Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Program na přehrávání notového zápisu

2019

Petr Němeček

Vedoucí práce: doc. RNDr. Miroslav Kolařík, Ph.D.

Studijní obor: Aplikovaná informatika, kombinovaná forma

Bibliografické údaje

Autor:	Petr Němeček
Název práce:	Program na přehrávání notového zápisu
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2019
Studijní obor:	Aplikovaná informatika, kombinovaná forma
Vedoucí práce:	doc. RNDr. Miroslav Kolařík, Ph.D.
Počet stran:	30
Přílohy:	1 DVD
Jazyk práce:	český

Bibliograhic info

Author:	Petr Němeček
Title:	Program for playing sheet music
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2019
Study field:	Applied Computer Science, combined form
Supervisor:	doc. RNDr. Miroslav Kolařík, Ph.D.
Page count:	30
Supplements:	1 DVD
Thesis language:	Czech

Anotace

Aplikace „Přehrávač not“ analyzuje a přehrává digitální fotokopie notového zápisu. K pořízení digitální fotokopie používá kameru zařízení, na kterém je spuštěna. Je sestavena pro operační systém Android. Byla napsána v programovacím jazyce Java s využitím knihovny OpenCV. Metody knihovny OpenCV byly použity k detekci objektů a struktur v obraze a jejich klasifikaci. Aplikace přehrává jednoduché notové zápisy se stupnicí podle houslového klíče v rozsahu tónů h až h”.

Synopsis

The „Sheet Music Player“ application analyzes and plays digital photocopies of musical notation. To make a digital photocopy, the camera uses the device on which it is running. It is built for the Android operating system. It was written in Java programming language using OpenCV library. OpenCV library methods were used to detect objects and structures in an image and classify them. The application plays simple sheet music scaled by the treble clef in the range of tones h to h”.

Klíčová slova: strojové vidění; přehrávač notového zápisu; OpenCV; kaskádové klasifikátory; Android; Java

Keywords: machine vision; sheet notes player; OpenCV; cascade classifiers; Android; Java

Děkuji vedoucímu mé práce doc. RNDr. Miroslavu Kolaříkovi, Ph.D. za velmi zajímavé téma ke zpracování i jeho ochotu při konzultacích v průběhu realizace aplikace. Také moc děkuji mé ženě Martině za neutuchající podporu a trpělivost během celé doby mého studia. Na závěr ještě děkuji mým synům za to, s jakou trpělivostí snášeli všechna omezení plynoucí z náročnosti příprav na zkoušky z jednotlivých předmětů i této práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Seznam zdrojových kódů

1 Úvod

Aplikace „Přehrávač not“ přehrává jednoduchý notový zápis (NZ) z jeho digitální fotokopie (DF) a slouží jako jednoduchý zpěvník. Přehrává NZ podle houslového klíče (HK). Rychlost přehrávání se nastavuje pro každou píseň zvlášť. V aplikaci se definují názvy písní, kterým jsou DF listů přiřazovány v době pořízení. Přehrání se dá provést pro konkrétní list nebo pro všechny listy písně. K pořízení DF se využívá digitální kamera zařízení.

Aplikace je sestavena pro zařízení s operačním systémem Android. Odzkoušena byla na zařízeních s operačním systémem Android od verze 4 po verzi 8. Na zařízeních s verzí 4 operačního systému nebyla aplikace plně funkční – nepodařilo se převést DF NZ na zvuk. Na zařízeních s vyššími verzemi operačního systému aplikace fungovala podle předpokladů.

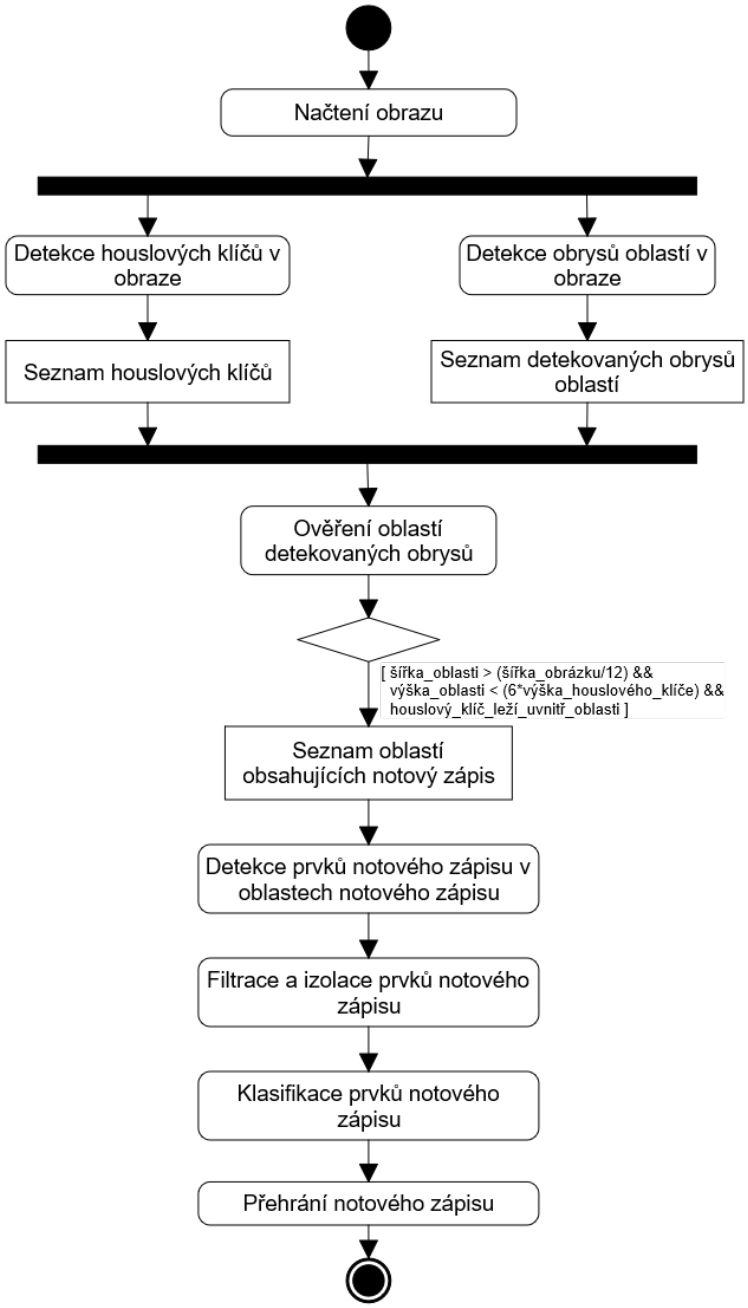
Aplikace je vytvořena v programovacím jazyku Java s využitím knihovny OpenCV. Knihovna OpenCV je využita pro detekci a klasifikaci prvků NZ v obraze. Aplikace je poměrně hardwarově náročná. Podle zkušeností pracuje přijatelně rychle na zařízeních, jejichž operační paměť je alespoň 2 GB.

V době programování aplikace jsem nenašel žádnou jinou, která by funkci přehrání DF NZ umožňovala.

2 Popis práce programu

2.1 Úvod

V této sekci přiblížím kroky nutné k převedení DF NZ na zvukový výstup, který by měl co nejvíce korespondovat s obsahem NZ (obr. 1).



Obrázek 1: Přehled aktivit přehrání digitální fotokopie notového zápisu

2.2 Načtení obrazu

Soubory pro načtení jsou uloženy ve formátu JPEG/JFIF v adresáři `SheetMusicPlayer`. Ten je podadresářem standardního adresáře prostředí pro ukládání obrázků uživatele. Jeho cesta je uložena v proměnné prostředí `DIRECTORY_PICTURES`. Obraz je po načtení pro účely analýzy převeden do odstínů šedi.

2.3 Detekce houslových klíčů v obraze

Pro určení pozice výskytu NZ na fotokopii je nutné určit grafický prvek, který by měl obsahovat každý NZ. Tento prvek musí být dostatečně výrazný a nezaměnitelný s jiným grafickým prvkem. V našem případě je takovým prvkem HK. Ten by měl stát vždy na začátku NZ.

NZ se zapisuje zleva doprava. Toho využívám k urychlení analýzy při detekci HK – není prováděna na celé DF, ale jen výřezu její levé části, který odpovídá třetině šířky DF.

K detekci HK v obraze jsou použity metody knihovny OpenCV s využitím natrénovaných kaskádových klasifikátorů (Haar Cascade Classifier). Tyto klasifikátory jsou trénovány na sadě pozitivních a negativních příkladů. Pozitivní příklady jsou vyznačené výskyty objektu (v našem případě HK) v DF. Negativní příklady jsou DF, kde se daný objekt nenachází. V tomto případě se jednalo o 153 příkladů pozitivních a 116 příkladů negativních (zdr. kód 1).

HK bývají na fotokopii různě deformované. Pro zvýšení pravděpodobnosti detekce jsem natrénoval 7 sad klasifikátorů s různým nastavením šířky pozitivního příkladu v rozmezí 12 a 18 pixelů. Výška pozitivního příkladu byla definována ve všech případech stejně a to 50 pixelů. Z těchto natrénovaných sad se pro vlastní detekci využívá 6 sad klasifikátorů – sada s nastavenou šířkou 17 pixelů detekuje HK často v místech, kde se nevyskytují.

Pozice a rozměry nalezených nepřekrývajících se oblastí výskytu HK jsou pro další použití uloženy jako prvky seznamu.

```
1 rm vystupA/*
2 opencv_createsamples -bgthresh 80 -info info.txt -w 15 -h
    ↪ 50 -vec pos-samplesA.vec -maxxangle 0.10 -maxyangle
    ↪ 0.10 -maxzangle 0.10
3 opencv_traincascade -featureType HAAR -data vystupA -vec
    ↪ pos-samplesA.vec -bg negativni.txt -precalcValBufSize
    ↪ 2048 -precalcIdxBufSize 2048 -numPos 153 -numNeg 116
    ↪ -nstages 25 -minhitrate 0.995 -w 15 -h 50
    ↪ -maxFalseAlarmRate 0.5
```

Zdrojový kód 1: Trénink klasifikátorů pro detekci houslového klíče

2.4 Detekce obrysů oblastí v obraze

V případě nalezení HK v obraze předpokládám, že se na fotokopii nachází také NZ příslušící HK. Je potřeba zjistit, jak velká je oblast NZ pro analýzu, která k danému HK přísluší.

Ke zjištění rozměrů této oblasti provádím, za využití metod knihovny OpenCV, vyhledání obrysů. Toto vyhledání obrysů je prováděno nad kopií obrázku v odstínech šedi, na kterou bylo aplikováno adaptivní prahování (zdr. kód 2).

```
1  Imgproc.adaptiveThreshold(imageGray, imageThreshold, 255,
       ↪ Imgproc.ADAPTIVE_THRESH_MEAN_C,
       ↪ Imgproc.THRESH_BINARY, 7, 2);
2  Imgproc.findContours(imageThreshold, contours, hierarchy,
       ↪ Imgproc.RETR_LIST, Imgproc.CHAIN_APPROX_SIMPLE);
```

Zdrojový kód 2: Vyhledávání obrysů v obraze

2.5 Vytvoření seznamu oblastí obsahujících notový zápis

Z detekovaných obrysů v obraze je nutné vybrat obrysy hranic NZ. Seznam oblastí ohraničených detekovanými obrysy je proto procházen podle klíče:

1. šířka oblasti je větší než dvanáctina šířky obrázku
2. výška oblasti je menší než šestinásobek výšky HK
3. HK leží uvnitř oblasti.

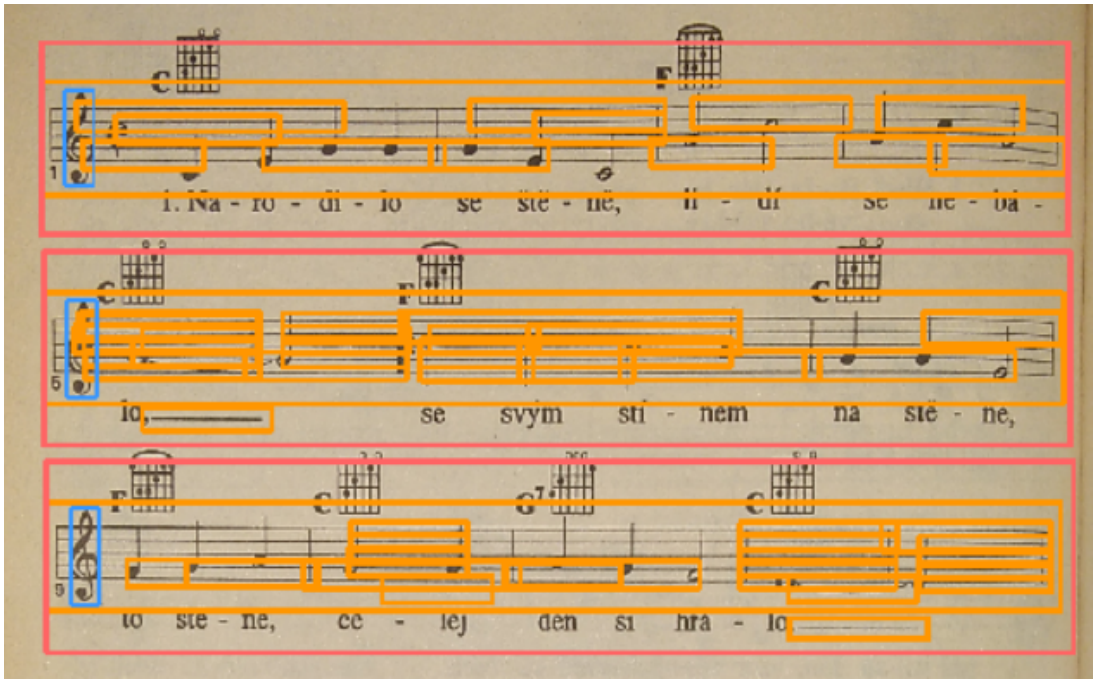
Po splnění všech tří podmínek klíče se porovnává šířka oblasti s doposud maximálně zjištěnou šířkou pro daný list DF. Pokud je širší, je její hodnota nastavena jako nová maximální šířka.

Na seznam oblastí určených pro další analýzu je přidána oblast definovaná parametry (obr. 2):

1. souřadnice levého horního rohu oblasti NZ podle horizontální osy odpovídají souřadnici levého horního rohu HK podle horizontální osy minus šířka HK
2. souřadnice levého horního rohu oblasti NZ podle vertikální osy odpovídají souřadnici levého horního rohu HK podle vertikální osy minus polovina výšky HK
3. šířka oblasti NZ odpovídá maximální zjištěné šířce oblasti
4. výška oblasti NZ odpovídá dvojnásobku výšky HK.

2.6 Detekce prvků notového zápisu v oblastech s notovým zápisem

Pro detekci prvků NZ v oblasti NZ využívám metody knihovny OpenCV s natrénovanými kaskádovými klasifikátory. V případě prvků NZ byly klasifikátory trénovány na 1323 příkladech pozitivních a 116 příkladech negativních (zdr. kód 3).



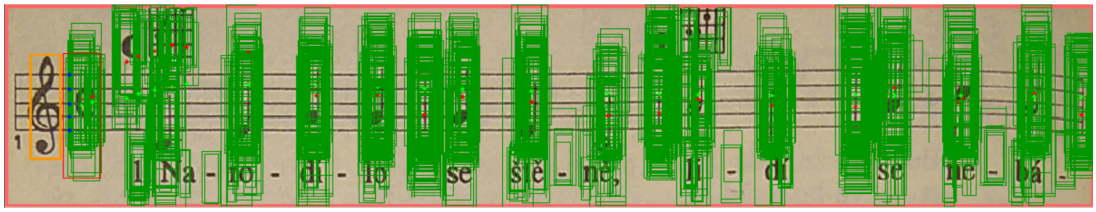
Obrázek 2: Oblasti notového zápisu
modrá kontura – detekovaný houslový klíč; oranžové kontury – detekované oblasti splňující nejmenší povolenou šířku a výšku; růžová kontura – výsledná oblast notového zápisu

Z důvodu zvýšení pravděpodobnosti detekce prvku NZ jsem natrénoval 3 klasifikátory, které se liší v nastavení šířky pozitivních příkladů. Ta nabývá hodnot 10, 14 a 17 pixelů. Výška je ve všech případech 50 pixelů.

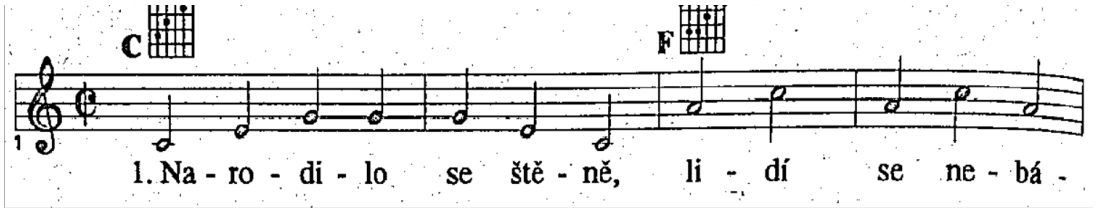
```
1  rm vystupA/*
2  opencv_createsamples -num 1323 -bgthresh 80 -info
       ↪ info.txt -w 10 -h 50 -vec pos-samplesA.vec
       ↪ -maxxangle 0.10 -maxyangle 0.10 -maxzangle 0.10
3  opencv_traincascade -featureType HAAR -data vystupA -vec
       ↪ pos-samplesA.vec -bg negativni.txt
       ↪ -precalcValBufSize 2048 -precalcIdxBufSize 2048
       ↪ -numPos $(((1323*905)/1000)) -numNeg 116 -nstages
       ↪ 25 -minhitrate 0.995 -w 10 -h 50 -maxFalseAlarmRate
       ↪ 0.5
```

Zdrojový kód 3: Trénink klasifikátorů pro detekci prvků notového zápisu

Při detekci prvků NZ dochází k vícenásobnému nalezení jednoho prvku a zároveň jsou detekovány i prvky, které součástí NZ nejsou (obr. 3). Výsledný seznam detekovaných prvků je seřazen podle levých okrajů oblastí jejich výskytu zleva doprava.



Obrázek 3: Detekované prvky notového zápisu



Obrázek 4: Aplikace adaptivního prahování na výřez notového zápisu

2.7 Filtrace a izolace prvků notového zápisu

Pro účely filtrace a izolace prvků provádím adaptivní prahování nad výřezem oblasti NZ (zdr. kód 4). Výsledkem je černobílý obraz (obr. 4), na kterém se provádí vlastní filtrace a izolace prvků NZ.

```
1       Imgproc.adaptiveThreshold(imageArea, imageThreshold,
           ↪ 255, Imgproc.ADAPTIVE_THRESH_GAUSSIAN_C,
           ↪ Imgproc.THRESH_BINARY, 19, 9);
```

Zdrojový kód 4: Adaptivní prahování výřezu notového zápisu

Filtrací rozumím nalezení všech prvků NZ, které nesou informaci potřebnou pro zvukovou interpretaci DF NZ a jsou součástí NZ (nejedná se o text písně, kytarové akordy apod.). Informace o poloze každého nalezeného prvku by měla být ve výsledném seznamu prvků pouze jednou.

Izolací rozumím nalezení horizontálního a vertikálního ohraničení prvku, které označuje oblast jeho výskytu takovou, že obsahuje celý prvek a zároveň nese veškerou informaci potřebnou k určení hodnoty prvku.

Při filtraci detekovaných prvků NZ se vychází z polohy a rozměrů HK. Ten se nachází na začátku příslušného NZ. Pro zpřesnění rozměrů HK se provádí ořez jeho vertikálních rozměrů tak, aby došlo k redukci bílého místa na horním a dolním okraji. Filtrace detekovaných prvků NZ se provádí dvoufázově.

V první fázi se prochází všechny detekované prvky NZ a na seznam pro filtraci v druhé fázi se přidávají ty, které splňují následující podmínky:

1. Souřadnice levého okraje detekovaného prvku na horizontální ose je větší než souřadnice levého okraje HK zvětšená o dvě třetiny šířky HK.
2. Rozdíl pozice středu detekovaného prvku podle vertikální osy vůči referenční hodnotě vypočítané ze středů předcházejících je menší než třetina

výšky HK.

Výchozí hodnotou referenční hodnoty je střed HK podle vertikální osy. Pokud jsou splněny první tři podmínky, potom se nová referenční hodnota vypočítá podle vzorce:

$$refHodnota = \frac{23 * refHodnota + 2 * středPrvkuPodleVertikálníOsy}{25}$$

3. Souřadnice pravého dolního rohu detekovaného prvku podle vertikální osy je větší než 1,2násobek referenční hodnoty nebo souřadnice levého okraje detekovaného prvku podle horizontální osy je menší než souřadnice levého okraje HK podle horizontální osy zvětšená o trojnásobek jeho šířky.
4. Vzdálenost středu prvku podle horizontální osy od předchozího prvku přidaného na seznam je větší než 0,8násobek šířky HK.

Po splnění všech 4 podmínek je na seznam přidán prvek daný průměrem souřadnic středů detekovaných prvků podle horizontální osy následujících od předchozího přidaného a referenční hodnotou, která udává střed prvku podle vertikální osy.

V druhé fázi se prochází všechny prvky z výsledného seznamu první fáze a produkuje se seznam nový. Provádí se tyto kroky:

1. Výpočet referenční hodnoty středu prvku podle vertikální osy. Výchozí hodnotou referenční hodnoty je střed HK podle vertikální osy. Dojde k výpočtu nové referenční hodnoty podle vzorce

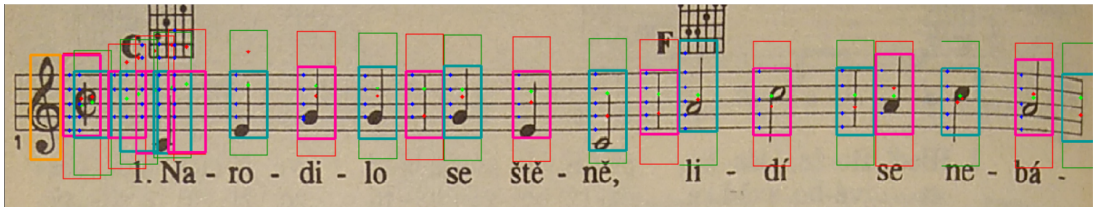
   $$refHodnota = \frac{23 * refHodnota + 2 * středPrvkuPodleVertikálníOsy}{25}$$

   Pokud je rozdíl mezi novou a předchozí referenční hodnotou větší než 10 %, je nová referenční hodnota přepočítána podle vzorce

   $$refHodnota = \frac{992 * refHodnota + 8 * středPrvkuPodleVertikálníOsy}{1000}$$

   Konstanty použité v předchozím vzorci zajišťují, že i při velkém posunu středu prvku podle vertikální osy oproti referenční hodnotě nedojde k posunu nově vypočítané referenční hodnoty mimo NZ.
2. Porovnávají se po sobě následující středy prvků podle horizontální osy. Pokud je horizontální vzdálenost středu prvku od posledního přidaného na seznam větší než polovina šířky HK, je na výsledný seznam druhé fáze přidán prvek daný průměrem souřadnic středů detekovaných prvků podle horizontální osy následujících od předchozího přidaného a referenční hodnotou, která udává střed prvku podle vertikální osy.

Výsledkem druhé fáze je seznam bodů definujících středy oblastí, kde by se měly nacházet prvky NZ. Pro každý takový bod je potřeba určit rozsah oblasti,



Obrázek 5: Výsledek filtrace a izolace prvků notového zápisu

která by měla obsahovat prvek NZ. V horizontálním směru bude středem oblasti horizontální souřadnice bodu a šířkou 1,2násobek šířky HK. Ve vertikálním směru bude středem oblasti vertikální souřadnice bodu a výškou 1,2násobek výšky HK.

Do takto definovaných oblastí často zasahují grafické prvky, které nenesou informaci nutnou k přehrání daného prvku NZ. Ty je nutné z oblasti odstranit jejím zmenšením ve vertikálním směru tak, aby v dané oblasti zůstal jen prvek NZ vhodný pro následnou klasifikaci. To provádím procesem izolace.

Izolace se provádí v oblasti prvku NZ z výřezu NZ, na který bylo aplikováno adaptivní prahování (obr. 4). Pro každý řádek oblasti je spočítána průměrná hodnota ze všech bodů nacházejících se na daném řádku. Průměrná hodnota nabývá hodnot v rozmezí 0 až 255.

Izolace v horní části oblasti prvku NZ probíhá od horního okraje. V případě, že od prvního řádku jsou průměrné hodnoty větší než 240, je shora oříznuta celá oblast výskytu těchto hodnot. V případě, že jsou průměrné hodnoty hned od prvního řádku menší nebo rovny 240, hledá se v horní části oblasti prvku NZ podoblast, kde by byly hodnoty větší než 240 a počet řádků takové podoblasti je větší než 10 % všech řádků oblasti prvku NZ. Pokud je taková podoblast nalezena, je oříznuta část oblasti prvku NZ od prvního řádku oblasti prvku NZ po nejspodnější řádek této nalezené podoblasti.

Izolace v dolní části oblasti prvku NZ se týká spodních 42 % této oblasti. Zde je hledán horní okraj podoblasti větší než 10 % počtu všech řádků oblasti prvku NZ, kde byly všechny průměrné hodnoty vyšší než 240. Pokud je taková podoblast nalezena, jsou oříznuty všechny řádky oblasti prvku NZ od horního okraje této podoblasti po poslední řádek oblasti prvku NZ.

Počet řádků mezi horním a spodním okrajem izolovaného prvku NZ by měl být nejméně 77,5 % výšky HK.

Tímto postupem jsou stanoveny polohy a rozměry oblastí prvků NZ (obr. 5), u kterých je následnou klasifikací možné určit jejich konkrétní hodnotu.

2.8 Klasifikace prvků notového zápisu

K určení hodnot oblastí prvků NZ využívám algoritmus k-nejbližších sousedů. Trénovací množina algoritmu má 8716 prvků rozdělených nerovnoměrně do 105 tříd (tab. 1). Nerovnoměrné rozdělení je způsobené nedostatkem vhodných vzorků. Velikost vektoru prvku trénovací množiny je 30 sloupců a 60 řádků.

Velikost každé oblasti prvku NZ určené ke klasifikaci je upravena tak, aby její

rozměry v době klasifikace odpovídaly šířce 30 pixelů a výšce 60 pixelů. Následně je klasifikována algoritmem k-nejbližších sousedů. Při klasifikaci je vyhledáván nastavený počet nejbližších sousedů, v našem případě 4. Převažující třída je určena jako třída klasifikovaného prvku a určuje jeho hodnotu, ze které je možné odvodit název tónu a jeho délku. Klasifikovaný prvek je přidán na seznam prvků pro přehrání.

2.9 Přehrání notového zápisu

Seznam prvků pro přehrání je seznamem názvů DF listů NZ a dalších informací nutných k přehrání a zobrazení přehrávané DF listu. Jsou to informace o umístění a rozměrech HK na DF listu, dále informace o umístění a rozměrech přehrávaných oblastí na DF listu a nakonec umístění, rozměry a klasifikovaná hodnota prvku NZ v přehrávané oblasti DF listu.

Každá DF listu je přehrávána po jednotlivých oblastech NZ v pořadí podle umístění na DF listu od horního okraje ke spodnímu. Prvky NZ jsou přehrávány podle umístění v oblasti NZ od levého okraje k pravému. DF listů NZ jsou přehrávány podle pořadí pořízení DF od nejstarší po nejnovější.

Klasifikovaná hodnota prvku NZ obsahuje název tónu a délku tónu. Název tónu je pro účely přehrání převeden na název mediálního souboru ve formátu MP3. Doba, po kterou bude soubor přehráván je dána délkou tónu. Ta je v hodnotě prvku udána číselnou hodnotou (01 – celá nota, 02 – půlová nota, 04 – čtvrťová nota, 08 – osminová nota, 16 – šestnáctinová nota). Doba přehrávání je z délky tónu a zadané rychlosti přehrávání písně vypočítána podle vzorce:

$$dobaPřehráníTónu(ms) = \frac{60 * \frac{4}{délkaTónu}}{nastavenáRychlostPřehrávání} * 1000$$

Pokud je na začátku NZ posuvka určující snížení nebo zvýšení výšky tónu a je správně klasifikována, je pro název tónu, kterého se posuvka týká, přehrán MP3 soubor obsahující takto zvýšený nebo snížený tón.

Pokud hodnota prvku určuje pomlku, je přehrávání všech tónů pozastaveno na vypočítaný počet milisekund.

Tabulka 1: Počty prvků vybraných tříd trénovací množiny

Třída		Počet prvků
Nota	Délka noty	
c’	celá	3
c’	půlová	30
c’	čtvrťová	147
c’	osminová	127
c’	šestnáctinová	11
d’	celá	5
d’	půlová	42
d’	čtvrťová	231
d’	osminová	195
d’	šestnáctinová	29
e’	celá	5
e’	půlová	67
e’	čtvrťová	302
e’	osminová	358
e’	šestnáctinová	66
f’	celá	16
f’	půlová	43
f’	čtvrťová	267
f’	osminová	274
f’	šestnáctinová	38
g’	celá	2
g’	půlová	48
g’	čtvrťová	355
g’	osminová	339
g’	šestnáctinová	51
a’	celá	4
a’	půlová	37
a’	čtvrťová	257
a’	osminová	357
a’	šestnáctinová	41
h’	celá	9
h’	půlová	44
h’	čtvrťová	189
h’	osminová	223
h’	šestnáctinová	34
pomlka	celá	121
pomlka	půlová	211

3 Omezení

Některé NZ jsou velmi složité a často obsahují mnoho informací, které s interpretací NZ nesouvisí. Příkladem může být text písně případně akordy pro kytaru a jiné. Nelze postihnout všechny případy.

3.1 Tónový rozsah, hodnota noty, posuvky

Zvolená metoda je určena k přehrávání jednoduchých NZ, kdy nedochází k paralelnímu přehrávání tónů.

Tónový rozsah podle HK, který lze bezpečně určit (nota je správně oříznuta ve většině případů) je v rozmezí tónů h a h".

Délky tónů, které se programem detekují jsou od nejdelší celé noty po nejkratší šestnáctinovou notu. S prodlouženími délky tónu o polovinu, které je v NZ vyjádřeno tečkou za notou, program nepracuje.

Posuvky určující snížení nebo zvýšení tónu o půltón, jsou detekovány a analyzovány pouze na začátku NZ v blízkosti HK. Pokud stojí přímo u noty, ukázalo se jejich určení jako velmi problematické.

3.2 Hardware a operační systém

Variabilnost NZ a nepřesnosti izolace prvku NZ vedou k tomu, že je nutné pro klasifikaci prvků algoritmem k-nejbližších sousedů vytvořit dostatečně velkou trénovací množinu. Pro účely této práce se mi podařilo vytvořit trénovací množinu prvků, jejichž počet je 8716. Před zahájením analýzy DF listu NZ je nutné celou tuto množinu načíst a natrénovat jí algoritmus k-nejbližších sousedů. Tato operace je paměťově a časově náročná a rychlost natrénování se velmi liší podle typu hardware zařízení.

Testování programu na různých zařízeních také ukázalo, že verze knihovny OpenCV (3.4.1), která byla použita, není plně funkční ve všech verzích operačního systému Android. Klasifikace prvků NZ ve verzích operačního systému nižších než Android 5 skončí přerušením vykonávání kódu programu a pádem aplikace (tab. 2).

4 Programátorská dokumentace

4.1 Použité technologie a knihovny

4.1.1 Java

Java je primárně objektově orientovaný programovací jazyk od roku 2014 s prvky funkcionálního programování. Byl vyvinut společností Sun Microsystems a představen v roce 1995. Roku 2007 Sun uvolnil zdrojové kódy Javy od té doby je vyvíjena jako jazyk s veřejně dostupným zdrojovým kódem. Po akvizici společnosti Sun Microsystems společností Oracle je Oracle současným vlastníkem oficiální

Tabulka 2: Funkčnost analýzy NS v závislosti na typu hardware, verzi operačního systému Android, velikosti operační paměti a doba nutná pro natrénování algoritmu k-nejbližších sousedů

Název zařízení	Verze OS	Typ procesoru	Paměť (GB)	Nahrání trénovací množiny algoritmu k-nejbližších sousedů (sekundy)	Funkční analýza NS
LG Optimus F6 D505	4.4.2	Qualcomm Snapdragon S4 Plus 1,2 GHz, 2-core	1	505,79	Ne
Acer Iconia One 8 B1-810	4.4.4	Intel Atom Z3735G 1,3 GHz, 4-core	1	152,18	Ne
Prestigio MultiPad 3797 3G	5.1.1	Intel Atom x3 C3230-RK 1,2 GHz, 4-core	1,5	415,12	Ano
Vodafone VFD 500	6.0	ARM Cortex A53 1,0 GHz, 4-core	1	280,419	Ano
ZTE Blade A512	6.0.1	ARM Cortex-A53 1,4 GHz 4-core	2	191,583	Ano
Vodafone VFD 820	8.1.0	ARM Cortex-A53 1,8 GHz 8-core	3	38,361	Ano

implementace Java SE platformy. V současnosti je to jeden z nejpoužívanějších programovacích jazyků na světě.

Jeho syntaxe vychází z jazyka C a C++. Jedná se o interpretovaný jazyk, takže místo strojového kódu se vytváří pouze tzv. bajtkód. Díky tomu program může běžet na libovolném zařízení, kde je dostupný interpret bajtkódu nazývaný virtuální stroj Javy (Java Virtual Machine).

Jazyk Java je používán jako základní prostředek pro vývoj aplikací pro operační systém Android, který však využívá svou vlastní knihovnu tříd a Java se využívá pouze jako prostředek syntaxe.[8]

4.1.2 OpenCV

Knihovna OpenCV (Open Source Vision Library) je softwarová knihovna s veřejně dostupným zdrojovým kódem pro počítačové vidění a strojové učení. Po-

skytuje společnou infrastrukturu pro aplikace počítačové vidění a byla vytvořena pro urychlení používání strojového vnímání v komerčních aplikacích. Z toho důvodu je poskytována pod BSD licencí.

Původně byla knihovna vyvíjena a v roce 1999 představena společností Intel. První oficiální verze byla vydána s označením 1.0 v roce 2006. Od té doby se stále vyvíjí a v současnosti je aktuální verzí verze 4.1.2. Pro účely této práce byla použita verze 3.4.1. Podpora OpenCV byla v srpnu roku 2012 převzata neziskovou organizací OpenCV.org, která udržuje vývojářské a uživatelské stránky projektu.

Knihovna je napsána v jazyce C a C++. Součástí jsou rozhraní pro programovací jazyky C++, Python, Java a MATLAB. Podporuje operační systémy Windows, Linux, Android a Mac OS.

Knihovna obsahuje více než 2500 algoritmů, které mohou být použity k detekci a rozpoznávání obličejů, identifikaci objektů, sledování pohybujících se objektů, nalezení podobných obrázků z databáze atd.[10, 11]

V této práci byla použita pro detekci a klasifikaci prvků NZ.

4.1.3 SQLite

Knihovna SQLite je softwarovou knihovnou vyvinutou v programovacím jazyce ANSI-C, která poskytuje relační sytém řízení báze dat. Nejedná se o tradiční architekturu systému řízení báze dat, která je založena na architektuře klient/server, kde se pro přístup do databáze využívá spojení pomocí protokolu TCP/IP. SQLite databáze je integrována s aplikací, která do SQLite databáze přistupuje přímo čtením a zápisem do souborů uložených na disku.[9]

SQLite databáze nevyžaduje žádné nastavení a instalaci a je integrována přímo v prostředí operačního systému Android. Z toho důvodu se výborně hodí pro ukládání aplikačních dat.

V této práci byla použita pro ukládání základních dat o písních a DF listů NZ, jako jsou jejich názvy, nastavení a příslušnost listů k písním.

4.2 Struktura projektu

Celá struktura a rozložení adresářů je automaticky generováno vývojovým prostředím, ve kterém byl projekt vytvořen. V tomto případě to je Android Studio verze 3.0.1. Umístění knihoven, mediálních souborů a dalších zdrojových souborů je pevně dané a vlastní struktura je velmi rozvětvená. Proto popíši umístění jen nejdůležitějších částí zdrojových dat.

4.2.1 Knihovna OpenCV

Soubory související s knihovnou OpenCV verze 3.4.1 se nachází v adresáři „`/src/SheetMusic/openCVLibrary341/`“.

4.2.2 Kaskádové klasifikátory

Soubory natrénovaných kaskádových klasifikátorů se nachází v adresáři „`/src/SheetMusic/app/src/main/res/raw/`“. Maska souborů obsahujících klasifikátory pro detekci HK je „`hkcascade[a-g].xml`“. Maska souborů obsahujících klasifikátory pro detekci prvků NZ je „`npcascade[a-c].xml`“.

4.2.3 Trénovací množina algoritmu k-nejbližších sousedů

Soubory sloužící ke klasifikaci prvků NZ se nachází v adresáři „`/src/SheetMusic/app/src/main/assets/`“. Přestože soubory mají příponu souborů archivu souborů ZIP, jedná se o formát souborů GZIP. Přípona „`zip`“ byla zvolena, neboť soubory s příponou „`gz`“ nebylo možné ve vývojovém prostředí načíst.

4.2.4 Zvuky tónů

Soubory se zvuky tónů ve formátu MP3 se nachází v adresáři
„`/src/SheetMusic/app/src/main/res/raw/`“
a jejich maska je „`piano*.mp3`“.

4.2.5 Zdrojové kódy v programovacím jazyku Java

Zdrojové kódy tříd aplikace se nachází v adresáři
„`/src/SheetMusic/app/src/main/java/nemecek/petr/sheetmusic/`“.

Soubory obsahují zdrojové kódy následujících tříd:

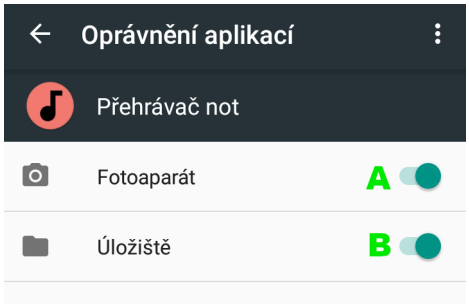
- **AnalyzeSheetActivity**
  Třída provádí analýzu DF listů NZ a vytváří seznam prvků NZ pro přehrání.
- **CameraPreview**
  Třída zajišťuje správné zobrazení náhledu při pořizování DF listu písně.
- **CameraView**
  Třída pořizuje DF listu písně.
- **ImageView**
  Třída načítá, zobrazuje, maže, spouští analýzu a přehrání DF listu NZ.
- **MainScreen**
  Třída úvodní obrazovky načítá a zobrazuje seznamu uložených písní, vkládá nové písně, filtruje podle názvů písní, volí píseň.
- **MyDB**
  Třída zajišťuje komunikaci s databází SQLite, kde jsou uloženy informace o uložených písních a jejich listech.
- **PlaySheetActivity**
  Třída přehrává a graficky znázorňuje přehrání analyzovaného listu písně.

- **SongmainActivity**
  Třída načítá a zobrazuje názvy listů, maže listy, nastavuje rychlost přehrávání písně, volí list písně pro zobrazení.

4.2.6 Konfigurační soubory prvků obrazovek aplikace

Definice a rozložení prvků obrazovek je definováno v souborech formátu XML, které jsou uloženy v adresáři
„`/src/SheetMusic/app/src/main/res/layout/`“.

- **activity_analyze_sheet.xml**
  Konfigurační soubor obrazovky příslušný třídě AnalyzeSheetActivity.
- **activity_camera_view.xml**
  Konfigurační soubor obrazovky příslušný třídě CameraView.
- **activity_image_view.xml**
  Konfigurační soubor obrazovky příslušný třídě ImageView.
- **activity_main_screen.xml**
  Konfigurační soubor obrazovky příslušný třídě MainScreen.
- **activity_play_sheet.xml**
  Konfigurační soubor obrazovky příslušný třídě PlaySheetActivity.
- **activity_songmain.xml**
  Konfigurační soubor obrazovky příslušný třídě SongmainActivity.



Obrázek 6: Oprávnění aplikace
A – tlačítko udělení oprávnění přístupu k fotoaparátu; B – tlačítko udělení oprávnění přístupu k úložišti souborů

5 Uživatelská dokumentace

V následující sekci popíši, jak přidělit aplikaci potřebná práva k využívání zdrojů hardwaru a softwaru a možnosti, které poskytuje uživatelské rozhraní aplikace. Cílem je možnost, použít aplikaci nejenom k analýze a přehrání DF listu NZ, ale také jako jednoduchý zpěvník. Aplikace automaticky detekuje nastavený jazyk systémového prostředí a podle toho se může přepnout do tří jazykových módů – anglického, českého a německého. Jako výchozí je zvolen anglický. Pro účely této práce je zvolen mód český. Aplikace se při zvoleném českém módu jmenuje „Přehrávač not“.

5.1 Oprávnění aplikace

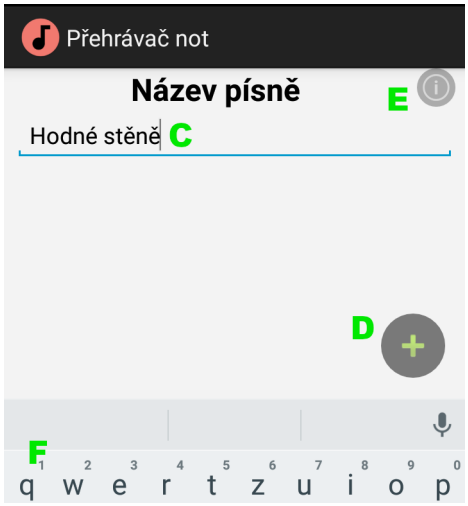
Před použitím je nutné udělit aplikaci oprávnění používat fotoaparát zařízení k pořízení DF listu NZ a přistupovat k úložišti, kam bude ukládat a odkud bude načítat jednotlivé DF listů NZ (obr. 6). Možnost nastavení oprávnění aplikací se v operačním systému Android nalézá v nastavení systému v sekci „Aplikace“.

5.2 Uživatelské rozhraní aplikace

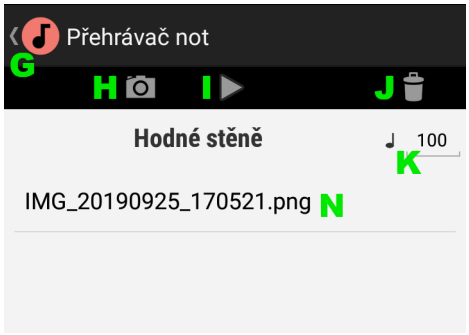
Po spuštění aplikace se objeví úvodní obrazovka aplikace (obr. 7), která obsahuje seznam již zadaných písní.

V zadaných písních lze vyhledávat psaním části řetězce názvu písně do pole pro zadání názvu písně. Toto vyhledávání není citlivé na velikost písmen.

V případě zadávání nové písně se do pole pro zadání názvu písně zadá její název a stiskne se tlačítko „+“ pro přidání nové písně. Klepnutí na název zadané písně nebo přidání nové písně spustí obrazovku vybrané písně (obr. 8). Ovládací prvky na této obrazovce umožňují přejít k pořízení nové DF listu písně, analyzovat a přehrát všechny DF listů zvolené písně, smazat všechny údaje a soubory vztahující se ke zvolené písni a zadat tempo přehrávání písně. Tempo přehrávání



Obrázek 7: Obrazovka se seznamem písní
C – pole zadání názvu písně; D – tlačítko přidání nové písně; E – tlačítko informace o aplikaci; F – virtuální klávesnice



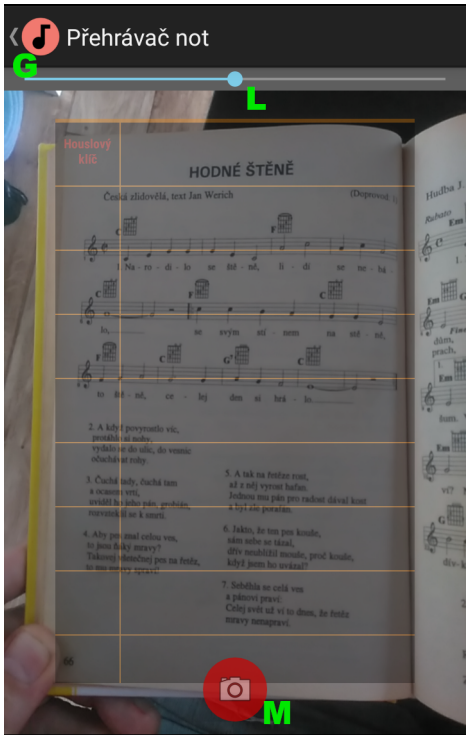
Obrázek 8: Obrazovka vybrané písně
G – tlačítko návratu na předchozí obrazovku; H – tlačítko spuštění fotoaparátu; I – tlačítko analýzy a přehrání listů písně; J – tlačítko smazání písně; K – pole zadání rychlosti přehrávání; N – název souboru digitální fotokopie listu písně

se udává jako počet úderů metronomu za 1 minutu a zadává se do textového pole označeného čtvrťovou notou.

Obrazovka vybrané písně obsahuje seznam názvů souborů již pořízených DF listů písně v pořadí, v jakém byly pořízeny. Klepnutím na název souboru nebo bezprostředně po pořízení se DF zobrazí (obr. 10).

Klepnutím na ikonu aplikace v levém horním rohu umožňuje návrat na obrazovku se seznamem písní.

Klepnutím na ikonu fotoaparátu se spustí obrazovka pořizování DF (obr. 9). Klepnutí na ikonu fotoaparátu v červeném poli vede k pořízení DF. Při pořizování DF je nutné dbát na to, aby fotografovaný list ležel uvnitř ztmavené oblasti náhledu, HK ležely při levém okraji této oblasti a NZ byl pokud možno co nejvíce rovnoběžný s horizontálními linkami v oblasti. Ovládacím prvkem v horní části

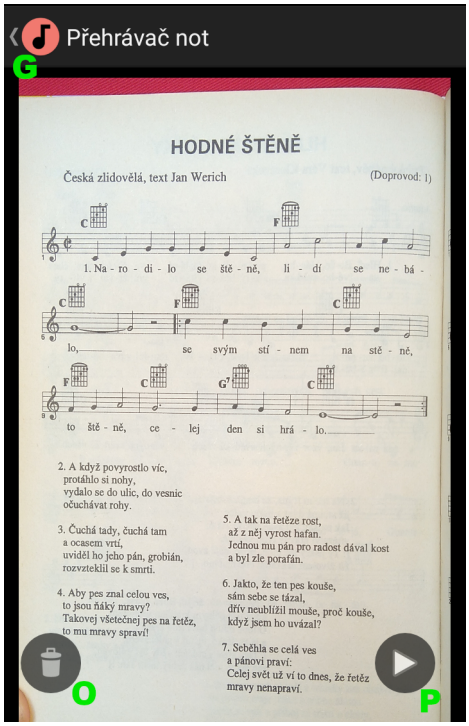


Obrázek 9: Obrazovka pořizování digitální fotokopie listu písně
G – tlačítko návratu na předchozí obrazovku; L – nastavení korekce a kompenzace expozice; M – pořízení digitální fotokopie

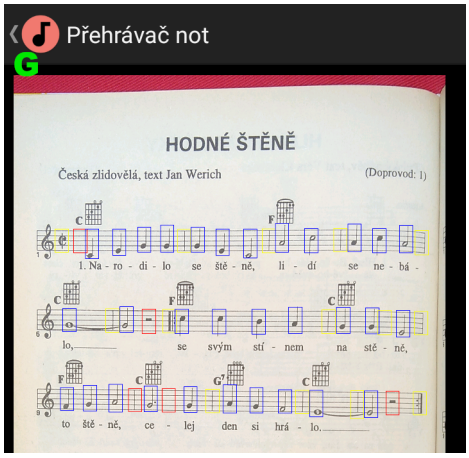
náhledu je možné provést posunem vlevo korekci a posunem vpravo kompenzaci expozice. Pořízená DF se zobrazí na obrazovce listu písně (obr. 10).

Z obrazovky listu písně lze list smazat nebo spustit analýzu a přehrání listu. Pokud píseň obsahuje více listů, lze přejetím prstem doleva či vpravo po obrazovce načíst následující nebo předcházející list.

Po spuštění analýzy a jejím provedení se začne list přehrávat (obr. 11). Přehrávané prvky jsou postupně označovány ohraničením modré, červené a žluté barvy. Modrá barva označuje prvky, které jsou přehrány, červená barva označuje pomlky a žlutá prvky, které nejsou přehrány. Stisk ikony v levém horním rohu vede k ukončení přehrávání a návratu na předchozí obrazovku.



Obrázek 10: Obrazovka listu písně
G – tlačítko návratu na předchozí obrazovku; O – tlačítko smazání listu písně;
P – tlačítko analýzy a přehrání listu písně



Obrázek 11: Obrazovka přehrávání listu písně
G – tlačítko zastavení přehrávání a návratu na předchozí obrazovku

Závěr

Byla vytvořena aplikace, která detekuje a následně automaticky přehrává notový zápis obsažený v digitální fotokopii. Z důvodu komfortního používání je aplikace vytvořena pro přenosná zařízení s operačním systémem Android, jako jsou mobilní telefony a tablety. Součástí většiny těchto zařízení je kamera, kterou je možné digitální fotokopii pořídit.

Aplikace je vytvořena v programovacím jazyce Java. K detekci prvků notového zápisu a jejich klasifikaci je použita knihovna OpenCV. Kaskádové klasifikátory detektoru využívají Haarových příznaků. Pro klasifikaci prvků notového zápisu se využívá algoritmu k-nejbližších sousedů. Aplikace je schopna přehrávat jednoduché notové zápisy podle houslového klíče v rozsahu tónů h až h". Ve většině případů dokáže odfiltrovat okolí notového zápisu (text písně, kytarové akordy) a ke klasifikaci poslat správný výřez prvku notového zápisu. Jednotlivé znaky notového zápisu nejsou rovnoměrně zastoupeny v trénovací množině pro klasifikátor a z toho důvodu dochází u některých prvků notového zápisu k jejich chybné klasifikaci a následně chybnému přehrání. Trénovací množina klasifikátoru obsahuje v současnosti 8716 prvků. Je žádoucí ji rozšířit a zajistit rovnoměrné rozložení znaků notového zápisu v trénovací množině.

Z důvodu velikosti trénovací množiny je aplikace poměrně hardwarově náročná a podle zkušeností pracuje uspokojivě na mobilních zařízeních s operační pamětí nejméně 2 GB. Správná funkcionalita aplikace byla ověřena na zařízeních s operačním systémem Android verze 5 až verze 8. Na zařízeních s nižšími verzemi operačního systému Android nefungovala.

Pořízené digitální fotokopie notových zápisů je možné v aplikaci ukládat a přiřazovat názvům písní. Z toho důvodu lze aplikaci využít také jako zpěvník.

Přestože má aplikace u některých notových zápisů problém se správnou detekcí a klasifikací prvků notového zápisu, ukazuje jeden z možných směrů, jakým by se k této problematice dalo přistupovat. Práce při zpracování tohoto tématu byla pro mě velmi objevná a zajímavá. Na zdokonalení aplikace bych chtěl pracovat i po dokončení studia.

Conclusions

There was created an application that detects and then automatically plays the notation in digital photocopy. For ease of use, the application is designed for portable devices running operating system Android, such as mobile phones and tablets. Most of these devices have a camera that can be used for digital photocopying.

The application is created in Java programming language. The OpenCV library is used to detect the notation elements and their classification. Cascade detector classifiers use Haar-like features. The k-nearest neighbor algorithm is used to classify musical notation elements. The application is able to play simple musical notes according to the treble clef in the range of tones h to h". In most cases, it can filter out the surroundings of musical notes (song lyrics, guitar chords). The training set of the classifier currently contains 8716 elements. It is desirable to extend it and to ensure an even distribution of the characters of the notation in the training set.

Due to the size of the training set, the application is relatively hardware intensive and, according to experience, works satisfactorily on mobile devices with at least 2 GB memory. The correct functionality of the app has been verified on devices running operating system Android version 5 through version 8. It did not work on devices with lower versions of operating system Android.

Recorded digital photocopies of musical notation can be saved in the application and assigned to song titles. For this reason, the application can also be used as a songbook.

Although the application has a problem with the correct notation and classification of the notation elements for some scores, it shows one of the possible ways in which this issue could be approached. Working on this topic was very revealing and interesting for me. I would like to work on improving the application even after graduation.

A Obsah přiloženého DVD

Na samotném konci textu práce je uveden stručný popis obsahu přiloženého DVD, tj. jeho závazné adresářové struktury, důležitých souborů apod.

bin/
: Instalátor aplikace Přehrávač not ve formátu Android Package.

doc/
: Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/
: Kompletní zdrojové texty programu Přehrávač not se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu.

readme.txt
: Instrukce pro instalaci a spuštění programu Přehrávač not, včetně všech požadavků pro jeho bezproblémový provoz.

Navíc DVD obsahuje:

houslovyKlicPozice/
: Trénovací sada obrázků, konfigurační soubory pozitivních a negativních vzorků, trénovací skript a výsledné klasifikátory pro detekci HK.

notyHodnota/
: Trénovací sada obrázků prvků NZ roztříděná do adresářů podle hodnoty, skript generující soubory a výsledné soubory sloužící jako trénovací množina vektorů pro algoritmus k-nejbližších sousedů.

notyPozice/
: Trénovací sada obrázků, konfigurační soubory pozitivních a negativních vzorků, trénovací skript a výsledné klasifikátory pro detekci prvků NZ.

U veškerých cizích převzatých materiálů obsažených na DVD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na DVD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.

Seznam zkratek

DF	digitální fotokopie
HK	houslový klíč
NZ	notový zápis

Literatura

[1] ROSENBROCK, A. Practical Python and OpenCV: An Introductory, Example Driven Guide to Image Processing and Computer Vision. 3rd Edition. 2016. Dostupné z: https://www.pyimagesearch.com/

[2] ROSENBROCK, A. Practical Python and OpenCV: Case Studies. 3rd Edition. 2016. Dostupné z: https://www.pyimagesearch.com/

[3] BATCHELOR, B.G. Machine vision handbook. London: Springer, 2012. ISBN 978-1-84996-168-4.

[4] HLAVÁČ, V., ŠONKA, M. Počítačové vidění. Praha: Grada, 1992. ISBN 978-80-8542-467-6.

[5] VOLNÁ, E. Umělá inteligence: rozpoznávání vzorů v dynamických datech. Praha: BEN – technická literatura, 2014. ISBN 978-80-7300-497-2.

[6] RUSSIS, L.D., SACCO, A. OpenCV Java Tutorials [online]. 2019-09-08. [cit. 2019-09-08]. Dostupné z: https://opencv-java-tutorials.readthedocs.io/

[7] ALPHABET INC. Developer Guides [online]. 2018-03-02. [cit. 2018-03-02]. Dostupné z: https://developer.android.com/guide/

[8] Java (programovací jazyk) [online]. 2019-11-01. [cit. 2019-11-01]. Dostupné z: https://cs.wikipedia.org/wiki/Java_(programovací_jazyk)

[9] What is SQLite. [online]. 2019-11-01. [cit. 2019-11-01]. Dostupné z: https://www.sqlitetutorial.net/what-is-sqlite/

[10] About OpenCV. [online]. 2019-11-01. [cit. 2019-11-01]. Dostupné z: https://opencv.org/about/

[11] OpenCV. [online]. 2019-11-01. [cit. 2019-11-01]. Dostupné z: https://en.wikipedia.org/wiki/OpenCV

[12] OpenCV TEAM. Cascade Classifier. [online]. 2019-11-02. [cit. 2019-11-02]. Dostupné z: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[13] OPENCV TEAM. Cascade Classifier Training. [online]. 2019-11-02. [cit. 2019-11-02]. Dostupné z: https://docs.opencv.org/3.4/dc/d88/tutorial_traincascade.html

[14] OpenCV TEAM. K-Nearest Neighbour. [online]. 2019-11-02. [cit. 2019-11-02]. Dostupné z: https://docs.opencv.org/master/d0/d72/tutorial_py_knn_index.html