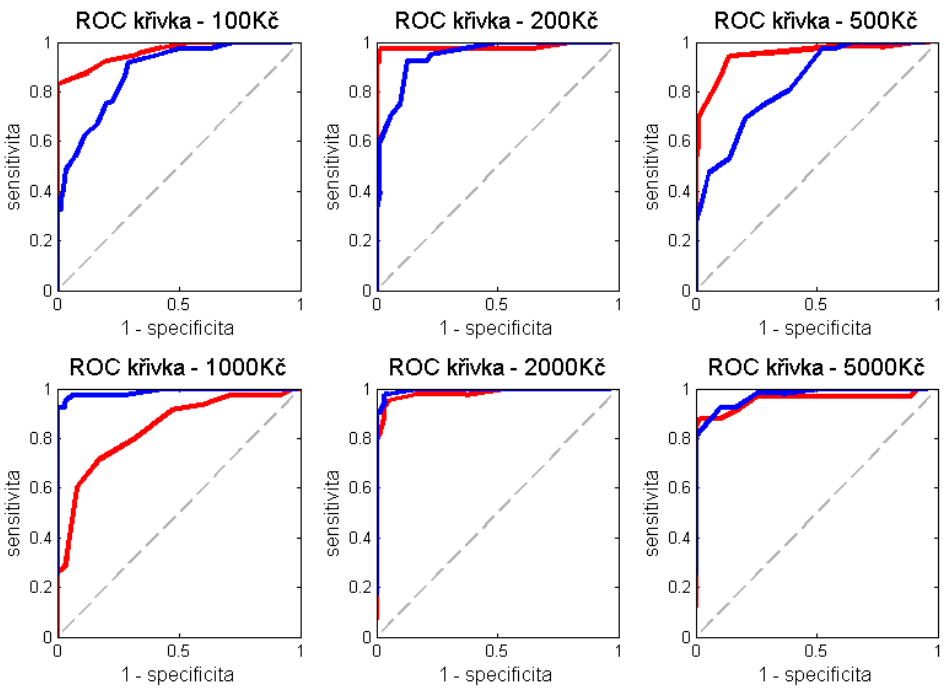
4.4.7 Použité třídy

Aplikace obsahuje třídu MainActivity, ve které je obsaženo UI a logika aplikace. MainActivity obsahuje dva objekty AsyncTask, jeden pro načtení extrahovaných příznaků trénovací množiny ze souborů a druhý, ve kterém se vykonává samotný algoritmus detekce. Současně může být spuštěn pouze jeden objekt AsyncTask. Třída NoteLoader obsahuje statické metody pro extrakci příznaků z trénovací množiny a konverze mezi objekty Mat a řetězci reprezentující JSON objekt. Dále aplikace obsahuje třídy, ve kterých jsou trénovací data pro potřeby uložení do souboru BankNote, která obsahuje informace o jedné bankovce a NoteGroup, která obsahuje všechny bankovky z jedné měny.

4.4.8 Volba prahů pro určení kvality shod

Pro volbu prahů th_1 a th_2 v algoritmu detekce byla vytvořena aplikace, která pro každý sejmutý obraz zaznamenává počet shod s každou bankovkou a jejich vzdálenosti. Práh th_1 byl určen fixně z pozorování jako $th_1 = 10$. To znamená, že musí být zaznamenáno alespoň 10 dobrých shod mezi scénou a bankovkou aby bylo rozpoznávání vyhodnoceno jako pozitivní. Kvalitu shody určuje práh th_2, který je porovnáván se vzdálenostmi jednotlivých shod. Pokud je vzdálenost shody menší než th_2, je shoda vyhodnocena jako dobrá. Práh th_2 byl nastaven pomocí ROC (receiver operating characteristic) křivky a byly zvoleny dva přístupy:

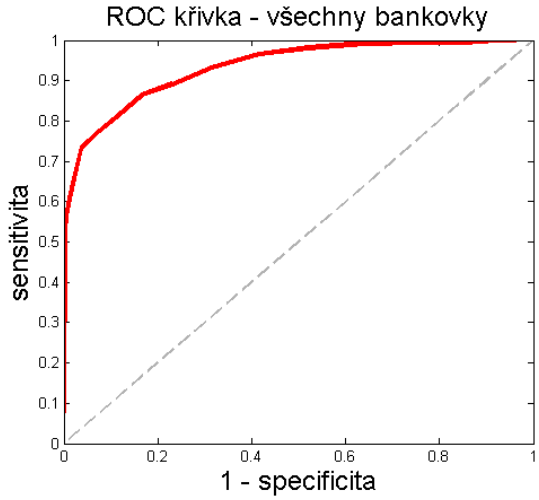
1. Pro každou bankovku je nastaven jiný práh v závislosti na její ROC křivce 4.8.
2. Práh je nastaven v rámci jedné měny pro všechny bankovky na stejnou hodnotu 4.9.



Obrázek 4.8. ROC křivky pro jednotlivé bankovky. Červená křivka reprezentuje líc bankovky, modrá křivka rub.

Obrazy, ze kterých se ROC křivky sestavovaly, tvořily přibližně z jedné třetiny pozitivní třídy, tedy bankovky hodnoty, která byla ve scéně rozpoznávána. Tyto bankovky byly snímány celé, různě natočené, případně s částečně zakrytým povrchem. Negativní třídy tvořily z jedné poloviny ostatní bankovky příslušné měny (célé, natočené či překryté) a z druhé poloviny scény neobsahující žádné bankovky. Pro každou bankovku (rub i líc) bylo snímáno přibližně 150 různých scén.

Hodnoty specificit a senzitivit jsou vyobrazeny pro práh th_2 v rozmezí $\langle 30, 60 \rangle$.



Obrázek 4.9. ROC křivka pro všechny bankovky.

Prahy odečtené ze zjištěných ROC křivek byly vybírány podle kritéria, které upřednostňuje více falešně negativních detekcí a minimalizuje počet falešně pozitivních. Toto kritérium je výhodné díky tomu, že snímaná scéna je vyhodnocována nepřetržitě, tedy pokud scéna obsahuje bankovku a aplikace nahlásí falešně negativní nález, je možné, že se bankovka rozpozná z následujících snímků. Jednotný práh pro všechny bankovky byl z ROC křivky nastaven na hodnotu $th_2 = 40$.

Získané prahy z vyhodnocovaných ROC křivek jsou svázané se zařízením, na kterém byly prováděny jednotlivé detekce.

Kapitola 5
Testování

Tato kapitola popisuje postup a výsledky testování implementovaných aplikací.

5.1 Rozpoznávání barev

Testování aplikace detekce barev bylo zaměřeno na přesnost určení předem vybraných barev z barevné palety. Použity byly vzorníky barev pro pokojové barvy a předměty denní potřeby, jako je oblečení, ovoce, zelenina, obaly potravin a jiné.

Test byl proveden za denního světla, s umělým osvětlením a s osvětlením přisvětlovací diodou. Použitým zařízením byl chytrý telefon Samsung Galaxy Note II (GT–N7100)[1]), který disponuje přisvětlovací diodou.

Druhým testem, kterému byla aplikace podrobena, je správná segmentace dominantní barvy ze snímané scény. V tomto testu byly snímány především předměty denní potřeby, jako jsou obaly produktů, oblečení s potiskem či tiskoviny. Snímaná scéna byla dobře nasvícena a byly v ní eliminovány odlesky, které by mohli vnášet do segmentace chybu.

5.1.1 Test přesnosti rozpoznání

V následujících třech tabulkách jsou uvedeny výsledky testování aplikace pro 3 různé světelné podmínky. V řádcích tabulek jsou uvedeny skutečné hodnoty snímané barvy a sloupce obsahují aplikací rozpoznané barvy. Pro každou barvu bylo provedeno 20 pokusů rozpoznání různých vzorků. Pro zobrazení tabulek v jednom řádku jsou názvy barev uvedeny zkratkou, kde **ČRV** je čevená, **ZLN** je zelená, **MDR** je modrá, **FLV**je fialová, **HND** je hnědá, **ŽLT** je žlutá, **RŽV** je růžová, **ORŽ** je oranžová, **ČRN** je černá, **BLÁ** je bílá a **ŠED** je šedá.

5.1.2 Osvětlení přisvětlovací diodou

	ČRV	ZLN	MDR	FLV	HND	ŽLT	RŽV	ORŽ	ČRN	BLÁ	ŠED	správně
ČRV	**16**	0	0	0	1	0	1	0	0	1	1	80%
ZLN	0	**19**	0	0	0	0	0	0	0	1	0	95%
MDR	0	0	**17**	1	0	0	0	0	0	1	1	85%
FLV	1	0	0	**15**	0	0	1	0	0	1	2	75%
HND	1	0	0	0	**16**	1	0	1	1	0	0	80%
ŽLT	0	1	0	0	0	**14**	0	2	0	2	0	70%
RŽV	3	0	0	3	1	0	**9**	0	0	2	2	45%
ORŽ	2	2	0	0	2	3	0	**8**	0	1	2	40%
ČRN	0	0	1	0	0	0	0	0	**14**	0	5	70%
BLÁ	0	0	1	0	0	0	0	0	0	**15**	4	75%
ŠED	0	0	2	0	0	0	0	0	0	3	**15**	75%

Tabulka 5.1. Výsledky testování aplikace při osvětlení přisvětlovací diodou.

[1]) `http://www.samsung.com/global/microsite/galaxynote/note2`

Při osvětlení scény přisvětlovací diodou některé objekty začaly vytvářet odlesky. Tyto odlesky, pokud byl objekt snímán z malé vzdálenosti, zabíraly v sejmutém obrazu velkou plochu. To se projevilo špatně určenými barvami, kdy většinou byla barvě přiřazena hodnota bílá či šedá. Další chybou, která se při tomto osvětlení vyskytovala bylo přesvícení scény, kdy se z tmavých objektů detekovaly světlejší barvy, a to především u objektů s lesklým povrchem. U matných objektů, zejména u oblečení, naopak docházelo k přesnému rozpoznání barvy.

5.1.3 Umělé osvětlení

V tomto testu bylo cílem napodobit světelné podmínky, se kterými běžně příjde uživatel do styku například uvnitř budovy či doma. Jako zdroj světla byla vybrána úsporná žárovka s chromatickou teplotou přibližně 2700K.

	ČRV	ZLN	MDR	FLV	HND	ŽLT	RŽV	ORŽ	ČRN	BLÁ	ŠED	správně
ČRV	**18**	0	0	0	1	0	0	0	0	0	1	90%
ZLN	0	**18**	0	0	0	0	0	0	1	0	1	90%
MDR	0	0	**18**	0	0	0	0	0	1	0	1	90%
FLV	1	0	0	**18**	0	0	0	0	0	0	1	90%
HND	0	0	0	0	**16**	0	0	1	3	0	0	80%
ŽLT	0	2	0	0	4	**10**	0	3	0	0	1	50%
RŽV	2	0	0	1	2	0	**8**	1	0	2	4	40%
ORŽ	3	0	0	0	3	1	0	**10**	0	0	3	50%
ČRN	0	0	0	0	1	0	0	0	**18**	0	1	90%
BLÁ	0	0	0	0	2	0	0	0	0	**16**	2	80%
ŠED	0	0	0	0	1	0	0	0	2	0	**17**	85%

Tabulka 5.2. Výsledky testování aplikace při osvětlení žárovkou.

Intenzita světla produkovaná žárovkou byla mnohem menší než při použití LED přisvětlení. Proto se jevili snímané objekty tmavší než byly ve skutečnosti. To se projevilo zvýšeným počtem detekce šedé barvy a pro tmavé odstíny modré, hnědé a zelené zvýšenou detekcí černé. Problémem bylo také to, že světelný zdroj né vždy osvětloval objekt přímo, např. pokud uživatel stál ke zdroji světla zády, snímaný objekt byl zastíněn jeho tělem.

5.1.4 Denní světlo

V tomto testu byla aplikace testována venku při denním světle. Objekty, ze kterých se snímala barva nebyly vystavěny přímému slunečnímu svitu, tak aby se potlačily odlesky a přesvícení, jak tomu bylo při testování s přisvětlovací diodou.

V denním světle nebylo rozpoznávání barev zatíženo žádnou viditelnou chybou. Telefón, na kterém byla aplikace spuštěna správně nastavil vyvážení bílé a tedy i barvy zpracovávané aplikací byly věrné své předloze. Bílá barva byla několikrát klasifikována jako šedá, což je v tomto případě způsobeno nastaveným modelem rozpoznávání v HSL barevném prostoru.

5.1.5 Vyhodnocení testu přesnosti rozpoznávání

Ve všech světelných podmínkách byly barvy růžová a oranžová, případně žlutá klasifikovány s mnohem menší přesností než ostatní barvy viz tabulky 5.1, 5.2 a 5.3. Tato chyba byla pravděpodobně způsobena menším zastoupením těchto barev v trénovací množině,

	ČRV	ZLN	MDR	FLV	HND	ŽLT	RŽV	ORŽ	ČRN	BLÁ	ŠED	správně
ČRV	**18**	0	0	0	1	0	0	1	0	0	0	90%
ZLN	0	**20**	0	0	0	0	0	0	0	0	0	100%
MDR	0	0	**19**	0	0	0	0	0	0	0	1	95%
FLV	0	0	1	**18**	0	0	0	0	0	1	0	90%
HND	1	0	0	0	**17**	0	0	0	1	0	1	85%
ŽLT	0	3	0	0	2	**14**	0	0	0	1	0	70%
RŽV	3	0	0	2	0	0	**9**	1	0	3	2	45%
ORŽ	3	2	0	0	4	0	0	**9**	0	0	2	45%
ČRN	0	0	0	0	0	0	0	0	**18**	0	2	90%
BLÁ	0	0	0	0	0	0	0	0	0	**17**	3	85%
ŠED	0	0	1	0	0	0	0	0	0	1	**18**	90%

Tabulka 5.3. Výsledky testování aplikace za denního světla.

případně nepokrytím celé jejich škály v trénovací množině. Ostatní barvy byly rozpoznány s relativně vysokou úspěšností. Pokud se přihlédne k tomu, že uživatel bude test opakovat je pravděpodobné, že mu bude prezentována správná barva.

5.1.6 Test segmentace dominantní barvy

Pro testování přesnosti určení dominantní barvy ve scéně bylo vybráno 30 objektů denní potřeby, které nesou jednu výraznou barvu a několik dalších minoritních barev. Objekt byl vycentrován do středu scény a byla zjistěna jeho dominantní barva. Z 30 provedených testů byla tato barva správně určena v 29 případech. V případě, kdy barva byla určena nesprávně, byl pokus opakován a v druhém měření již byla dominantní barva zjistěna správně. Nesprávné určení bylo zapříčiněno nevhodným náhodným výběrem množiny barev, na kterých se provádí algoritmus K-Means. Určení barev z některých objektů je vyobrazeno na obr. 5.1.

Obrázek 5.1. Příklad určení dominantní barvy ve scéně.

5.2 Rozpoznávání bankovek

Testování přesnosti rozpoznání vychází z podobných podmínek jako určování prahu pomocí ROC křivky. Pro každou bankovku je provedeno 20 testů, pro každou stranu bankovky, ve kterých se bankovka snímá v různých podmínkách. Tyto podmínky zahrnují natočení bankovky vůči kameře v různých směrech, přehnutí či zakrytí části bankovky a pomačkání bankovky. Pokud není bankovka rozpoznána v 5 porovnání příznaků, je výsledek považován za negativní. Testovací množina dat obsahuje jiné bankovky než ty, ze kterých byla vytvořena trénovací množina dat, proto byl také test prováděn pouze na korunách českých, od kterých byl k dispozici velký počet bankovek.

Test byl proveden na zařízení Samsung Young ve dvou světelných podmínkách:

- Osvětlení velké intenzity (ekvivalent denního světla či umělého přisvětlení z malé vzdálenosti).
- Osvětlení malé intenzity (ekvivalent šera či umělého osvětlení malé intenzity).

5.2.1 Osvětlení velké intenzity

Tento test byl proveden za denního světla, kdy bylo zaručeno, že bankovka je osvícená světlem stejné intenzity rovnoměrně v celé své ploše zabírané kamerou.

	100A	100B	200A	200B	500A	500B	1000A	1000B	2000A	2000B	5000A	5000B	NEG
100A	**17**	0	0	0	0	0	0	0	0	0	0	0	3
100B	0	**13**	0	0	0	0	0	0	0	0	0	0	7
200A	0	0	**17**	0	0	0	0	0	0	0	0	0	3
200B	0	0	0	**18**	0	0	0	0	0	0	0	0	2
500A	1	0	0	0	**10**	0	0	0	0	0	0	0	9
500B	0	0	0	0	0	**18**	0	0	0	0	0	0	2
1000A	0	0	0	0	0	0	**16**	0	0	0	0	0	4
1000B	0	0	0	0	0	0	0	**17**	0	0	0	0	3
2000A	0	0	0	0	0	0	0	0	**17**	0	0	0	3
2000B	0	0	0	0	0	0	0	0	0	**18**	0	0	2
5000A	0	0	0	0	0	0	0	0	0	0	**19**	0	1
5000B	0	0	0	0	0	0	0	0	0	0	0	**19**	1
NEG	0	0	0	0	0	0	2	0	0	0	0	0	**98**

Tabulka 5.4. Výsledky testování aplikace za osvětlení velké intenzity.

Rozpoznání bankovek za intenzivního osvětlení dávalo dobré výsledky v případě, že byla zabírána celá bankovka. V případě zakrytí části bankovky záleželo na části bankovky která byla zakrytá. Pokud byla zakrytá část obsahující velké množství příznaku, rozpoznání trvalo déle či nebylo úspěšné. Špatné výsledky byly dosaženy při pomačkání bankovky, v takovém případě bylo velmi obtížně bankovku rozpoznat, a to i v případě, že byla obsažená ve scéně svou celou plochou.

Pouze jednou byla bankovka klasifikována jako jiná, což je dáno nastavením menšího prahu th_2, který zajišťuje menší počet falsešně pozitivních určení bankovek. Při snímání scén neobsahující žádné klasifikované bankovky (tyto scény byly tvořeny i bankovkami jiné měny), byla dvakrát nesprávně klasifikována bankovka s hodnotou 1000Kč. To bylo zapříčiněno tím, že se ze vzoru této bankovky extrahovalo více příznaků, než z ostatních.

5.2.2 Osvětlení malé intenzity

Tento test byl proveden v místnosti s umělým osvětlením, kdy snímané bankovky byly zastíněny tak, že na ně nedopadalo světlo přímo ze zdroje.

V testování při malé intenzitě světla trvalo rozpoznání bankovek delší dobu, než při intenzivním osvětlení. Také nastalo více případů, kdy nebyly bankovky rozpoznány vůbec. To je dáno menší hloubkou detailu ve snímaných bankovkách při méně intenzivním osvětlení a tudíž menším počtem příznaků získaných ze snímané scény, která taktéž obsahovala více šumu. Pokud byly bankovky rozpoznány, byly vždy zařazeny do správné třídy. V případě testování scén neobsahující bankovky rozpoznávané měny bylo vždy správně vyhodnoceno, že obraz žádnou bankovku neobsahuje.

	100A	100B	200A	200B	500A	500B	1000A	1000B	2000A	2000B	5000A	5000B	NEG
100A	**15**	0	0	0	0	0	0	0	0	0	0	0	5
100B	0	**12**	0	0	0	0	0	0	0	0	0	0	8
200A	0	0	**14**	0	0	0	0	0	0	0	0	0	6
200B	0	0	0	**18**	0	0	0	0	0	0	0	0	2
500A	0	0	0	0	**11**	0	0	0	0	0	0	0	9
500B	0	0	0	0	0	**16**	0	0	0	0	0	0	4
1000A	0	0	0	0	0	0	**16**	0	0	0	0	0	4
1000B	0	0	0	0	0	0	0	**16**	0	0	0	0	4
2000A	0	0	0	0	0	0	0	0	**15**	0	0	0	5
2000B	0	0	0	0	0	0	0	0	0	**17**	0	0	3
5000A	0	0	0	0	0	0	0	0	0	0	**18**	0	2
5000B	0	0	0	0	0	0	0	0	0	0	0	**17**	3
NEG	0	0	0	0	0	0	0	0	0	0	0	0	**100**

Tabulka 5.5. Výsledky testování aplikace za osvětlení malé intenzity.

Kapitola 6
Diskuze

Tato kapitola obsahuje zhodnocení implementovaných aplikací a možnost dalšího vývoje v budoucnu.

6.1 Rozpoznávání barev

Implementace rozpoznávání barev pro chytré telefony je v samotném zadání omezena kvalitou kamery, která se liší u každého zařízení. Pokud by byl brán v potaz chytrý telefon s nejlepší, ve smyslu věrnosti reprezentovaných barev, kamerou dostupnou na současném trhu, stále nelze výsledné detekované barvy s jistotou považovat za optimálně rozpoznané. Toto je zapříčiněno zejména nekotrolovatelnými světelnými podmínkami, ve kterých bude aplikace používána. Dalšími chybami vnesenými do rozpoznání je samotné přesné zaměření kamery na objekt, které je stíženo faktem, že aplikace je primárně používaná nevidomými uživateli. Řešením by mohl být samostatný modul, dodávaný jako rozšíření, který by byl připojen k telefonu například pomocí technologie Bluetooth, nebo USB kabelem. To ovšem stojí proti filozofii samotné aplikace, která se snaží prostředkovat funkci rozpoznávání barev uživateli, který má k dispozici pouze chytrý dotykový telefon.

6.1.1 Project Ara

Project Ara [1]) (v době psaní této práce ve fázi prototypu Spiral 2) je modulární chytrý telefon vyvýjený týmem Google ATAP (Google Advanced Technology and Projects). Koncept tohoto telefonu stojí na absolutní hardwarové modularitě chytrého telefonu. Základem je kovový rámeček (endoskeleton) obstarávající funkci sběrnice, který bude jako jediný produkován společností Google. Všechny ostatní prvky (moduly) budou volitelnými, v případě displaye, baterie a procesoru nezbytnými, rozšířeními a v podstatě tak budou nabízet absolutní přizpůsobitelnost chytrého telefonu na míru uživateli. Toho by mohlo být využito ve spojení chytrých telefonů a asistivních technologií. Pro nevidomé uživatele by byl distribuován základní model s nezbytnímy moduly (baterie procesor, konektivita) s možností doplnění o pokročilé moduly jako například modul pro rozpoznávání barev.

6.1.2 Modul pro rozpoznávání barev

Navrhovaný modul by sestával z senzoru barvy tvořeným alespoň třemi LED, které emitují světlo na různých vlnových délkách, např. červená, modrá, zelená a fotoprvku. Každá z LED by osvítila snímaný objekt na krátky časový okamžik, dostatečně dlouhý na to, aby fotocitlivý prvek mohl sejmou intenzitu odraženého světla. Z naměřených hodnot by posléze byla vypočítána barva zkoumaného objektu. Tento přístup by odstranil problém se zaměřením, jelikož by se senzor přikládal přímo na zkoumaný objekt.

[1]) `http://www.projectara.com/`

Také by byl konstrukcí modulu zamezen přístup ambientního světla v okamžiku měření a tím by se eliminoval problém s různými světelnými podmínkami.

Obrázek 6.1. Project Ara rámeček s moduly.

6.1.3 Vylepšení implementované aplikace

V samotné aplikaci je prostor pro vylepšení v implementaci algoritmu KMeans, který pracuje se vzdálenostmi v barevném prostoru Lab. Inicializace K prvních průměrů barev, v současné implementaci náhodná, by mohla být optimalizována na základě jiného kritéria. Časový interval vyhodnocení dominantní barvy by mohl být zkrácen, například ekeftivnějšími výpočty průměrné barvy, nebo rychlejšími výpočtmi vzdáleností.

Přesnost rozpoznání barvy by mohla být vylepšena pro achromatické barvy lepším modelem určenví v HSL barevném prostoru, který by byl vytvořen odděleně pro různé typy použitých kamer. Větší trénovací množina a s ní použítí většího parametru K v kNN by mohla také zpřesnit výsledek rozpoznávání.

6.2 Rozpoznávání bankovek

Pro aplikaci rozpoznávání bankovek jsou nejdůležitějšími aspekty dostatečné osvětlení bankovky a snímání scény kvalitní kamerou. První z uvedených je snadno splněn v případě, že telefon disponuje přisvětlovací diodou. Druhý aspekt je naplněn obtížněji, vzhledem k tomu, že primární určení aplikace je pro nevidomé uživatele a tito uživatelé si ve většině případů pořizují telefony z levnější, nižší třídy. Takové telefony mívají kamery s malým rozlišením a špatnou optikou. Jedním z telefonů, který je distribuovaný na českém trhu pro nevidomé uživatele je telefon Samsung Young S6310, který disponuje kamerou, která dokáže snímat video v maximálním rozlišení 640x480 pixelů. Tento telefon patří mezi nejlevnější chytré telefony distribuované na českém trhu.

Následující popsané řešení je návrhem pro telefony disponující velmi kvalitními kamerami. Premisa, ze které vychází, je taková, že trh s mobilními technologiemi se stále vyvíjí a s ním i parametry nabízených telefonů. A tedy v nějakém časovém horizontu budou na trhu nabízeny chytré telefony v přijatelné cenové kategorii srovnatelné s těmi, které v současné době mají parametry považované za velmi dobré. Také lze uvažovat o možnosti popsané v sekci o rozpoznávání barev, že uživatel bude disponovat telefonem vyvíjeným jako Project Ara a kvalitní kameru si pořídí jako modul.

6.2.1 Využití EURion konstelace

Navrhnuté řešení, pro zařízení, která disponují automatickým zaostřováním a kamerou ve vysoké kvalitě, staví na algoritmu detekce příznaků FAST. Ze snímané scény se využije pouze jeden barevný kanál, v závislosti na barvě EURion konstelace příslušné bankovky. V tomto kanále se jeví mezikruží EURion konstelace vysoce kontrastní oproti svému pozadí, což zaručuje možnost práce s vysokým prahem algoritmu FAST a zkoumání jen případů, kdy jsou body na okolní kružnici tmavší než centrální bod. Tím je potlačena detekce příznaků, které nejsou součástí EURion konstelace. Upravený algoritmus FAST pracuje ve své zrychlené variantě, kdy nejprve kontroluje body na čtyřech pozicích, které jsou od sebe na zkoumané kružnici posunuty o 90 stupňů. Pokud jsou tyto body vyhodnoceny jako pozitivní nález (jejich jas je menší než rozdíl jasu centrálního bodu a zadaného prahu), je test proveden i na zbylých bodech kružnice. Poté je vyhodnoceno zda se jedná o příznak, či nikoliv. Potvrzení, že nalezený bod je skutečně bodem EURion konstelace je, že pokud se zvyšuje poloměr zkoumané kružnice, nastane okamžik, kdy jas všech bodů vzroste o hodnotu prahu oproti jasu nalezeného mezikruží.

Výhodou této metody je, že není nutné zkoumat orientaci příznaků, jelikož EURion konstelace je složena z kružnic. Po nalezení příznaků ve zkoumané bankovce lze postupovat stejně jako v případě implementované metody rozpoznání bankovek popsané v této práci. Nevýhodou je, že tuto metodu ochrany před paděláním nepoužívají všechny bankovky, je tedy stále nutné použít jiné metody identifikace na bankovky, které EURion konstelaci neobsahují.

6.2.2 Vylepšení implementované aplikace

V implementované aplikaci je možné vylepšit nastavení parametrů pro detekci jednotlivých příznaků. Využitá knihovna OpenCV je původně vyvíjena pro programovací jazyk C++ a pro platformu Android jsou přepsané jednotlivé části této C++ knihovny. Android verze však neobsahuje plnou funkcionalitu a u některých tříd či metod není možné nastavit všechny parametry. Toto lze obejít využitím Android NDK (Native Development Kit), což je soubor nástrojů, který dovoluje implmentovat části Android aplikace v progamovacím jazyku C++, v tomto případě by byl Android NDK využit pro implementaci OpenCV knihovny v C++.

Heuristika stavějící na různých barvách bankovek může zmenšit počet nesprávně pozitivně určených bankovek. V takovém případě by byl z části obrazu, ve kterém se nachází bankovka určen histogram ve všech barevných kanálech, který by byl následně vyhodnocen, jestli odpovídá vzorové bankovce. Pokud by byla bankovka určena porovnáním příznaků, ale kontrolou histogramu by bylo zjistěno že se o identifikovanou bankovku pravděpodobně nejedná, bude vyhodnocena jako negativní (ve většině případů pravdivě negativní).

6.2.3 Doporučení pro užívání aplikace

Vzhledem k tomu, že tato aplikace slouží pro reálné rozpoznávání bankovek, každá nesprávně pozitivní identifikace bankovky může v krajních případech vést k finanční újmě uživatele. Z tohoto důvodu je doporučeno provést rozpoznávání bankovky několikrát, a to alespoň z rubu a líce bankovky. Pokud vrátí stejný výsledek, dá se ve většině případů bankovka považovat za správně rozpoznanou.

Ačkoliv není pro rozpoznání potřeba snímání celé bankovky, správné natočení bankovky tak, aby byla celá obsažená ve snímané scéně a její narovnání dopomáhají k lepšímu a rychlejšímu rozpoznání.

Kapitola 7
Závěr

Cílem této práce bylo usnadnit těžce zrakově postiženým osobám každodenní úkony, například oblékání, rozlišení ruzných produktů či manipulaci s bankovkami. Tyto problémy byly adresovány skrze návrh a implementaci dvou mobilních aplikací pro chytré dotykové telefony.

První z implementovaných aplikací byla aplikace pro rozpoznávání barev. Před implementací bylo vycházeno z premisy, že čím bude osvětlení intenzivnější, tím bude rozpoznávání kvalitnější. Toto se však ukázalo být nepravdivým předpokladem, jelikož z testování aplikace vyplývá, že za použití přisvětlovací diody může dojít k vážnému zkreslení při vyhodnocování barvy. Toto zkreslení je způsobeno výraznými odlesky na lesklých předmětech, které jsou ovlivněny úhlem pod kterým jsou objekty snímány. Naopak předpoklad, že osvětlení nízké intenzity rozpoznávání ztíží, byl potvrzen. Z hlediska segmentace obrazu, a vybrání nejvíce v něm zastoupené barvy, aplikace zcela splnila požadavky.

Druhá z implementovaných aplikací, rozpoznávání bankovek, byla navrhnuta tak, aby zrakově postiženým usnadnila manipulaci s bankovkami. To zahrnuje rychlé určení hodnoty bankovky, např. v obchodě, kdy je v současnosti využíváno speciální šablony. Cílem apliakce bylo rozpoznat bankovky co nejspolehlivěji, což může u některých bankovek způsobit delší dobu rozpoznávání, aby se předešlo případným finančním újmám. Kvalita rozpoznání také přináší psychologický efekt, který, v případě vždy přesné klasifikace bankovky, podporuje důvěru uživatele v aplikaci. Tato nabraná důvěra může přispět k tomu, že si uživatelé zvyknout používat mobilní aplikace, které jim pomáhají zprostředkovat informace o svém okolí. Do budoucna lze tedy připravit cestu k dalším aplikacím, které budou zrakově postižení uživatelé používat.

Osobně doufám, že se na trhu objeví zařízení, které budou mít hardware mobilního telefonu přizpůsobený potřebám zrakově postižených uživatelů. V tomto ohledu je zajímavým projektem Google Project Ara, který by mohl nabídnout alternativní platformu pro uživatele se zrakovým postižením. Toto zařízení by mohlo využívat specializovaného hardwaru, jako např. navrhnutý modul pro rozpoznávání barev. Dalšími periferiemi by mohly být vstupy či výstupy implementující braillovo písmo, či jiná hardwarová tlačítka.

Literatura

[1] WORLD HEALTH ORGANIZATION. *Visual impairment and blindness* [online]. 2014 [cit. 2014-11-25]. Dostupné z:
`http://www.who.int/mediacentre/factsheets/fs282/en/`

[2] ČESKÁ NÁRODNÍ BANKA. *České bankovky* [online]. 2014 [cit. 2014-11-25]. Dostupné z:
`https://www.cnb.cz/cs/platidla/bankovky/`

[3] THE BUREAU OF ENGRAVING AND PRINTING. *Meaningful access to U.S. currency for blind and visually impaired individuals*. 2013. Dostupné z:
`http://www.moneyfactory.gov/images/Meaningful_Access_White_Paper_06-25-13.pdf`

[4] AMERICAN FOUNDATION FOR THE BLIND. *Keeping Track of Money* [online]. 2014 [cit. 2014-11-25]. Dostupné z:
`http://www.afb.org/info/living-with-vision-loss/money/25`

[5] HASANUZZMAN F. M., Yang X. a Tian Y., *Robust and Effective Component-Based Banknote REcognition for the Blind*, IEEE rtansactions on systems, man and czbernetics – part C, vol. 42, no. 6, pp 1021–1030 listopad 2012.

[6] DIGMAN M. a ELDER C., *Mobile Banknote Recognition and Conversion*, Stanford University, 2013.

[7] SCHANDA, Edited by János. *Colorimetry Understanding the CIE System*. Hoboken: John Wiley, 2007. ISBN 04-701-7562-1.

[8] ROSTEN, Edward a Tom DRUMMOND. *Machine learning for high-speed corner detection*. Department of Engineering, Cambridge University, UK, 2006.

[9] RUBLEE, Ethan, Vincent RABAUD, Kurt KONOLIGE a Gary BRADSKI. *ORB: an efficient alternative to SIFT or SURF*. Willow Garage, Menlo Park, California, 2011.

[10] LOWE D., *Distinctive image features from scale–invariant keypoints*, Int. J. Comput. Vis., vol. 60, pp. 91–110, 2004.

[11] BAY H., TUYTELLARS T. a GOOL L., *SURF: Speeded up robust features*, in Proc. 9th Eur. Conf. Comput. Vis., 2006, pp. 404—417.

[12] KUHN, M. G. *The EURion constellation*, 2002.

Příloha A
Použité zkratky

API	Application programming interface.
BRIEF	Binary Robust Independent Elementary Features.
CIE	International Commission On Illumination.
CZK	Koruna česká – měnová jednotka České reubliky.
EUR	Euro – měnová jednotka státu patřící do Evropské měnové unie.
FAST	Features from Accelerated Segment Test.
JSON	JavaScript Object Notation.
kNN	K nearest neighbours.
LED	Light emiting diode – Dioda emitující světlo.
OpenCV	Open Source Computer Vision.
ORB	Oriented FAST and Rotated BRIEF.
OS	Operační systém.
ROC	Receiver operating characteristic.
SDK	Software Development Kit.
SONS	Sjednocená organizace nevidomých a slabozrakých.
UI	User interface.
USA	Spojené státy americké.
USD	Americký dolar – měnová jednotka Spojených států amerických.

Příloha B
Obsah přiloženého CD

CD obsahuje zdrojové soubory obou aplikací, data pro určení prahu rozpoznávání bankovek a tuto práci ve formátu PDF. Přiložené CD neobsahuje žádne fotografie bankovek.

- **BNRec** – zdrojové kódy a aplikace rozpoznávání bankovek.
- **ColorFetch** – zdrojové kódy a aplikace rozpoznávání barev.
- **OpenCV-2.4.8-android-sdk** – knihovna OpenCV, nutná pro kompilaci aplikace rozpoznávání bankovek.
- **ROC** – data, ze kterých byly sestaveny ROC křivky pro určení prahů aplikace rozpoznávání bankovek.
- **pechaja6_dp.pdf** – PDF verze této práce.