



System pro automatický asistovaný překlad na mobilních zařízeních s OS Android

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie
Autor práce: **Bc. Václav Bárta**
Vedoucí práce: doc. Ing. Josef Chaloupka, Ph.D.



Prohlášení autora

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum 15. 05. 2014

Podpis Bc. Václav Bárta

Anotace

Tato práce pojednává o implementaci aplikace pro asistovaný překlad pro platformu Android včetně rozpoznávání textu z obrázku. Součástí práce je seznámení s problematikou asistovaného překladu, strojového překladu a souvisejících technik. Aplikace byla vytvořena s využitím těchto technologií: TTS, OCR, Java.

Klíčová slova

asistovaný překlad, android, ocr, tts, java, tezaurus

Title

System for automatic assisted language translation on mobile devices with operating system Android

Annotation

This thesis deals with implementation of application for computer assisted translation to Android platform, even with optical character recognition. Thesis includes introduction into computer assisted translation, automatic translation and related technique. Application was created by using following technologies: TTS, OCR and Java.

Keywords

computer assisted translation, android, ocr, tts, java, thesaurus

Obsah

Seznam zkratk	7
Seznam obrázků	8
Seznam tabulek	10
1 Úvod	11
1.1 Požadované vlastnosti vyvíjené aplikace	11
2 Implementace	12
2.1 Specifika vývoje aplikací pro chytré telefony	12
2.2 Aplikace pro operační systém Android	12
2.2.1 Aktivity a služby.....	12
2.2.2 Dodavatelé obsahu a záměry	14
2.2.3 Konfigurace aplikace	14
2.2.4 Aplikační prostředky	15
2.3 Vlastní implementace	16
2.3.1 Use Case diagramy	16
2.3.2 Realizace klíčových částí aplikace	19
2.3.3 Překladačové jádro	19
2.3.4 Strojový překlad	19
2.3.5 Rozpoznávání textu z obrazu.....	20
2.3.6 Rozhraní pro slovníky a tezaury	21
2.3.7 Hlasová syntéza	21
2.3.8 Práce se soubory	22
2.3.9 Nastavení aplikace	22
2.4 Aktivity aplikace	23
2.5 Služby aplikace.....	24
2.6 Použité knihovny / komponenty	24
2.6.1 tesseract-android-tools.....	24
2.6.2 microsoft-translator-java-api	25
2.6.3 BigHugeThesaurus	25
2.6.4 cpdetector.....	25
2.6.5 ZoomableImageView	26
2.6.6 Andexplorer	26

2.7	Popis implementace	27
2.7.1	TranslatorHomeActivity	27
2.7.2	TranslatorAssistantActivity	27
2.7.3	TranslatorProcessFragmentActivity	28
2.7.4	TranslatorResultActivity	30
2.7.5	TranslatorDictionaryActivity	30
2.7.6	TesseractOCRActivity	31
3	Implementace systému audio-vizuální syntézy řeči.....	34
3.1	Systém audio-vizuální syntézy řeči	34
3.2	Implementace.....	34
4	Vlastnosti aplikace, navržené funkce a jejich využití.....	38
4.1	Minimální softwarové požadavky pro zprovoznění navržené aplikace	38
4.2	Jazyky podporované aplikací	39
4.2.1	Podpora pro strojový překlad	39
4.2.2	Podpora pro rozpoznávání skenovaného textu	39
4.3	Slovníky a tezaury	40
4.4	Startovací obrazovka	40
4.5	Nastavení aplikace	41
4.6	Práce se soubory	42
4.7	Překlad z textu	43
4.7.1	Překládaný text	43
4.7.2	Překlad	43
4.7.3	Přeložený text	45
4.7.4	Rozpoznávání textu z obrazu.....	45
4.8	Rozpoznávání textu z obrazu v reálném čase	47
5	Výsledky rozpoznávání	48
5.1	Jazyky - Asie	48
5.2	Jazyky – Evropa	50
5.3	Shrnutí rozpoznávání.....	54
6	Závěr	55
A	Rozpoznávání dokumentů	57
B	Obsah přiloženého DVD	71

Seznam zkratek

- CAT** - **Computer-Aided Translation (Computer-Assisted Translation)** - počítačem podporovaný překlad
- MT** - **Machine Translation** – strojový překlad
- TTS** - **text-to-speech** – akustická syntéza řeči z textu
- Tezaurus** - Slovník nabízející synonyma či jinak související slova
- OCR** - **optical-character-recognition** – zpracování textu z obrazu do elektronické podoby
- JNI** - **Java Native Interface** - rozhraní umožňující propojit kód běžící na virtuálním stroji Javy s nativními programy a knihovnamy napsanými v jiných jazycích – např. C, C++, Assembler, apod.
- UI** - **User Interface** - Uživatelské rozhraní
- DOM** - **Document Object Model** – standard pro práci s XML dokumenty
- DVM** - **Dalvik Virtual Machine** – virtuální stroj pro mobilní zařízení zpracovávající zkompileovaný Java kód (bytekód)
- ADT** - **Android Development Tools** – rozšíření vývojového prostředí Eclipse o nástroje pro vývoj aplikací pro Android
- SDK** - **Software Development Kit** – sada nástrojů pro vývoj softwaru, která usnadňuje vývoj pro určenou platformu
- XML** - **Extensible Markup Language** – značkový jazyk umožňující strukturování, ukládání a přenášení dat
- XSD** - **XML Schema Definition** – XML schéma, které popisuje strukturu XML dokumentu. Popisuje přípustný obsah dokumentu a umožňuje kontrolovat správnost dat

Seznam obrázků

Obrázek 1 - Životní cyklus aktivity	13
Obrázek 2 - Use Case A - aplikace	16
Obrázek 3 - Use Case B - překlad	17
Obrázek 4 - Use Case C - rozpoznávání	18
Obrázek 5 - UML aktivity	27
Obrázek 6 - UML data překladu	29
Obrázek 7 - UML Dictionary	31
Obrázek 8 - UML OCR	32
Obrázek 9 - XSD schéma pro označení bodů na snímku	35
Obrázek 10 - Snímek pro vizuální syntézu	36
Obrázek 11 - Zastoupení verzí systému	38
Obrázek 12 - Úvodní obrazovka	41
Obrázek 13 - Nastavení	41
Obrázek 14 - Nastavení RT	42
Obrázek 15 - Nastavení jazyku	42
Obrázek 16 - Výběr souboru	42
Obrázek 17 - Ukládání souboru	42
Obrázek 18 - Aktivita	43
Obrázek 19 - Režim překladu	43
Obrázek 20 - Překlad segmentu	44
Obrázek 21 - Překlad segmentu - možnosti /1/	44
Obrázek 22 - Překlad segmentu - možnosti /2/	44
Obrázek 23 - Přeložený text – možnosti	44
Obrázek 24 - Strojový překlad /1/	45
Obrázek 25 - Strojový překlad /2/	45
Obrázek 26 - OCR předloha	46
Obrázek 27 - OCR rozpoznaný text	46
Obrázek 28 - OCR z fotografie /1/	46
Obrázek 29 - OCR z fotografie /2/	46
Obrázek 30 - OCR v reálném čase /1/	47
Obrázek 31 - OCR v reálném čase /2/	47
Obrázek 32 - Japonský nápis /1/	48
Obrázek 33 - Japonský nápis /2/	48
Obrázek 34 - Japonský nápis /3/	48
Obrázek 35 - Čínský nápis /1/	49
Obrázek 36 - Čínský nápis /2/	49
Obrázek 37 - Čínský nápis /3/	49
Obrázek 38 - Čínský nápis /4/	49
Obrázek 39 - Německý nápis /1/	50
Obrázek 40 - Německý nápis /2/	50
Obrázek 41 - Německý dokument /1/	50

Obrázek 42 - Anglický nápis /1,2,3/.....	51
Obrázek 43 - Anglický nápis /4/.....	51
Obrázek 44 - Francouzský nápis /1/.....	51
Obrázek 45 - Francouzský nápis /2/.....	51
Obrázek 46 - Francouzský nápis /3/.....	52
Obrázek 47 - Italský nápis /1/.....	52
Obrázek 48 - Italský nápis /2/.....	52
Obrázek 49 - Italský nápis /3,4/.....	53
Obrázek 50 - Španělský nápis /1/.....	53
Obrázek 51 - Španělský nápis /2,3,4/.....	53
Obrázek 52 - Český dokument /1/.....	54
Obrázek 53 - Počet instalací aplikace.....	55
Obrázek 54 - Japonský dokument.....	57
Obrázek 55 - Čínský dokument.....	59
Obrázek 56 - Korejský dokument.....	61
Obrázek 57 - Arabský dokument.....	62
Obrázek 58 - Německý dokument /2/.....	63
Obrázek 59 - Anglický dokument /1/.....	64
Obrázek 60 - Anglický dokument /2/.....	65
Obrázek 61 - Francouzský dokument.....	66
Obrázek 62 - Italský dokument.....	67
Obrázek 63 - Španělský dokument.....	68
Obrázek 64 - Český dokument /2/.....	69
Obrázek 65 - Ruský dokument.....	70

Seznam tabulek

Tabulka 1 - Verze systému Android.....	14
Tabulka 2 - Příklade využití aplikačních prostředků k lokalizaci aplikace.....	15
Tabulka 3 - Strojový překlad	20
Tabulka 4 - Zastoupení verzí systému	38
Tabulka 5 - Podporované jazyky překladu	39
Tabulka 6 - Podporované jazyky OCR.....	40
Tabulka 7 - Japonský dokument OCR+MT	58
Tabulka 8 - Čínský dokument - OCR+MT.....	60
Tabulka 9 - Korejský dokument - OCR+MT	61
Tabulka 10 - Arabský dokument - OCR+MT	62
Tabulka 11 - Německý dokument /2/ - OCR+MT	63
Tabulka 12 - Anglický dokument /1/ - OCR+MT.....	64
Tabulka 13 - Anglický dokument /2/ - OCR+MT.....	65
Tabulka 14 - Francouzský dokument - OCR+MT	66
Tabulka 15 - Italský dokument - OCR+MT	67
Tabulka 16 - Španělský dokument – OCR+MT.....	68
Tabulka 17 - Český dokument /2/ - OCR.....	69
Tabulka 18 - Ruský dokument - OCR+MT.....	70

1 Úvod

CAT nástroje se odlišují od běžných strojových překladačů přístupem k překládanému textu. Zatímco běžný překladový nástroj nabídne pouze automatický překlad celého textu (např. aplikace „Překladač Google“), CAT nástroje rozdělí text na menší logické části a kromě automatického překladu nabízí pokročilé vlastnosti pro zkvalitnění překladu – tezaury, překladové paměti, terminologické databáze apod. MT nástroje se snaží úplně nahradit lidského překladatele, CAT nástroje usnadňují nebo zkvalitňují jeho práci, přičemž MT využívají jako jeden z dostupných nástrojů při práci se segmenty. Výsledný překlad závisí na překladateli, nicméně lze tvrdit, že kvalita překladu při použití CAT nástroje je vyšší než při použití MT, nebo bez pomocných nástrojů.

Další důležitá odlišnost je podpora formátů textu k překladu – CAT nástroje podporují širokou škálu formátů a tedy mimo běžný text slouží k překladu dalších typů textu – např. titulků, lokalizací aplikací či zdrojových kódů. Výstupní soubor si uchovává svoji strukturu, překládá se pouze to, co je určeno k překladu.

Na platformě Android existuje velké množství strojových překladačů, nikoliv však nástrojů pro sofistikovanější asistovaný překlad, v současnosti nejsou tyto nástroje, na rozdíl od platformy Windows, dostupné. Tento stav není dle mého názoru optimální, a proto se dané problematice věnuji ve své práci.

Cílem této práce je navržení a implementace aplikace pro asistované překládání textu (CAT) na platformě Android. Tato práce si klade za cíl vlastní implementaci CAT nástroje, nicméně další možností by mohlo být portování a úprava existující aplikace (OmegaT - Java, OpenSource) na platformu Android. OmegaT je typickým zástupcem desktopového CAT nástroje a jako takový může sloužit pro porovnání dosažené funkcionality (s určitými omezeními danými platformou). Součástí aplikace bude i rozpoznávání textu z obrázku a jeho následný překlad a syntéza řeči (také systém audio-vizuální syntézy řeči).

1.1 Požadované vlastnosti vyvíjené aplikace

- Překlad textu po menších částech – *překladových jednotkách*
- Strojový překlad
- Tezaurus
- Podpora textových formátů souborů
- Načtení textu z obrázku – OCR textu
- TTS

2 Implementace

2.1 Specifika vývoje aplikací pro chytré telefony

Při vytváření aplikací pro PDA zařízení či telefony je potřeba respektovat výhody a nevýhody plynoucí z méně výkonného hardwaru nacházejícího se v těchto zařízeních. (1)

Konkrétně je třeba brát zřetel na:

- Velikost a rozlišení displeje
- Absenci fyzické klávesnice u většiny zařízení
- Nepřesnost polohovacích zařízení (prst oproti myši), stylus není příliš obvyklý
- Rychlost procesoru a velikost operační paměti
 - Rozdíly oproti počítačům
 - Rozdíly mezi jednotlivými zařízeními
- Podpora programovacího jazyku

2.2 Aplikace pro operační systém Android

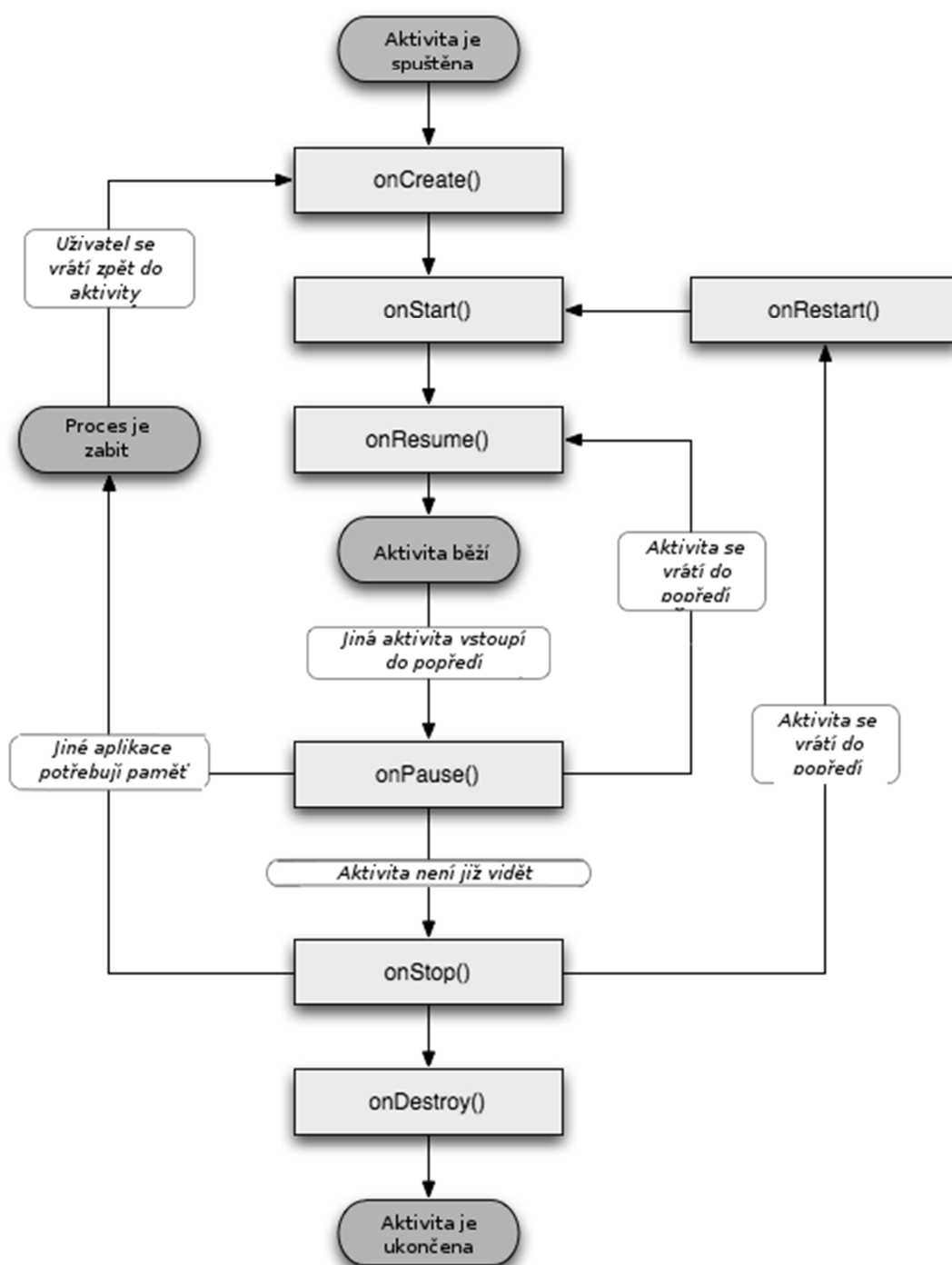
Operační systém Android je založen na linuxovém jádře, knihovnách v jazycích C a C++, které zahrnují např. přehrávání audia a videa, relační databázi, vykreslování fontů a 3D grafiky či prohlížeč webových stránek, a virtuálním stroji Dalvik (DVM). Dalvik má na starosti spouštění a běh aplikací, které jsou obvykle psány v jazyce Java. DVM je v podstatě revidovaný JVM, optimalizovaný pro stroje s menším výkonem, pamětí a úložným prostorem oproti dnešním počítačům. Místo Java byte code jsou použity .dex soubory, a výsledná nekomprimovaná velikost oproti .jar souborům je poloviční. Také počet vykonávaných instrukcí je nižší až o 30 procent. (2)

Pro vývoj aplikací je oficiálně podporováno vývojové prostředí Eclipse ve spojení s ADT pluginem. Další důležitou částí je SDK – obsahuje užitečné nástroje pro návrh i testování aplikace. Součástí SDK je emulátor platformy, jenž umožňuje testování aplikace na různých verzích a konfiguracích operačního systému, dále dokumentace, příklady zdrojových kódů, Google API a další.

2.2.1 Aktivity a služby

Stavební prvky aplikace pro operační systém Android se nazývají aktivity (activity). Každá aktivita představuje obrazovku aplikace, odpovídá nějaké činnosti, kterou aplikace poskytuje. Každá aplikace má jednu vstupní aktivitu – toto nastavení je v souboru `AndroidManifest.xml`. Mimo aktivit jsou důležité služby (service). Služby prezentují proces běžící na pozadí, který je k dispozici pro aktivity aplikace.

Životní cyklus aktivity je vymezen základními funkcemi, viz Obrázek 1 - Životní cyklus aktivity. Tyto funkce odpovídají situacím, které mohou s aktivitou nastat od okamžiku jejího spuštění. (3)



Obrázek 1 - Životní cyklus aktivity

- `onCreate()` – Vytvoření aktivity. Zde běžně dochází k inicializaci aktivity a nastavuje se obsah UI.
- `onStart()` – Aktivita viditelná pro uživatele.
- `onResume()` – Aktivita získala focus.
- `onPause()` – Těsně před ztrátou focusu. Po této funkci již není zaručen běh aktivity, pokud se operačnímu systému nedostává prostředků.
- `onStop()` – Aktivita přestává být viditelná pro uživatele.

- `onRestart()` – Těsně předtím než aktivita, která byla zastavena (byla na ní zavolána funkce `onStop()`), je znovu nastartována - ve funkci `onStart()`.
- `onDestroy()` – Těsně před zrušením instance aktivity. Poté již aktivita nemůže vykonávat žádnou činnost.

2.2.2 Dodavatelé obsahu a záměry

Dodavatelé obsahu zajišťují úroveň abstrakce jakýchkoliv dat uložených v zařízení, která jsou přístupná více různým aplikacím, a současně poskytují úplnou kontrolu nad způsobem přístupu k těmto datům. (1)

Záměry jsou v podstatě systémové zprávy, zpřístupňující aplikacím aktuální dění (např. upozornění na příchozí hovor), a umožňují reagovat na aktuální dění nebo slouží k sdělení záměru aplikace, který může vyžadovat obsluhu z jiné aktivity. Příkladem je odeslání textové zprávy – aktivita říká, jakou zprávu chce odeslat. Obsloužit tento požadavek může Gmail, SMS, Skype, Bluetooth, sociální síť a další zaregistrované aktivity - tím pádem se obsluha liší dle nainstalovaných aplikací v konkrétním zařízení, nicméně aktivita pouze vyvolá záměr (intent) a obsluhu požadavku, pokud je alespoň jedna aktivita zaregistrovaná jako obslužná, zařídí zvolená aktivita.

2.2.3 Konfigurace aplikace

Soubor `AndroidManifest.xml` obsahuje důležitá nastavení aplikace – mimo jiné informace o jednotlivých aktivitách, službách, zpracovávaných záměrech a dodavatelích obsahu, zvoleném tématu vzhledu, ikoně aplikace, jejím názvu a oprávněních, které aplikace vyžaduje pro svůj provoz (potvrzuje uživatel při instalaci aplikace).

V tomto souboru se nachází také údaje o minimální verzi systému nutné pro instalaci a spuštění aplikace `<uses-sdk>` a verzi samotné aplikace.

Tabulka 1 - Verze systému Android

Verze	API	Kód verze
Android 4.4	19	KITKAT
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2.x	17	JELLY_BEAN_MR1
Android 4.1.x	16	JELLY_BEAN
Android 4.0.3-4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0.0-4.0.2	14	ICE_CREAM_SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.3-2.3.4	10	GINGERBREAD_MR1
Android 2.3.0-2.3.2	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR

Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

Relevantní novinky představené v jednotlivých verzích jsou:

- 1.x – Vyhledávání hlasem, gesta, syntéza řeči, nahrávání videa
- 2.x – Digitální zoom, podpora prosvětlovací diody, více nastavení fotoaparátu a kamery, podpora různých velikostí a rozšíření, instalace aplikace na paměťovou kartu
- 3.x – Optimalizace pro tablety, lepší multitasking, USB host
- 4.x – Vylepšené rozpoznání hlasu, rozpoznávání hlasu offline

2.2.4 Aplikační prostředky

Prostředky (resources – složka res/) mohou obsahovat obrázky, ikony, lokalizační řetězce, rozložení obrazovek, další xml soubory atd. S prostředky aplikace souvisí vygenerovaná třída R.java, kde jsou seskupeny identifikátory. S její pomocí lze k prostředkům přistupovat z kódu nebo na ně odkazovat v definicích rozložení GUI (layout). (4)

- res/layout/ - definice uživatelského prostředí aplikace - rozložení obrazovek pro různé orientace a velikosti displeje
- res/drawable/ - statické ikony a obrázky
- res/values/ - pojmenované řetězce, pole, barvy a rozměry – oddělení od kódu aplikace
- res/raw/ – nezpracovávaná „surová“ data v libovolném formátu – libovolné soubory
- res/xml/ - neměnné xml soubory s daty či strukturami

Tabulka 2 - Příklade využití aplikačních prostředků k lokalizaci aplikace

Angličtina	Čeština
<string name="btn_next"> Next </string>	<string name="btn_next"> Následující </string>
<string name="btn_previous"> Previous </string>	<string name="btn_previous"> Předchozí </string>
<string name="btn_translate"> Translate </string>	<string name="btn_translate"> Přeložit </string>

2.3 Vlastní implementace

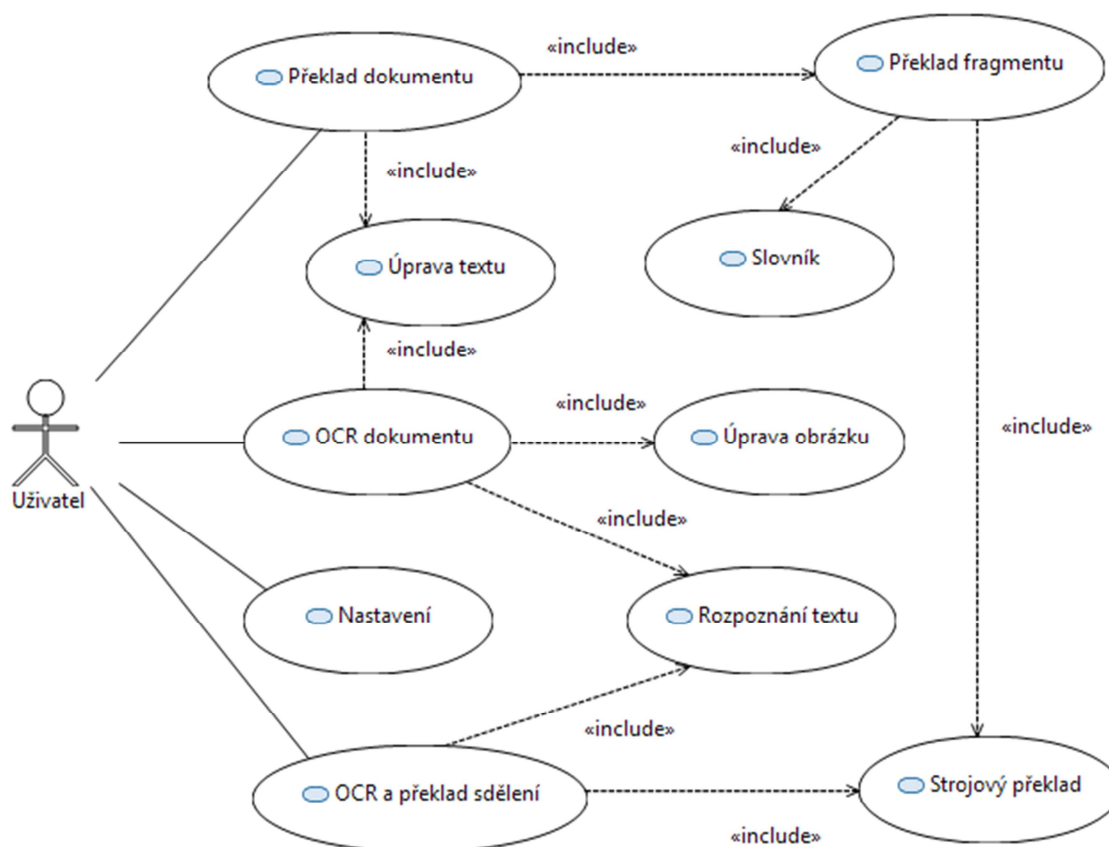
Jako model vývoje byl zvolen iterativní cyklus s přírůstký, který umožnil průběžné rozšiřování aplikace po menších částech dle poznatků získaných při jejím vývoji. (5) Popis vlastní implementace aplikace vychází z výsledného stavu dosaženého iteracími cykly.

Požadovaná oprávnění potřebná pro tuto aplikaci jsou:

- android.permission.CAMERA
- android.permission.ACCESS_NETWORK_STATE
- android.permission.INTERNET
- android.permission.READ_EXTERNAL_STORAGE
- android.permission.WRITE_EXTERNAL_STORAGE

2.3.1 Use Case diagramy

Na následujících Use Case diagramech je nastíněn rámec funkcionality aplikace z pohledu uživatele.



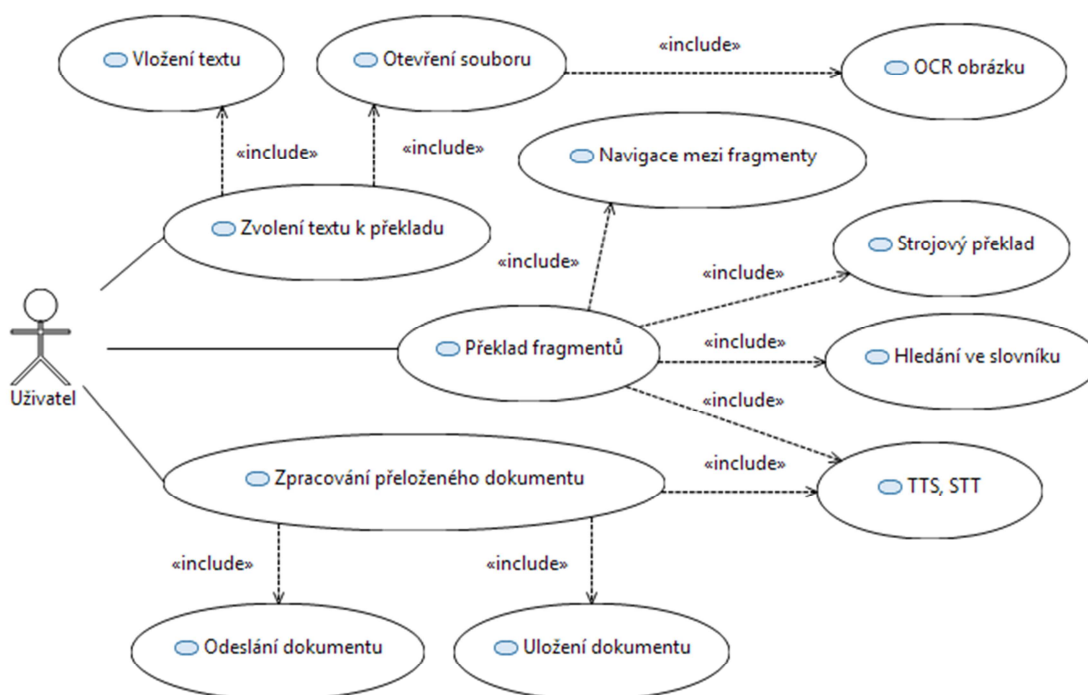
Obrázek 2 - Use Case A - aplikace

Obrázek 2 - Use Case A - aplikace zachycuje zjednodušeně možnosti použití aplikace.

Překlad dokumentu je případem užití, kdy uživatel má v textové podobě překládaný text a kritériem je buď rychlost překladu (strojový překlad), umožňující získat rychlý přehled o obsahu překládaného textu, či kvalita překladu, která je zajištěna postupným překladem jednotlivých fragmentů textu s asistencí dostupných nástrojů (např. slovník). Tento případ je detailněji zde: Obrázek 3 - Use Case B - překlad.

OCR dokumentu je scénářem užití, kdy uživatel nemá k dispozici textovou reprezentaci překládaného textu, pouze jeho vizuální podobu ve formě obrázku (fotografie, sken, snímek obrazovky). Zde dochází k úpravám obrázku (některé úpravy mají vliv na výsledek rozpoznávání) a samotnému rozpoznávání textu z obrázku. Rozpoznávaný text je možné ručně zkorigovat s pomocí původního obrázku a posléze dále využít např. jako textovou podobu pro překlad dokumentu popsáno v předchozím odstavci.

OCR a překlad sdělení je případem užití, kdy uživatel využívá aktuálně snímaný obraz z fotoaparátu a cílem je rychlé rozpoznávání textu z obrázku a jeho strojový překlad. Běžně se jedná o informační cedule, dopravní značení a krátké nápisy. V tomto případě nejde o samotnou kvalitu překladu ale primárně o rychlost, s jakou je toto krátké sdělení zpracováno.



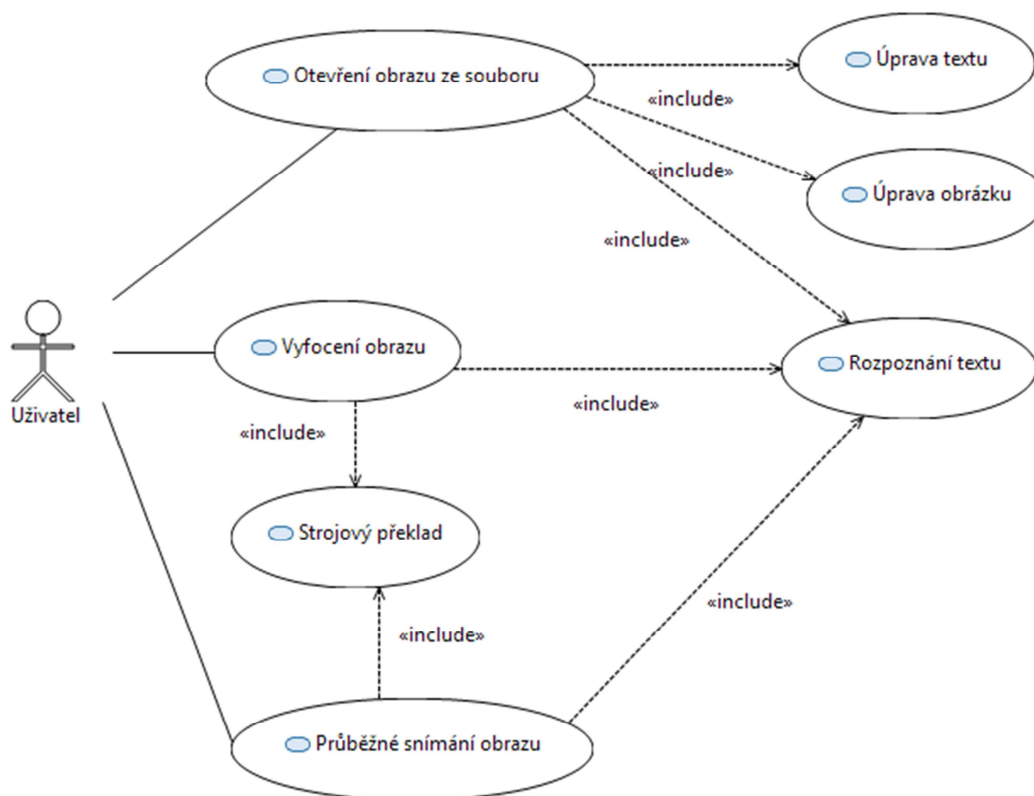
Obrázek 3 - Use Case B - překlad

Obrázek 3 - Use Case B - překlad zachycuje podrobněji případ překladu dokumentu.

Překlad dokumentu je složen ze třech na sebe navazujících činností. Nejprve jde o zvolení textu k překladu – možnosti zahrnují otevření ze souboru, vložení ze schránky, ruční zadání na klávesnici (včetně hlasového zadávání) či automatické rozpoznání z obrázku. Zde také dochází k výběru režimu překladu. Tato volba ovlivňuje další činnost – překlad fragmentů.

Překlad fragmentů se skládá nejprve z rozdělení textu k překladu na části dle zvoleného režimu překladu, posléze z práce se samotnými fragmenty. Jako příklad fragmentu lze použít jednu větu. Po přeložení všech (či pouze některých) fragmentů dojde ke složení přeloženého dokumentu.

Zpracování přeloženého dokumentu zahrnuje práci s přeloženým textem. Takto získaný text lze ještě upravit a posléze uložit či odeslat.



Obrázek 4 - Use Case C - rozpoznávání

Obrázek 4 - Use Case C - rozpoznávání zachycuje podrobněji případy využívající OCR.

Případ OCR dokumentu využívá otevření obrazu ze souboru, případ OCR a překlad sdělení využívá možnosti vyfocení obrazu nebo jeho průběžného snímání. Odlišnost mezi těmito případy byla již vysvětlena.

2.3.2 Realizace klíčových částí aplikace

Samotná realizace výše popsané funkcionality zahrnovala tyto části:

- Překladové jádro
- Strojový překlad
- Rozpoznávání textu z obrazu
- Rozhraní pro slovníky a thesaury
- Hlasová syntéza
- Práce se soubory
- Nastavení aplikace

2.3.3 Překladové jádro

Překladové jádro má na starosti rozdělení textu k překladu na fragmenty dle zvoleného režimu, udržování stavu fragmentů a jejich složení do přeloženého dokumentu.

Jednotlivé režimy jsou prostý text, titulky (.srt), lokalizační soubory (.ini) a strojový překlad. Při strojovém překladu není rozdělení nutné, celý text se jako jeden fragment přeloží a obsahuje tedy přeložený dokument. Režimy titulky a lokalizační soubory využívají k rozdělení na fragmenty specifikaci formátu – tedy u titulků je fragmentem každý titulek a u lokalizačního souboru je fragmentem překladový pár.

Nejsložitější je dělení u prostého textu. Komplikací je to, že přirozené dělení na věty ohraničené tečkou, vykřičníkem a otazníkem nelze použít (resp. lze ale výsledné rozdělení má problém např. s uvozovkami, trojtečkou (výpustkou), titulem a datem mimo jiné). Zde je využít `BreakIterator` – metoda `getSentenceInstance()` – toto rozdělení je daleko přesnější ve srovnání s přirozeným dělením interpunkčními znaménky na konci vět.

2.3.4 Strojový překlad

Strojový překlad (machine translation) je automatický překlad mezi dvěma jazyky pomocí počítačového systému. Takto přeložený text jde většinou snadno rozlišit od textu přeloženého člověkem, jelikož výstup není dokonalý a proto je vhodný pouze pro orientační překlad textu, nicméně pro rychlý náhled na význam textu je dostačující. Kvalita strojového překladu se liší nejen dle použitého překladače, ale závisí i na jazycích, mezi kterými je překládáno, některé dvojice jsou převáděny přes třetí jazyk, což snižuje kvalitu překladu. Dále má také vliv složitost jazyků, mezi kterými je překládáno. Komplikací je odlišná struktura jazyků, jako příklad lze uvažovat třeba pravidla změn tvarů slov v češtině a v angličtině.

Pro ilustraci je přiložen odstavec textu přeloženého pomocí překladačů (Google Translate a Bing Translate v tomto pořadí) z angličtiny do češtiny.

Tabulka 3 - Strojový překlad

On a basic level, MT performs simple substitution of words in one natural language for words in another, but that alone usually cannot produce a good translation of a text because recognition of whole phrases and their closest counterparts in the target language is needed. Solving this problem with corpus and statistical techniques is a rapidly growing field that is leading to better translations, handling differences in linguistic typology, translation of idioms, and the isolation of anomalies.

Na základní úrovni, MT vykonává jednoduchou náhradu slov v jednom přirozeném jazyce pro slova v jiném, ale to samo o sobě obvykle nemůže produkovat dobrý překlad textu, protože je potřeba uznání celých vět a jejich nejbližšími protějšky v cílovém jazyce. Řešení tohoto problému s korpusem a statistických technik je rychle se rozvíjející obor, který vede k lepším překladům, zacházení s rozdíly v lingvistické typologii, překladu idiomů a izolace anomálií.

Na základní úrovni MT provádí jednoduché nahrazení slov v jednom přirozeném jazyce slova v jiném, ale že sám obvykle nemůže produkovat dobrý překlad textu protože je potřeba uznání celých vět a jejich nejbližší protějšky v cílovém jazyce. Řešení tohoto problému s corpus a statistických technik je rychle rostoucí pole, které vede k lepším překladům, manipulace s rozdíly v jazykové typologii, překladu idiomů a izolace anomálií.

Kritérii při výběru konkrétní služby byla dostupnost volně využitelného API rozhraní, kvalita překladu a možnost bezplatného překladu. Tyto podmínky bohužel nesplňuje Google Translate API, kde došlo v roce 2011 ke změně podmínek a bezplatné využívání již není možné. Zvolen byl Microsoft Translator API (Bing Translator), který nabízí měsíčně bezplatně překlad dvou milionů znaků (počítáno z překládaného textu) v kvalitě srovnatelné s řešením od Googlu za pomoci java wrapperu microsoft-translator-java-api. Jako alternativy lze zmínit Yandex Translate API, Yahoo Babelfish a MyMemory translation service. Z publikovaného článku srovnávajícího kvalitu překladu pro různé délky textu a kombinace jazyků vyplývá, že v kratších textech (pod 2000 znaků) dosahuje lepších výsledků řešení od Microsoftu, což vyhovuje předpokládanému použití na orientační překlad krátkých fragmentů textu.

2.3.5 Rozpoznávání textu z obrazu

Rozpoznávání textu z obrazu je klasickým problémem počítačového vidění. První částí je snímek předlohy, druhou částí je rozpoznání textu obsaženého na předloze. Výsledek je ovlivněn jak kvalitou předlohy, tak použitým OCR řešením. Obecně lze říci, že kde člověk při přepisu písma v pro něj známém jazyce z předlohy dělá minimum chyb i v případě horší kvality předlohy, tam je počet chyb při OCR značně závislý na kvalitě předlohy a je nutná kontrola výsledků a jejich případná oprava.

Z hlediska výběru OCR pro použití na platformě Android můžeme rozdělit dostupná řešení na typy „offline knihovna“ a „online služba“. Online služby

(<http://ocrapiservice.com>, ABBYY Cloud OCR, <http://www.ocr-it.com>) fungují na principu odeslání snímku předlohy na server služby, kde dojde k rozpoznání textu, načež je rozpoznáný text odeslán zpět na zařízení, přičemž obecně tyto služby nabízí jisté množství rozpoznání snímků zdarma (50-100) a placené balíčky nad toto množství. Offline knihovny (Tesseract OCR Engine, ABBYY Mobile OCR Engine, ocr4j) fungují na principu lokálního rozpoznání snímku přímo na zařízení. Za výhody online služeb lze považovat nižší paměťovou a procesorovou zátěž zařízení a jednodušší implementaci, naopak nevýhodná se jeví nutnost odesílat relativně velké množství dat (zaleží na velikosti snímku), což na rozdíl od „pevného internetu“ je u mobilního připojení stále nevýhodou vzhledem k jeho specifikům (FUP, rychlost dle pokrytí), a také omezení ohledně počtu rozpoznání v bezplatném režimu.

Zvolen byl Tesseract OCR Engine vzhledem k tomu, že oproti řešení od ABBYY se jedná o open source a zároveň v této kategorii poskytuje nejpřesnější výsledky. V současnosti je aktuální verze 3 s podporou několika desítek jazyků (včetně podpory jazyků jdoucích zprava doleva) a možností natrénovat další jazyky. Vzhledem k tomu, že jde o knihovnu napsanou v programovacích jazycích C a C++, je nutné využívat JNI rozhraní, které umožňuje propojit kód v jazyce Java s nativními knihovnami (C, C++, Assembler atd.).

2.3.6 Rozhraní pro slovníky a tezaury

Slovníky a tezaury slouží ke zkvalitnění překladu. Překladové slovníky obsahují překlad slov a frází mezi jazyky, slovníky výkladové definují a vysvětlují jednotlivá hesla, tezaury poskytují synonyma, antonyma a občas i možné verše.

Také v této oblasti jsou k dispozici online i offline řešení, nicméně vzhledem k malému objemu přenášených dat a malému množství dostupných kvalitních offline slovníků (převážně StarDict slovníky) nejsou tyto využity.

Prvním zvoleným řešením je DICT4J – klient pro přístup k serverům dle protokolu RFC 2229, konkrétně k dict.org – pomocí nějž je zpřístupněna široká škála slovníků. Druhým zvoleným řešením je vlastní implementace pomocí extrakce dat z online služeb (slovník.cz, slovník.seznam.cz, lekarske.slovníky.cz a ssjc.ujc.cas.cz) anebo jejich API (Big Huge Thesaurus).

2.3.7 Hlasová syntéza

Hlasová syntéza (ve smyslu TTS) převádí textovou reprezentaci jazyka na akustickou, je to umělá tvorba lidské řeči (v podstatě předčítání textu). Možnosti využití jsou široké, ale v kontextu CAT nástroje jde hlavně o možnosti poslechnout si překládaný text, což může pomoci při překladu, poslechnout si přeložený text, což může napomoci odhalit chyby v překladu nebo zvláštní formulace, anebo nechat automaticky přeložený text v jazyce, kterým uživatel plynne nevládne, přečíst a použít jako dorozumívací prostředek.

TTS na platformě android je součástí API od verze 4 (Android 1.6 – DONUT). Tento přístup umožňuje využít libovolný nainstalovaný hlasový syntetizátor kompatibilní s tímto API. Podpora jazyků taktéž závisí na nainstalovaném syntetizátoru – základní engine dodávaný jako součást systémů Android (záleží na výrobcí zařízení/ROM) má ve výchozím nastavení jazyková data pro pět světových jazyků (angličtina, francouzština, němčina, italština a španělština). Jako další lze zmínit např. syntetizátory SVOX, Accapela a eSpeak, které se liší hlavně v nabízených jazycích (desítky dostupných jazyků) – kvalita je srovnatelná, záleží do značné míry na subjektivním posouzení produkované řeči.

Dalším rozšířením hlasové syntézy je tzv. mluvící hlava – jedná se o kombinaci sluchových a zrakových vjemů – systém audio-vizuální syntézy řeči.

2.3.8 Práce se soubory

Práce se soubory zahrnuje výběr, otevření, načtení a uložení souboru v paměti zařízení. Obecná manipulace se soubory se dá shrnout jako činnosti správce souborů a načtení souborů zahrnuje navíc řešení problémů s kódováním souborů. Význam manipulace se soubory spočívá v možnosti načíst překládaný text ze souboru a přeložený text do souboru opět uložit.

Místo vlastní implementace správce souborů je využita užitečná vlastnost platformy Android, kdy aplikace může sdělit systému záměr (intent) provést operaci, a pokud je v systému zaregistrovaná aplikace schopná tento záměr vykonat, tak je operace provedena jejím prostřednictvím. Konkrétně je použit správce souborů AndExplorer, který vykonává záměr ACTION_PICK – výběr souboru.

Formát načtených souborů je detekován podle jejich přípony, přičemž textové formáty, u kterých se může lišit kódování, jsou zpracovávány pomocí cpdetectoru (CodePageDetectorProxy). Tento framework provádí detekci kódování textu strategií první kódování, které odpovídá, je správné, tím pádem se nejedná o stoprocentně spolehlivou metodu, ale alternativou je pouze ruční výběr kódování, což není přívětivější řešení pro uživatele.

2.3.9 Nastavení aplikace

Nastavení aplikace slouží k upravení chování aplikace dle potřeb a preferencí uživatele. Pro shodné chování a vzhled nastavení aplikace (v rámci platformy Android) je použit přístup umožňující definovat nastavení aplikace pomocí voleb, které jsou k dispozici, a o samotné vykreslení se postará operační systém.

Nastavení je společné pro celou aplikaci a zahrnuje volby jazyka, formát ukládání textových souborů, výběr používaného slovníku a detailní nastavení OCR při režimu průběžného snímání.

2.4 Aktivity aplikace

Obrazovky aplikace jsou následující (balíček `eu.mshot.translatorassistant`):

- Rozcestník (`TranslatorHomeActivity.java`)
 - Výběr režimu aplikace
 - Návrat k předchozí činnosti
- Výběrová (`TranslatorAssistantActivity.java`)
 - Zadání a úprava textu k překladu
 - Otevření ze souboru
 - Volba typu překladu
 - Uložení textu k překladu
 - Obnovení aplikace po ukončení z uloženého stavu
- Překladová (`TranslatorProcessActivity.java`)
 - Překlad jednotlivých překladových jednotek
 - Navigace mezi překladovými jednotkami
 - Strojový překlad – Bing Translate!
- Výsledková (`TranslatorResultActivity.java`)
 - Zobrazení výsledného přeloženého textu
 - Uložení překladu do souboru
 - Odeslání překladu

Tyto aktivity jsou jádrem aplikace – v tomto pořadí navazují obvykle činnosti, při překladu textu, na sebe.

Další obrazovky aplikace:

- Rozpoznávací (`TesseractOCRActivity.java`)
 - Načtení a zobrazení obrázku
 - Úpravy obrázku pro lepší výsledky rozpoznávání
 - OCR – Tesseract
 - Úpravy/opravy rozpoznávaného textu
- Slovníková (`TranslatorThesaurusActivity.java`)
 - Vložení hledaného slova
 - Získání/zobrazení výsledků z tezauru
- Nastavení (`Preferences.java`)
 - Jazyk zdroje
 - Jazyk cíle
 - Jazyk pro OCR

- Konec řádku
- Další nastavení

Aktivita Rozcestník je vstupním místem aplikace a jako taková se stará o obnovu aplikace na poslední známé pozici pro zachování pozitivní uživatelské zkušenosti.

Rozpoznávání je dostupné z úvodní aktivity jako alternativní metoda pro získání textu k překladu.

Slovník (tezaurus) je dostupný z překladové aktivity a slouží k zlepšení kvality překladu širší slovní zásobou.

Nastavení je globálně dostupné ze všech aktivit, pro které je relevantní (tedy všech kromě slovníku). Dostupné jazyky v nastavení zdroje a cíle odpovídají podporovaným jazykům strojového překladu – konkrétně Bing Translate! API, jazyky pro OCR potom odpovídají dostupným natrénovaným datům pro tesseract-ocr verze 3.

2.5 Služby aplikace

- Stahování natrénovaných dat pro OCR (`DownloadService.java`)
 - Navázáno na změnu jazyka pro OCR v nastavení
 - Ukazatel průběhu

Tato služba umožňuje stahování dat pro OCR na vyžádání (vyvolané změnou nastavení jazyka) – výhodou tohoto řešení je úspora místa, nevýhodou je nutnost internetového připojení při prvním použití nového jazyka. Pro ilustraci velikosti natrénovaných dat – čeština má 1MB, tradiční čínština 23.8MB, většina jazyků se pohybuje velikostí v dolní půli tohoto rozsahu.

2.6 Použité knihovny / komponenty

„Virtuální stroj Dalvik, který je jádrem systému Android, není přesně vzato Java prostředí a komponenty, které dodává ve svém SDK, nejsou přesně stejné jako v tradičním Java SDK. Přesto však množství Java knihoven třetích stran nabízí schopnosti, kterými Android nativně nedisponuje, a proto mohou být pro váš projekt užitečné.“ (1)

2.6.1 tesseract-android-tools

Nástroje Tesseract pro Android poskytují sadu API a obsahují soubory pro sestavení knihoven pro zpracování obrazu Tesseract OCR a Leptonica.

Pomocí JNI je aplikaci zpřístupněna knihovna tesseract-ocr (C, C++). Jedná se o knihovnu sloužící k rozpoznání textu z obrazu vyvíjenou společností HP (později Google) a uvolněnou pod open source licencí v roce 2005. Tato knihovna nabízí kvalitní převod do textu a podporuje okolo 40 jazyků.

V současné podobě (verze 3.xx) jsou aplikaci dostupné tato natrénovaná data: Afrikánština, Albánština, Arabština, Ázerbájdžánština, Baskičtina, Běloruština,

Bengálština, Bulharština, Katalánština, Chorvatština, Čeština, Dánština, Holandština, Angličtina, Esperanto, Estonština, Finština, Franština, Francouzština, Galícijština (Haličština), Němčina, Řečtina, Hebrejština, Hindština, Maďarština, Čerokézština, Zjednodušená čínština, Tradiční čínština, Indonéština, Italština, Japonština, Islandština, Kannadština (Kannarština), Korejština, Litevština, Lotyšština, Makedonština, Malajština, Malajalámština, Maltština, Norština, Polština, Portugalština, Rumunština, Ruština, Srbština, Slovenština, Slovinština, Španělština, Svahilština, Švédština, Tagalština (Tagalog), Tamilština, Telugština, Turečtina, Thajština, Ukrajinština, Vietnamština.

Pro další jazyky existuje možnost vlastního natrénování.

Dostupná pod licencí Apache License 2.0.

2.6.2 microsoft-translator-java-api

Tato knihovna zpřístupňuje Microsoft Translator API strojového překladače Bing Translator pomocí obalové třídy v jazyce Java. Jedná se o obdobnou službu jako Google Translate, její výhodou je, že po zpoplatnění Google Translate zde zůstal bezplatný tarif pro 2000000 znaků za měsíc.

V současnosti podporuje překlad mezi 40 jazyky (včetně češtiny). Kvalita překladu je dostatečná, pouze při srovnání s překladem z Google Translate se může jevit jako mírně horší (nepřesnější) – velmi záleží na překládaném textu, zdrojovém a cílovém jazyce, délce textu a dalších faktorech – více informací k této problematice lze nalézt např. v článku na Tcworld.info (6).

Dostupná pod licencí Apache License 2.0.

Alternativou by mohl být yandex-translator-java-api (Yandex online service for machine translation) s podporou cca 30 jazyků.

2.6.3 BigHugeThesaurus

Tato knihovna zpřístupňuje rozhraní slovníku synonym, antonym a rýmů (pouze angličtina). BigHugeThesaurus nabízí bezplatný tarif pro 10000 slov za měsíc, pro hledaná slova využívá data z Princetonské univerzity (WordNet database), Carnegie Mellon univerzity (Pronouncing Dictionary) a návrhy od uživatelů (pouze nevhodné výrazy jsou blokovány).

Jedná se o vlastní implementaci.

2.6.4 cpdetector

Název cpdetector je zkratkou z code page - detector. cpdetector je knihovnou pro konfigurovatelnou detekci kódování dokumentů. Lze ji využívat i pro detekci kódování dokumentů z vzdáleného úložiště. Tato knihovna nalézá uplatnění všude tam, kde není známo použité kódování, což je nutná informace pro využití informací z dokumentů.

Knihovna pro detekci kódování textu je potřebná při načítání dat ze souboru – v případě, že aplikace nezná kódování dat a zvolí špatné, potom místo očekávaného textu získá text poškozený. Právě tomuto problému cpdetector předchází.

Dostupná pod licencí Mozilla Public License 1.1 (MPL 1.1).

2.6.5 ZoomableImageView

Komponenta pro zobrazení obrázků s možnostmi přibližování a posunu. Aplikace ji využívá v části pro rozpoznání textu.

Dostupná pod licencí Apache License 2.0.

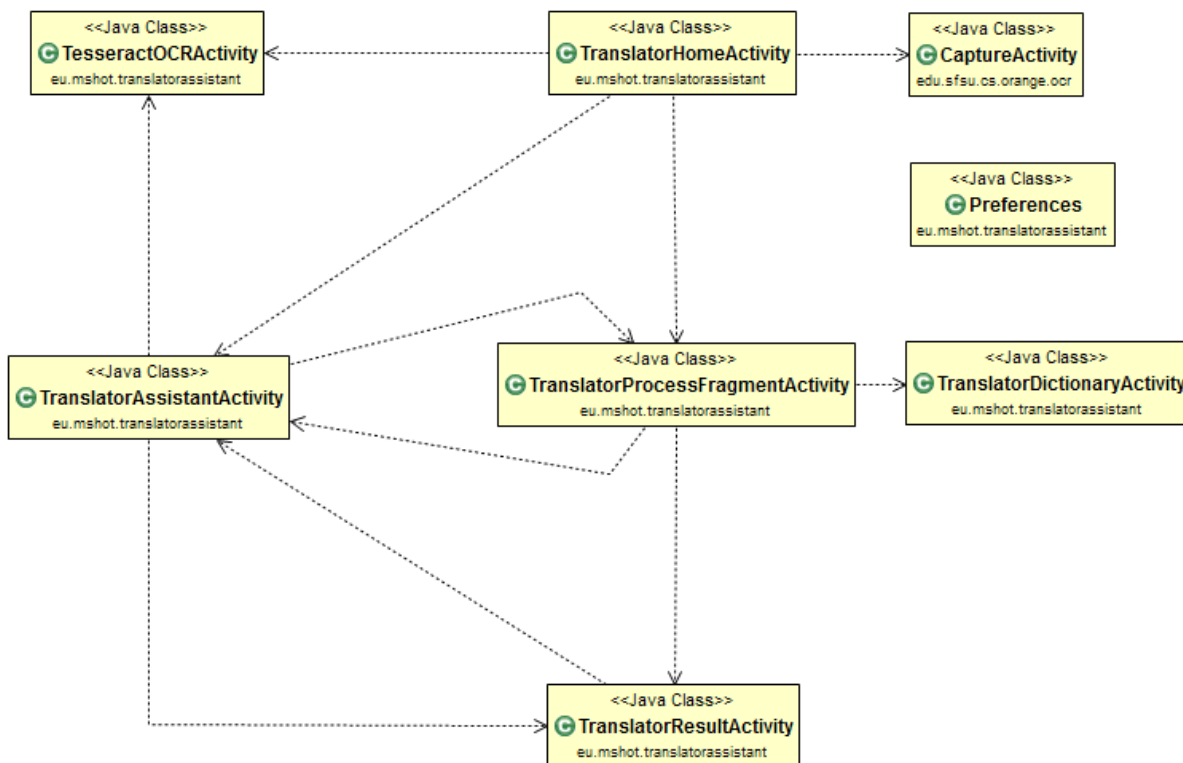
2.6.6 Andexplorer

Andexplorer je jednoduchý správce souborů, který nabízí „intent“ (aktivitu) pro výběr souboru z úložiště zařízení dalším aplikacím. Kromě procházení souborů umožňuje řadit soubory podle jména, velikosti a data, dále nabízí kopírování, vkládání, přejmenování a mazání souborů, vytváření složek, kompresi a dekompresi některých typů archivů či odesílání souborů emailem. Výhodou je, že nepožaduje jiná oprávnění, než ta pro práci se soubory.

Jedná se o Freeware aplikaci.

2.7 Popis implementace

Na diagramu (Obrázek 5 - UML aktivity) jsou zachyceny třídy aktivit aplikace a závislosti mezi nimi.



Obrázek 5 - UML aktivity

2.7.1 TranslatorHomeActivity

Obsahuje metody pro obsluhu voleb z rozcestníku (Obrázek 12 - Úvodní obrazovka) a změnu nastavení aplikace. Tyto jsou realizovány pomocí jednotlivých záměrů (Intent), které realizují zvolenou činnost.

2.7.2 TranslatorAssistantActivity

Atributy `OPEN_FILE_REQUEST`, `SAVE_FILE_REQUEST`, `RETURN_OCR_TEXT` slouží k obsluze vrácených výsledků aktivit v metodě `onActivityResult`, `sharedPref` slouží k přístupu ke sdíleným preferencím aplikace ve formě klíč-hodnota, `dialog` umožňuje zobrazení informace o průběhu akce.

Metoda `onCreate` inicializuje aktivitu, přiřazuje rozložení GUI z prostředků aplikace a inicializuje detektor kódové stránky, `onCreateOptionsMenu` inicializuje menu aplikace (taktéž z aplikačních prostředků).

V metodě `onOptionsItemSelected` je obsluha voleb z menu – načtení a uložení souboru, získání textu ze schránky, vyvolání nastavení aplikace a možnost OCR rozpoznávání. Pro část načtení a uložení souboru využívá aplikace aktivitu `Andexploreru` ve volání aktivitu pro výsledek např. `startActivityForResult(Intent.ACTION_PICK,`

OPEN_FILE_REQUEST) – tato část se stará o výběr souboru a samotné vyhodnocení probíhá až v metodě `onActivityResult`. Rozpoznávání je samostatná aktivita volaná obdobným způsobem pro výsledek. Nastavení má vlastní aktivitu `Preferences`. K přístupu do schránky slouží systémová služba - `getSystemService(CLIPBOARD_SERVICE)`.

Metoda `onActivityResult` reaguje na vrácení výsledku z aktivit pro něj volaných. Vrácená data jsou v objektu `Intent` – pro OCR jde o rozpoznávaný text, pro práci se soubory o jejich URI adresy. Při načítání souboru se podle jeho přípony rozhodne, jakou metodou se zpracuje např. pro docx `nacistDocxSoubor`. Tyto metody pro práci se soubory jsou odděleny v třídě `FileToString.java` – konkrétně se jedná o metody pro načtení txt, odt, docx, jpg a bmp. Při ukládání souboru se podle nastavení zvolí zakódování konce řádku (`\n` nebo `\r\n`).

Zachování posledního stavu aplikace mají na starosti metody `onPause`, `onStart` a `onResume` – stavová data aplikace se nachází ve sdílených preferencích aplikace. Umístění v těchto metodách zajišťuje, že i v případě ukončení aplikace operačním systémem bude po opětovném spuštění aplikace ve stejném stavu jako před svým ukončením (pokud uživatel nevymaže data aplikace), včetně přepnutí na poslední aktivní obrazovku – úvod/překlad/výsledek. Získání preferencí je realizováno metodou `getPrefs`, která zároveň řeší při dostupnosti internetového připojení start služby pro stažení natrénovaných dat jazyka nastaveného pro OCR.

Pro obsluhu UI slouží metoda `clear` – obnovení UI do výchozího stavu a metoda `translate` – ověří, zda byl zadán text k překladu, a v tom případě předá zadaný text a zvolený typ překladu překladové aktivitě.

2.7.2.1 *FileToString*

`FileToString` zapouzdřuje metody pro načítání souboru - `nacistOdtSoubor`, `nacistDocxSoubor`, `nacistOCRSoubor` a `nacistTxtSoubor`. Odt a Docx využívají extrakci textových dat z XML souborů pomocí DOM parseru – Celý XML dokument se načte najednou do stromové struktury v paměti (7).

2.7.3 **TranslatorProcessFragmentActivity**

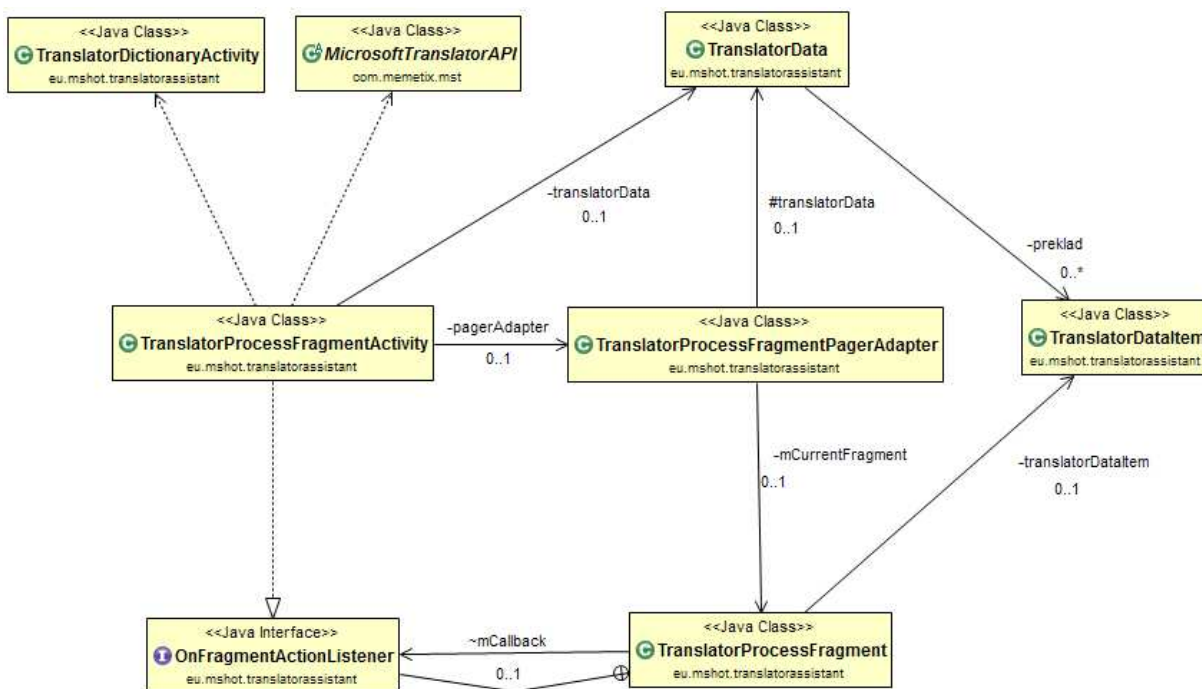
Atribut `translatorData` je datová struktura zaštiťující překládaný text, `sourceLanguagePreference`, `targetLanguagePreference` a `translatedTextBing` jsou určeny pro strojový překlad, `ttsSource` a `ttsTarget` určují, zda je k dispozici TTS pro tyto jazyky zatímco samotný inicializovaný TTS engine je v atributu `textToSpeech` a `mHandler` je atribut pro oddělení strojového překladu do vlastního vlákna. Oddělení do vlastního vlákna je z důvodu přístupu k online obsahu, jelikož v hlavním vlákne aplikace by případné problémy se získáním obsahu vedli k blokování uživatelského rozhraní – aplikace by „zamrzla“. (3)

Metoda `onCreate` inicializuje aktivitu, přiřazuje rozložení GUI z prostředků aplikace a načte data k překladu podle zvoleného typu překladu – `text/srt/ini`. O načtení se stará třída `TranslatorData.java` (viz Obrázek 6 - UML data překladu). Odlišnosti

v načtení spočívají ve struktuře dat. Text – překlad po větách, oddělovačem jsou diakritická znaménka; srt – čas titulku, titulek, oddělovačem je volný řádek; ini – zástupný text, rovnítko, text k překladu, oddělovačem konec řádku, možnost komentáře neurčeného k překladu.

V metodě `onOptionsItemSelected` je obsluha voleb z menu – načtení uloženého stavu aktuální překladové části, posuny aktuální pozice v překladu, TTS překládaného a přeloženého textu, vyvolání nastavení aplikace a spuštění Tezauru.

Zachování posledního stavu aplikace mají na starosti metody `onPause` a `onStart`. Metoda `onDestroy` je určena k vypnutí inicializovaného TTS engine.



Obrázek 6 - UML data překladu

Vzhledem k rozdělení překladu na menší části je implementováno za pomoci tříd `TranslatorProcessFragmentPagerAdapter` a `TranslatorProcessFragment` odpovídající fragmentové rozložení umožňující přecházení mezi jednotlivými částmi plynule za pomoci gest, fragmenty též umožňují zjednodušení UI a snadnější ovládání této části aplikace.

Pro zobrazení, editaci a manipulaci s pozicí překladových částí slouží metody `nastavitPostupPrekladu` (grafická informace o stavu překladu), `nastavitPoziciPrekladu` (textová informace o stavu překladu), `vypsatiAktualniKPrekladu` (zobrazení), `goToPozice`, `nacistDalsiKPrekladu`, `nacistPredchoziKPrekladu` (pozice v překladu), `ulozitPreklad`, `potvrditPreklad` (editace). Tyto metody úzce souvisejí s třídou `TranslatorData`.

Metody `najitKodJazyku`, `najitLocaleJazyku`, `prelozitBing` a `showResult` jsou určeny k strojovému překladu textu. Překlad spočívá v odeslání textu k překladu a doplňujících informací (zdrojový a cílový jazyk) na server a čekání na odpověď

s překladem. Jelikož čekání by blokovalo GUI, je pro strojový překlad vytvořeno vlastní vlákno, po jehož dokončení se zobrazí výsledek v UI.

Metody `onInit`, `ttsMenuInit`, `saySource` a `sayTarget` umožňují aplikaci využívat TTS engine (aktuální engine a dostupné jazyky závisí na zařízení). Metoda `onPrepareOptionsMenu` upravuje podle dostupnosti jazyků zobrazení menu aktivity – `menu.findItem(R.id.itemSayKPrekladu).setEnabled(ttsSource)`. Metoda `onInit` nastává po inicializaci TTS engine a nastavuje pomocné proměnné určující dostupnost jazyků, `ttsMenuInit` má obdobnou funkci a nastavává při změně jazyků v nastavení aplikace. Samotné použití TTS probíhá voláním metody `textToSpeech.speak(target, TextToSpeech.QUEUE_FLUSH, null)`.

2.7.3.1 *TranslatorData, TranslatorDataItem*

`TranslatorDataItem` reprezentuje jednotlivou část překladu nebo data, která jsou součástí textu, ale nepřekládají se. `TranslatorData` potom zaštituje celý překládaný text a pozici v překladu. Podle atributů jednotlivých částí překladu (`prelozeno` a `kPrekladu`) určuje třída `TranslatorData` stav překladu a umožňuje posuny v něm či vrácení výsledného překladu – pouze u částí označených za přeložené se použije přeložený text, u ostatních se použije text původní (u částí neurčených k překladu samozřejmě původní text).

`TranslatorDataItem` obsahuje metody pro zpracování překládaného textu na jednotlivé části podle typu překladu.

2.7.4 *TranslatorResultActivity*

Metoda `onCreate` inicializuje aktivitu, přiřazuje rozložení GUI z prostředků aplikace a zobrazí přeložený text.

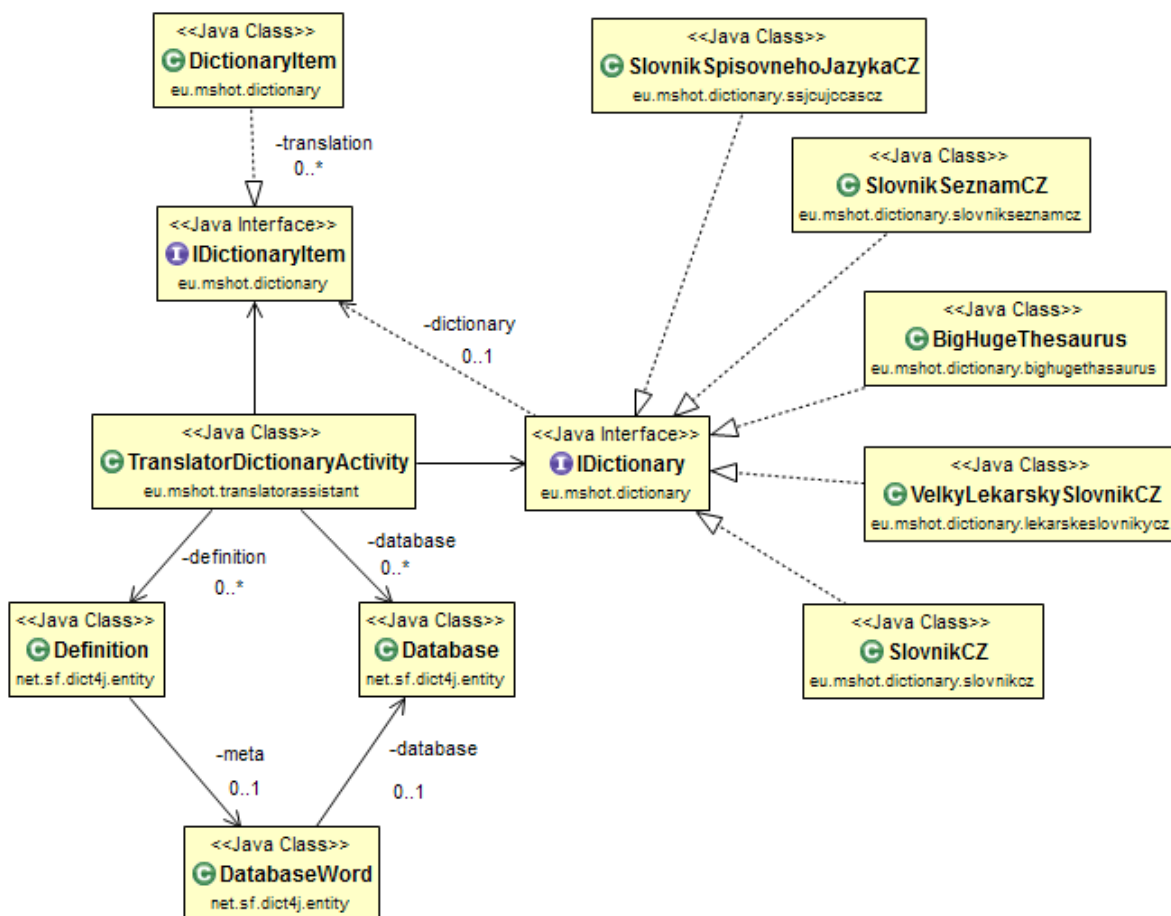
V metodě `onOptionsItemSelected` je obsluha voleb z menu – uložení do souboru, zkopírování do schránky, odeslání přeloženého textu, vyvolání nastavení aplikace, zahájení nového překladu a ukončení aplikace. Pro uložení souboru využívá aplikace aktivitu `Andexploreru` (stejně jako `TranslatorAssistantActivity`). Odeslání překladu využívá v systému zaregistrované možnosti odeslání – výběr z nabídky po zavolání `Intent(android.content.Intent.ACTION_SEND)` – možnosti odeslání (sdílení) jsou závislé na aplikacích nainstalovaných v zařízení např. email, facebook, sms.

Zachování posledního stavu aplikace mají na starosti metody `onPause` a `onStart`.

2.7.5 *TranslatorDictionaryActivity*

Atributy `translation`, `definition` obsahují vrácené výsledky vyhledávání a slouží pro jejich zobrazení, `mHandler`, `mHandlerDictionary` jsou atributy pro oddělení vyhledávání ve slovníku do vlastního vlákna.

Metoda `onCreate` inicializuje aktivitu, přiřazuje rozložení GUI z prostředků aplikace.



Obrázek 7 - UML Dictionary

Pro obsluhu UI slouží metoda `clear` – obnovení UI do výchozího stavu a metoda `find` – pokud je zadán řetězec k vyhledání a je k dispozici internetové připojení, tak s pomocí zvoleného slovníku nebo databáze (viz Obrázek 7 - UML Dictionary) najde překlady, výklady, synonyma, antonyma a rýmy v tomto zdroji (dle zvoleného zdroje se mění význam vrácených záznamů), poté s pomocí metody `showResult` a `createAdapter` je zobrazí. Stejně jako pro strojový překlad je i pro vyhledávání ve slovníku vytvořeno vlastní vlákno, po jehož dokončení se zobrazí výsledek v UI.

Volba zdroje je prováděna v preferencích aplikace, pokud nabízí jeden zdroj více slovníků či databází, tak jsou tyto zobrazeny v aktivitě pomocí metody `showDictionaries`.

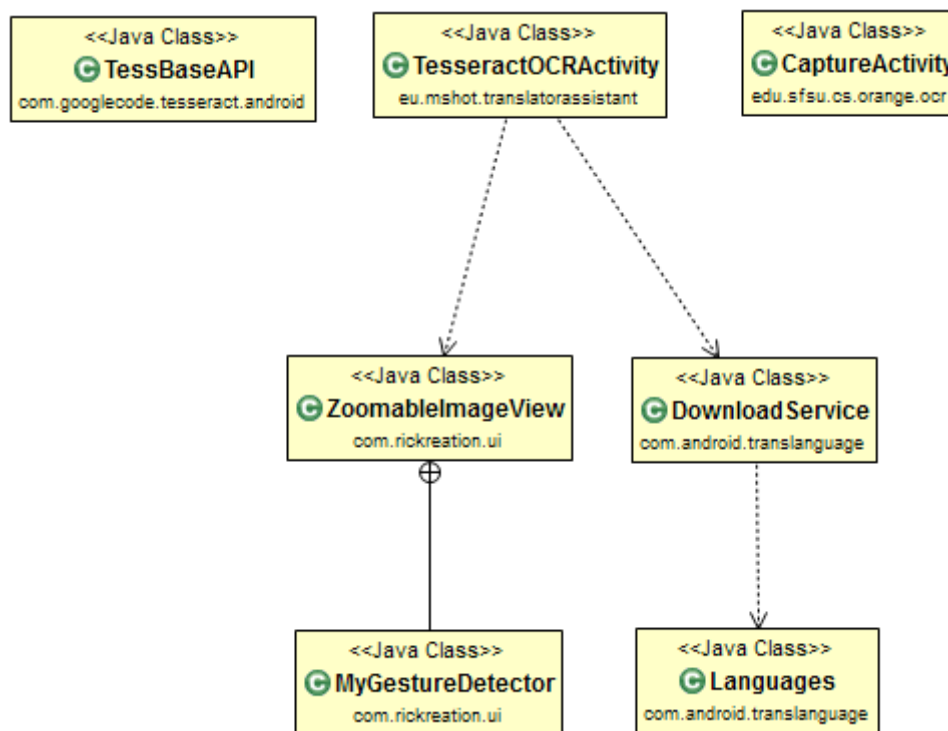
2.7.6 TesseractOCRActivity

Atribut `textFromImage` obsahuje rozpoznáný text, `bitmap` obsahuje rozpoznávaný obrázek a slouží k jeho úpravám, `mHandler` je atribut pro oddělení operací blokujících GUI do vlastního vlákna. `dialog` umožňuje zobrazení informace o průběhu akce.

Metoda `onCreate` inicializuje aktivitu, přiřazuje rozložení GUI z prostředků aplikace a nastaví obraz do horizontálního režimu – výhodnější režim pro většinu obrázků. Také pomocí metody `setOnLongClickListener` umožní přechod mezi rozpoznáním textem a obrázkem (a naopak) pomocí dlouhého dotyku – samotné přepnutí pohledu obstarává

metoda `switchView` pomocí nastavování viditelnosti prvků (`setVisibility(View.GONE/View.VISIBLE)`). Zde by jako alternativní způsob zobrazení šlo implementovat fragmenty.

Získání preferencí je realizováno metodou `getPrefs` (v metodě `onStart`), která zároveň řeší při dostupnosti internetového připojení start služby pro stažení natrénovaných dat jazyka nastaveného pro OCR.



Obrázek 8 - UML OCR

V metodě `onOptionsItemSelected` je obsluha voleb z menu – otevření a zobrazení obrázku, jeho úpravy, rozpoznání znaků, přepnutí pohledu mezi textem a obrázkem, vrácení rozpoznávaného textu a vyvolání nastavení aplikace. Pro otevření souboru využívá aplikace aktivitu `Andexploreru`, zobrazení zajišťuje komponenta `ZoomableImageView`. Úpravy obrazu slouží k lepším výsledkům OCR a sestávají se ze zoomování (2x zvětšení, možno volat opakovaně) a přidání okraje (25 pixelů ze všech stran, bílá barva). Obě tyto operace mají jisté výpočetní nároky, a proto jsou odděleny do vlastních vláken, po jejichž vykonání se promítne změna do UI. Vrácení rozpoznávaného textu do aktivity, která volala tento záměr, spočívá v přidání těchto dat k objektu `Intent`, který vrací tato aktivita – `returnIntent.putExtra("ocrText", ocrText)`, a v potvrzení kladného výsledku této aktivity – `setResult(RESULT_OK, returnIntent)`.

Metoda `ocrImage` obsahuje samotné rozpoznání textu z obrázku (opět samostatné vlákno). Nejprve převede bitmapu do pole hodnot `byte` pro třídu `ByteBuffer`, následně jsou tato data předána do třídy `Pix` (knihovna `Leptonica`, je součástí `tesseract-ocr`), a poté podle jazykových preferencí dojde k inicializaci OCR engine (pokud jsou k dispozici natrénovaná jazyková data) a k samotnému rozpoznání textu z obrázku. Tesseract je

následně uvolněn z paměti (jelikož jde o JNI je nutno paměť tímto způsobem uvolnit ročně) a výsledek zobrazen. Metoda `najitRetezecJazyku` slouží k přiřazení zvoleného jazyka rozpoznávání k názvu souboru s jeho natrénovanými daty.

`DownloadService` slouží pro stahování natrénovaných dat pro OCR dle aktuálně zvoleného jazyku. Dostupné jazyky jsou definované v třídě `Languages`.

3 Implementace systému audio-vizuální syntézy řeči

3.1 Systém audio-vizuální syntézy řeči

Systém audio-vizuální syntézy řeči je prostředek audio-vizuální syntézy řeči. Jedná se buď o renderovanou nebo snímkovou animaci hlavy, která artikuluje rty při syntéze řeči a tedy poskytuje vizuální interpretaci lidské řeči. Vizuální a akustická část musí být sesynchronizovaná, v opačném případě by výstup tohoto systému vypadal jako špatně nadabovaný herec z cizojazyčného filmu.

Pro implementaci systému audio-vizuální syntézy řeči je zapotřebí TTS knihovna poskytující kromě samotné syntézy řeči také dodatečné informace o obsažených fonémech a délce jednotlivých fonémů v syntetizovaném textu. Dále je potřeba (pro snímkovou animaci) dostatečné množství snímků mluvčího při vyslovování jednotlivých fonémů, tak, aby byl každý možný foném zastoupen ideálně vícekrát. Snímky, které posloužily pro vizuální syntézu, vznikly rozstříháním nahrávané promluvy (videa) na jednotlivé fonémy v ní obsažené.

Vzhledem k omezení, že pro český jazyk byla k dispozici pouze TTS knihovna pracující přes příkazový řádek, je tato část implementována jako desktopová aplikace v jazyce C#, výstupem je video, čímž není omezena cílová platforma. Jedná se spíše o základní demonstraci možností, jelikož komplexní zpracování systému audio-vizuální syntézy řeči značně přesahuje rozsah této práce.

3.2 Implementace

Syntézu řeči obstarává TTS knihovna, která převádí zadaný text do audio souboru .wav (Waveform audio file format – bezztrátový nekomprimovaný formát). K tomu generuje 2 další soubory – v jednom jsou postupně uvedeny fonémy v pořadí, ve kterém byly generovány, a v druhém je zaznamenána jejich délka. Tyto soubory jsou klíčové pro vizuální syntézu.

Před samotnou vizuální syntézou bylo zapotřebí označit na snímcích důležité body, konkrétně 6 následujících bodů:

- koutky očí – pro zarovnání snímku hlavy (eliminace naklonění)
- koutky úst – šířka úst
- středy okrajů dolního a horního rtu – výška úst

Ručně označené body na snímcích byly uloženy do XML souborů dle následujícího XSD schématu. Atribut position určuje, o který bod se jedná (z pozice subjektu snímku: 0 – koutek pravého oka, 1 – koutek levého oka, 2 – pravý koutek úst, 3 – levý koutek úst, 4 – střed okraje horního rtu a 5 – střed okraje dolního rtu).

Formát XML byl zvolen pro své výhody: Jednoduchá syntaxe, lze snadno generovat a zpracovávat, běžně čitelný a simulovatelný pomocí textového editoru, nezávislý na použitém programovacím jazyku a platformě. (8)

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="XMLFacer"
  targetNamespace="http://tempuri.org/XMLFacer.xsd"
  elementFormDefault="qualified"
  xmlns="http://tempuri.org/XMLFacer.xsd"
  xmlns:mstns="http://tempuri.org/XMLFacer.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Facer" type="mstns:FacerType"/>

  <xs:complexType name="FacerType">
    <xs:sequence>
      <xs:element name="Point" type="mstns:PointType"
        minOccurs="6" maxOccurs="6"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="PointType">
    <xs:sequence>
      <xs:element name="X" type="xs:integer"/>
      <xs:element name="Y" type="xs:integer"/>
    </xs:sequence>
    <xs:attribute name="position" type="xs:integer"/>
  </xs:complexType>

</xs:schema>
```

Obrázek 9 - XSD schéma pro označení bodů na snímku

Pro zvolenou metodu vizuální syntézy je nejdůležitějším údajem vzdálenost mezi dolním a horním rtem. Tento údaj poslouží pro napojení snímků vyjadřujících pohyb rtů při vyslovování fonémů bez zbytečných „skoků“ – možné pouze pokud je k dispozici více variant snímků jednotlivých fonémů, mezi kterými si lze volit.

Označené snímky byly zmenšeny na rozlišení 640*480 pixelů – dostatečné rozlišení pro předpokládané použití – a podle koutků očí byla hlava mluvčího umístěna do přibližně stejné pozice pro všechny snímky (zarovnání je ovlivněno přesností zadání těchto bodů).



Obrázek 10 - Snímek pro vizuální syntézu

Z takto připravených snímků rozřazených v adresářích dle jednotlivých fonémů (včetně odlišených variant) se sestaví datový model poskytující informace o jednotlivých dostupných variantách fonémů algoritmu pro vizuální syntézu.

Vstupem algoritmu vizuální syntézy jsou fonémy a jejich délky, z délky fonému se určí počet snímků, kterými bude reprezentován s přihlédnutím k předchozím snímkům – každý foném je zastoupen minimálně jedním snímkem, a vzhledem k tomu, že délka fonému není beze zbytku dělitelná délkou snímků (pro 25 snímků za vteřinu /FPS/ se jedná o 40ms), jsou zohledňovány tyto přesahy a tedy máme-li k dispozici po předchozím fonému např. 15ms a aktuální foném má délku 95ms tak po odečtení minimálního počtu snímků (1 snímek => 40ms) je k dispozici dohromady 70ms a tedy je prostor na dva další snímky reprezentující tento foném (celkem tři snímky), a pro další foném se uvažuje přesah -10ms.

Z výšky předchozího fonému (jeho prostředního snímku), se vybere pro aktuální foném varianta s nejmenším absolutním rozdílem výšek. Každá varianta fonému je určena různým počtem snímků a proto je potřeba zajistit, aby jich byl dostatečný počet, a zároveň, aby byly zvoleny snímky nejvíce reprezentující tento foném. Toho je dosaženo preferencí prostředních snímků. Pokud jsou pro foném potřeba např. 4 snímky a model pro nejvhodnější variantu fonému obsahuje pouze tři snímky, tak je prostřední snímek

zopakován, naopak pokud by model pro nevhodnější variantu fonému obsahoval snímky pět, tak by byl jeden krajní snímek vynechán.

Takto vybrané snímky jsou seřazeny za sebou a s pomocí FFmpeg/libavcodec převedeny společně s výše zmíněným audio souborem na video - soubor .mp4 s 25 FPS.

4 Vlastnosti aplikace, navržené funkce a jejich využití

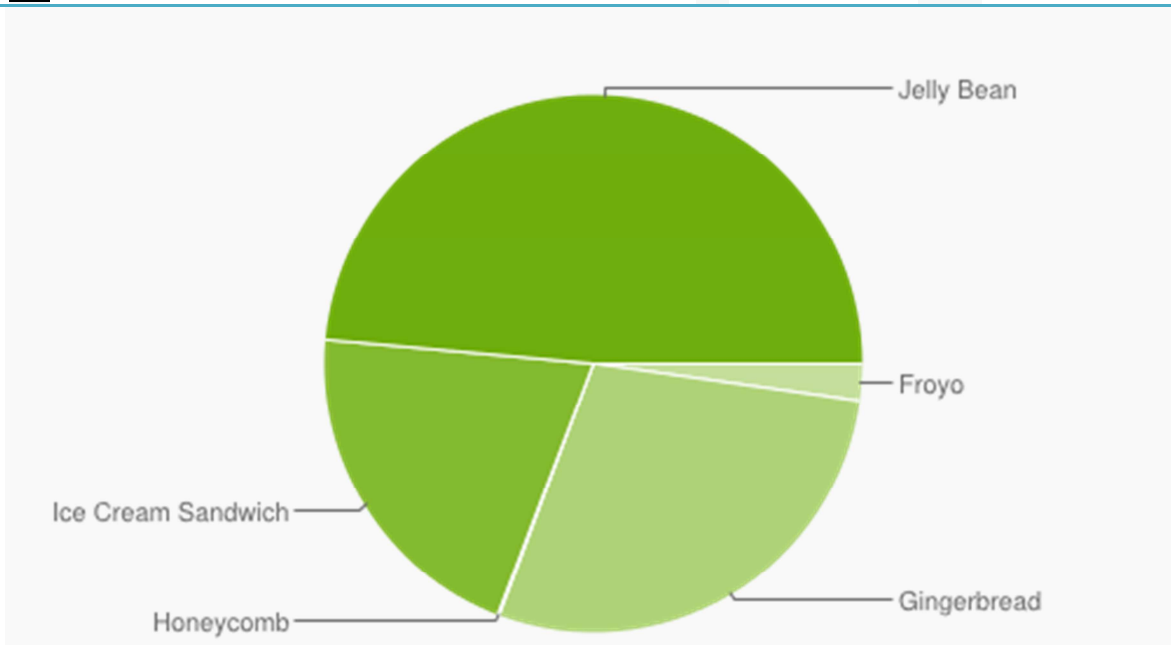
V této části bude nastíněno používání aplikace z uživatelského pohledu a možné výsledky dosažitelné s pomocí implementovaného OCR.

4.1 Minimální softwarové požadavky pro zprovoznění navržené aplikace

Aplikace je určena pro operační systém Android. Minimální vyžadovaná verze kompatibilní s aplikací je Android 2.2 Froyo (API Level 8). Dle naměřených dat by tedy měla být kompatibilní s 99,9% aktivně používaných telefonů či tabletů s tímto operačním systémem.

Tabulka 4 - Zastoupení verzí systému

Verze	Označení	API	Zastoupení
<u>1.6</u>	Donut	4	<0.1%
<u>2.1</u>	Eclair	7	<0.1%
<u>2.2</u>	Froyo	8	2.2%
<u>2.3 - 2.3.2</u>	Gingerbread	9	<0.1%
<u>2.3.3 - 2.3.7</u>		10	28.5%
<u>3.1</u>	Honeycomb	12	<0.1%
<u>3.2</u>		13	0.1%
<u>4.0.3 - 4.0.4</u>	Ice Cream Sandwich	15	20.6%
<u>4.1</u>	Jelly Bean	16	36.5%
<u>4.2</u>		17	10.6%
<u>4.3</u>		18	1.5%



Obrázek 11 - Zastoupení verzí systému

- ✓ V zařízení musí být dostatek volného místa nejen pro instalaci aplikace ale též pro jazykové balíčky k rozpoznávání textu z obrazu. Tyto balíčky se ukládají na rozšiřující SD kartu.
- ✓ Obecně lze doporučit větší rozlišení displeje a dostatečně výkonný procesor pro lepší komfort při využívání aplikace.
- ✓ Rozpoznávání v reálném čase vyžaduje přítomnost fotoaparátu v zařízení.
- ✓ Pro vyhledávání ve slovnících a tezaurech a též pro využití překladače Bing je nutné aktivní internetové připojení.
- ✓ Pro práci se soubory je potřeba aplikace AndExplorer – freeware dostupný v Google Play!.

4.2 Jazyky podporované aplikací

4.2.1 Podpora pro strojový překlad

V následující tabulce je výčet podporovaných jazyků, pro jazyky zde neuvedené nelze využít strojový překlad pomocí služby Bing Translator. Překládat ručně lze samozřejmě do libovolného jazyka, ale bez asistence strojového překladu.

Tabulka 5 - Podporované jazyky překladu

Arabic	French	Korean	Slovenian
Bulgarian	German	Latvian	Spanish
Catalan	Greek	Lithuanian	Swedish
Chinese Simplified	Haitian Creole	Malay	Thai
Chinese Traditional	Hebrew	Norwegian	Turkish
Czech	Hindi	Persian	Ukrainian
Danish	Hmong Daw	Polish	Urdu
Dutch	Hungarian	Portuguese	Vietnamese
English	Indonesian	Romanian	
Estonian	Italian	Russian	
Finnish	Japanese	Slovak	

4.2.2 Podpora pro rozpoznávání skenovaného textu

V následující tabulce je výčet jazyků, pro které jsou dostupná natrénovaná data (tesseract verze 3.x), a pro které tedy mohou být rozpoznávány texty z obrazu. Pro jiné jazyky by se musela nejprve data natrénovat.

Tabulka 6 - Podporované jazyky OCR

Afrikaans	Estonian	Indonesian	Russian
Albanian	Finnish	Italian	Serbian (Latin)
Arabic	Frankish	Italian (Old)	Swedish
Azerbajjani	French	Japanese	Tagalog
Basque	French (1400-1600)	Kannada	Tamil
Belarusian	Galician	Korean	Telugu
Bengali	German	Latvian	Thai
Bulgarian	Greek	Lithuanian	Slovak
Catalan	Greek (Ancient)	Macedonian	Slovenian
Croatian	Hebrew	Malay	Spanish
Czech	Hindi	Malayalam	Spanish (Old)
Danish	Hungarian	Maltese	Swahili
Dutch	Cherokee	Norwegian	Turkish
English	Chinese (Simplified)	Polish	Ukrainian
English (1100-1500)	Chinese (Traditional)	Portuguese	Vietnamese
Esperanto	Icelandic	Romanian	

4.3 Slovníky a tezaury

Jako tezaurus pro angličtinu je dostupný BigHugeThesaurus (<http://words.bighugelabs.com/>). Slovníky překladové jsou slovník.cz a slovník.seznam.cz, dále slovník spisovného jazyka českého (ssjc.ujc.cas.cz), a výkladový Velký lékařský slovník (lekarske.slovníky.cz).

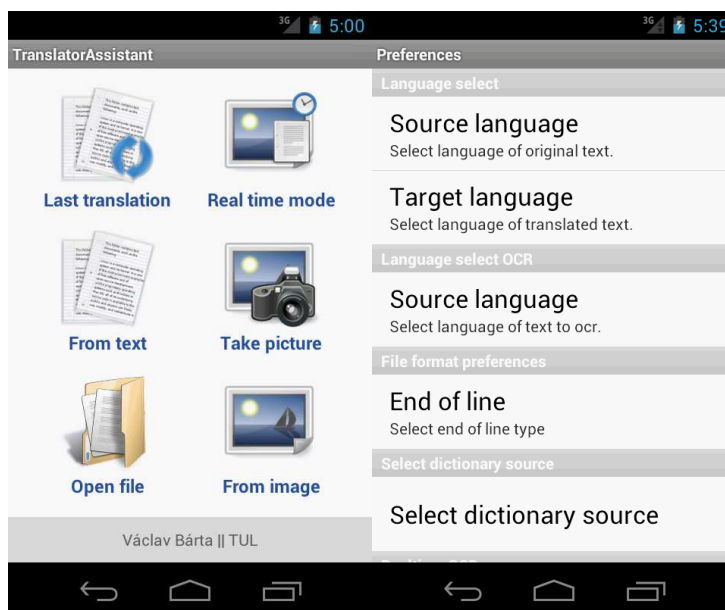
Další desítky slovníků nabízí dict.org databáze – jazykové i výkladové slovníky, výčet viz níže (www.dict.org/w/databases/start).

4.4 Startovací obrazovka

Úvodní obrazovka aplikace má formu rozcestníku. Jednotlivé volby odpovídají možnostem aplikace ve třech základních režimech. Prvním režimem je překlad z textu (Last translation/ From text / Open file), druhým je rozpoznávání textu z obrazu (From

image / Take picture) a třetím rozpoznávání textu z obrazu kamery v reálném čase s automatickým překladem (Real time mode).

V případě potřeby rychle přeložit neznámý text, např. na značce či jiném krátkém sdělení je vhodné použít rozpoznávání v reálném čase z fotoaparátu, při překladu delšího textu, u kterého záleží nikoliv na pochopení smyslu, ale na přesnosti překladu, je vhodné použít textový režim s asistenčními nástroji, jako jsou slovníky a tezaury, či orientační strojový překlad (prvky CAT).



Obrázek 12 - Úvodní obrazovka

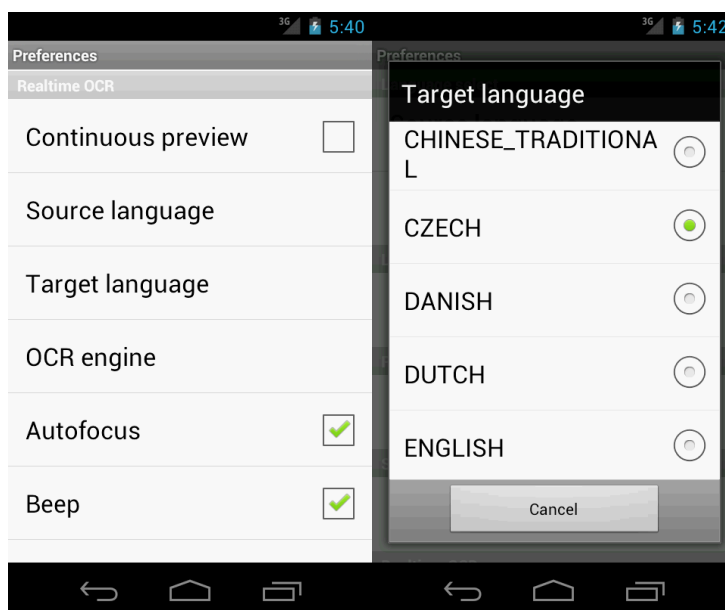
Obrázek 13 - Nastavení

4.5 Nastavení aplikace

Volba Nastavení (Preferences) v libovolné části aplikace zobrazí nastavení aplikace.

- ✓ Zdrojový a cílový jazyk textu pro automatický překlad v textovém režimu.
- ✓ Zdrojový jazyk textu v režimu rozpoznání textu z obrazu.
- ✓ Konce řádků ukládaných textových souborů.
- ✓ Preferovaný slovník.
- ✓ Živý náhled (RealTime).
- ✓ Zdrojový a cílový jazyk textu pro automatický překlad v RT režimu.
- ✓ OCR knihovna.

- ✓ Automatické zaostřování a další možnosti pro zpřesnění rozpoznávání.
- ✓ Zvuková signalizace.

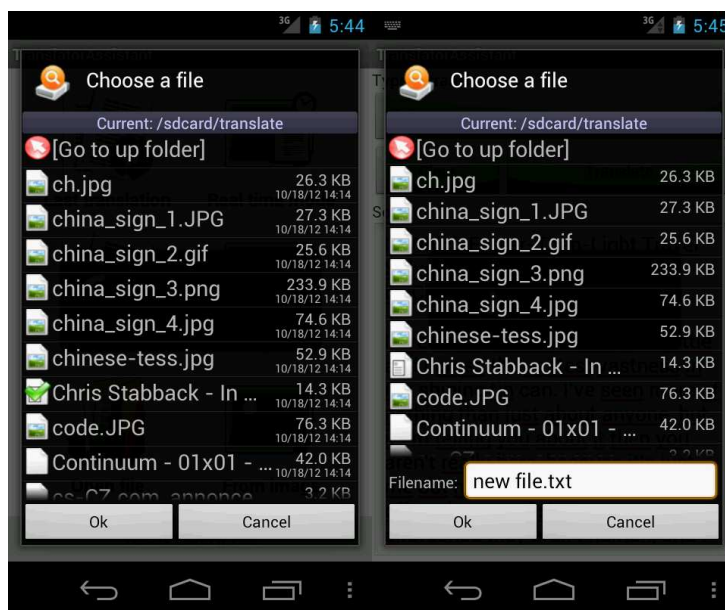


Obrázek 14 - Nastavení RT

Obrázek 15 - Nastavení jazyku

4.6 Práce se soubory

Pomocí okna AndEploreru je možné vybrat soubor k otevření (text k překladu nebo obrázek k rozpoznání), či k uložení (přepsání existujícího souboru / vytvoření nového).



Obrázek 16 - Výběr souboru

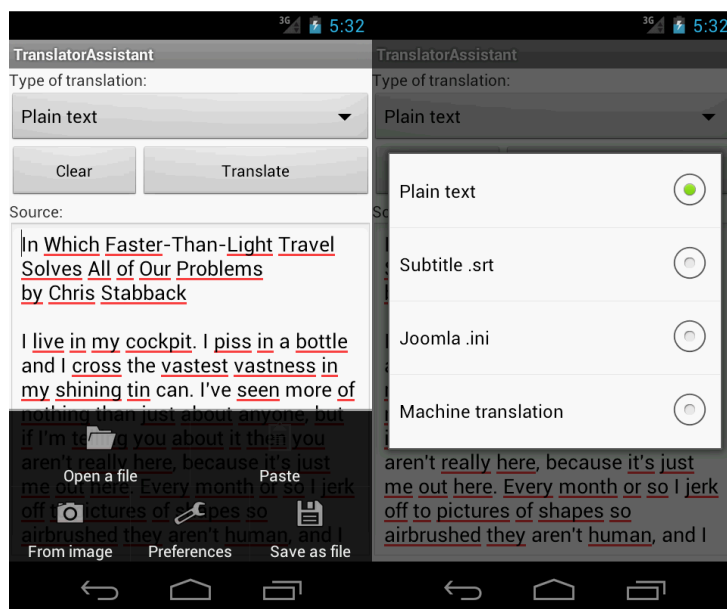
Obrázek 17 - Ukládání souboru

4.7 Překlad z textu

4.7.1 Překládaný text

První obrazovka aplikace slouží k výběru typu překladu a k zadání překládaného textu. Text je možno zadat ručně, ze schránky, ze souboru či pomocí rozpoznání z obrázku. Pomocí tlačítka „Clear“ lze resetovat zadané údaje, tlačítko „Translate“ slouží k použití zadaného textu k překladu, po jeho stisknutí dojde k přesunu na další obrazovku (viz Překlad).

- ✓ „Plain text“ – překlad libovolného textu, dělení na segmenty podle vět.
- ✓ „Subtitle“ – překlad titulků ve formátu SubRip (přípona .srt), dělení na segmenty podle jednotlivých titulků.
- ✓ „Joomla“ – překlad lokalizačních souborů – přípona .ini, dělení na segmenty podle NÁZEV=TEXT.
- ✓ „Machine translation“ – určeno pro automatický překlad, text bez segmentace.

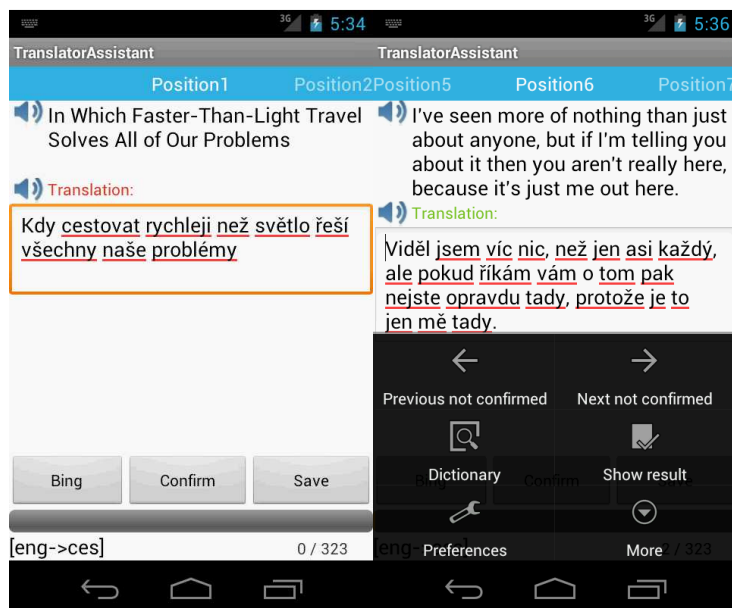


Obrázek 18 - Aktivita

Obrázek 19 - Režim překladu

4.7.2 Překlad

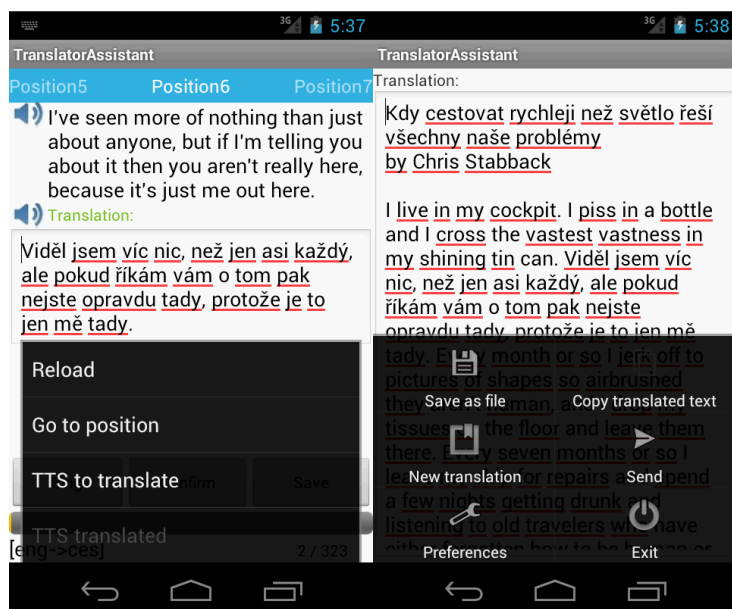
Překlad probíhá po jednotlivých částech – segmentech, obrazovka překlad proto obsahuje ve spodní části indikátor (grafický i textový) s informacemi o stavu překladu, v horní části je indikována pozice v překladu.



Obrázek 20 - Překlad segmentu

Obrázek 21 - Překlad segmentu - možnosti /1/

Jednotlivé segmenty je možno uložit („Save“) před potvrzením („Confirm“) – potvrzení také ukládá. Pouze potvrzené části jsou považovány za přeložené a z nich je složen výsledný text. Nepotvrzené části jsou uvozeny červenou barvou, potvrzené části barvou zelenou. „Bing“ slouží k strojovému překladu textu. Pokud je u textu k překladu nebo u titulku přeloženého textu ikona, znamená to, že pro tento text (závislé na jazyku zdroje a cíle) je možné provést syntézu řeči.



Obrázek 22 - Překlad segmentu - možnosti /2/

Obrázek 23 - Přeložený text – možnosti

„Reload“ slouží k načtení uloženého textu (tedy pro návrat k uloženému stavu překladu pro danou část, samozřejmě pokud nebyl nový překlad uložen). Volbou „Show result“ dojde k zobrazení obrazovky s výsledným překladem (lze se poté vrátit zpět a provést případné změny).

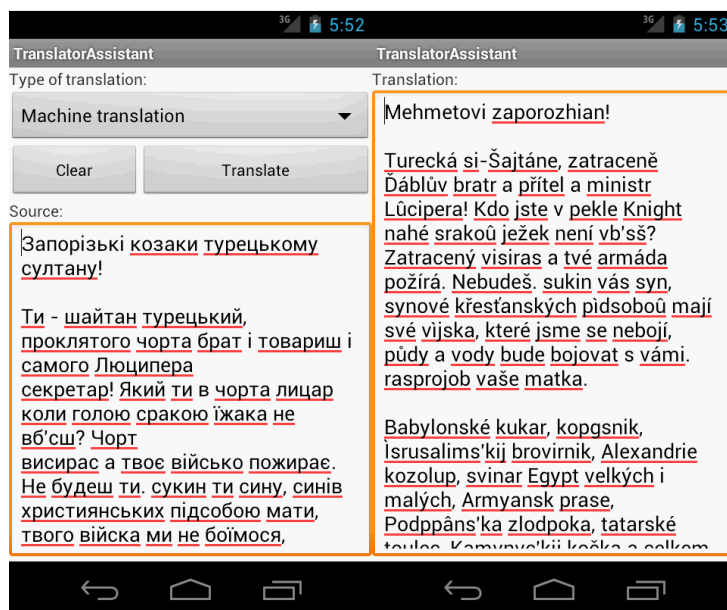
„Dictionary“ slouží k vyhledávání ve slovnících či tezaurech.

4.7.3 Přeložený text

Obrazovka výsledek zobrazuje přeložený text a umožňuje s ním provádět některé operace. Kromě možnosti dodatečné editace nabízí aplikace uložení textu do souboru („Save as file“), zkopírování do schránky („Copy translated text“), odeslání např. emailem („Send“).

„New translation“ slouží k přechodu na úvodní obrazovku a tedy k zahájení nového překladu. „Exit“ ukončí aplikaci.

Na obrázcích níže je vidět příklad automatického překladu – překládaný text v ukrajinštině a výsledek získaný pomocí strojového překladu do češtiny.



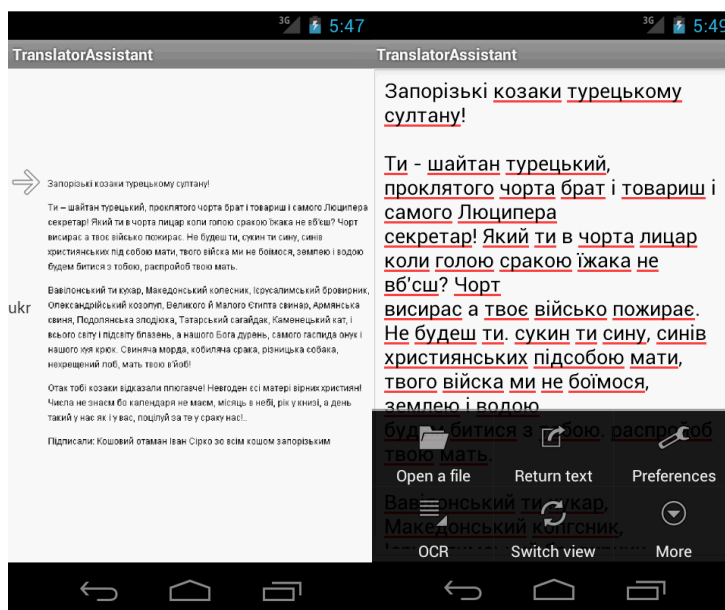
Obrázek 24 - Strojový překlad /1/

Obrázek 25 - Strojový překlad /2/

4.7.4 Rozpoznávání textu z obrazu

Volba „From image“ zobrazí obrazovku sloužící k rozpoznání textu z obrázku. Nejprve je potřeba pomocí volby „Open a file“ otevřít obrázek určený k rozpoznání. (Pokud není načten obrázek z fotoaparátu pomocí volby „Take picture“). Poté je možno ho např. zazoomovat (obsaženo ve volbě „More“), či přidat okraj pro lepší výsledek rozpoznávání nebo spustit samotné rozpoznání textu volbou „OCR“. Dialog informuje o průběhu rozpoznávání, po dokončení této operace je zobrazen rozpoznávaný text. Mezi

textem a obrazem lze přecházet buď pomocí volby „Switch view“ nebo dlouhým podržením v textu či šipky vlevo vedle obrázku.

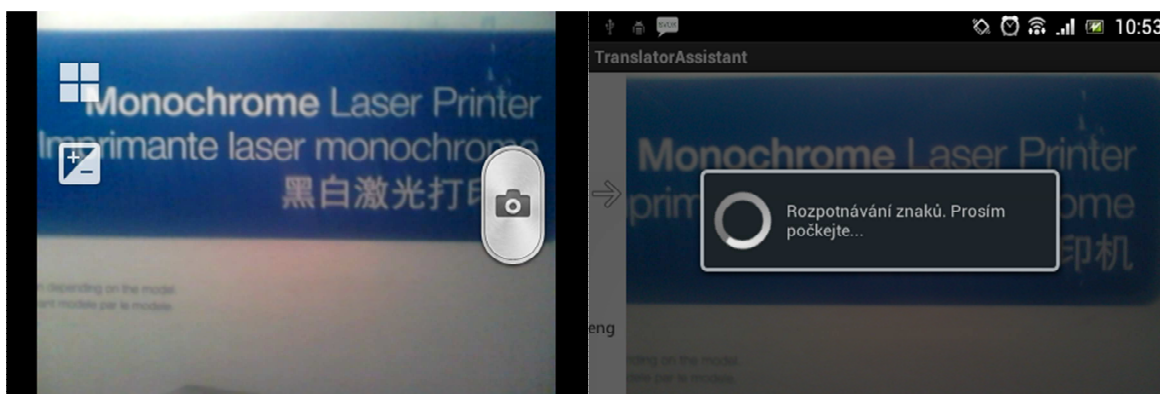


Obrázek 26 - OCR předloha

Obrázek 27 - OCR rozpoznáný text

Pomocí „Return text“ dojde k ukončení tohoto režimu a rozpoznáný text je předán k textovému překladu (viz Překládaný text). Předchozí obrázky v části Přeložený text ukazují příklad použití rozpoznávaného textu pro strojový překlad do jiného jazyka.

Samotné rozpoznávání (i úpravy obrázku) je výpočetně náročné – doba pro rozpoznání je závislá na rozpoznávaném jazyce, na kvalitě a rozměrech rozpoznávaného obrázku a na výkonu zařízení.

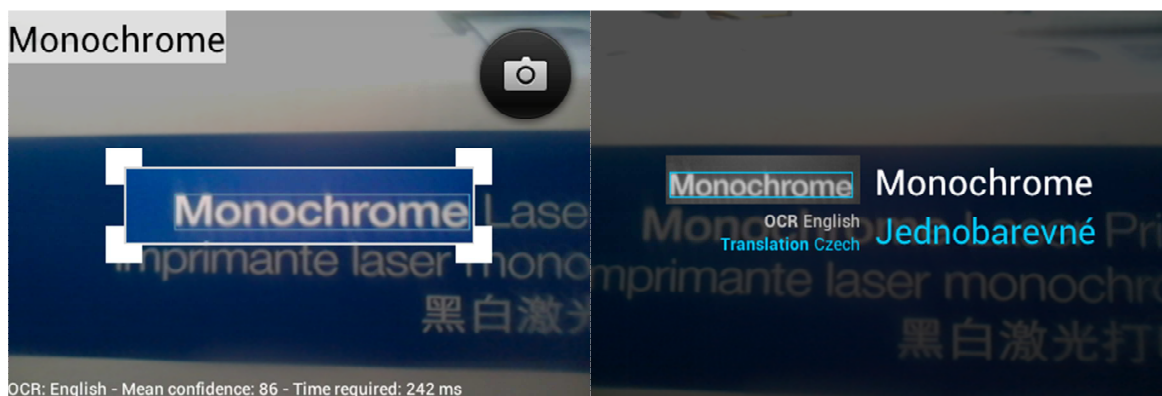


Obrázek 28 - OCR z fotografie /1/

Obrázek 29 - OCR z fotografie /2/

4.8 Rozpoznávání textu z obrazu v reálném čase

Rozpoznávání v reálném čase (živý režim) probíhá ve dvou krocích. Nejprve označíme rozpoznávaný text pomocí nastavitelné oblasti uprostřed záběru a zkontrolujeme, zda byl rozpoznán text (v levém horním rohu). Po stisku tlačítka s ikonou fotoaparátu se aplikace pokusí rozpoznat text automaticky přeložit a zobrazí tento výsledek. (Pokud není zvolen živý režim, OCR knihovna nenabízí náhled na rozpoznatý text v prvním kroku).



Obrázek 30 - OCR v reálném čase /1/

Obrázek 31 - OCR v reálném čase /2/

5 Výsledky rozpoznávání

Zvolené obrázky zachycují buď naskenované/vyfocené textové dokumenty (viz Příloha A) nebo krátká informační sdělení. Samotné rozpoznávání je výpočetně náročné – doba pro rozpoznání je závislá na rozpoznávaném jazyce, na kvalitě a rozměrech rozpoznávaného obrázku a na výkonu zařízení.

5.1 Jazyky - Asie



Obrázek 32 - Japonský nápis /1/

Obrázek 33 - Japonský nápis /2/

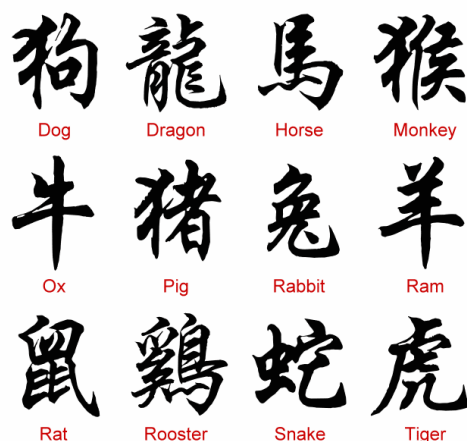
Obrázek 32 - Japonský nápis /1/ – rozpoznáno お手洗い – Bing: Washroom.

Obrázek 33 - Japonský nápis /2/ – bílé na černém nerozpoznáno správně, černé na bílém ano – 門前仲町駅 – Bing: Big tits Monzen nakacho station.



Obrázek 34 - Japonský nápis /3/

Obrázek 34 - Japonský nápis /3/ – rozpoznání znaků dostatečné (オナラの時の緩方躍誌。たばこの時は忘れてたzs), překlad pomocí Bingů nesmyslný.



Obrázek 35 - Čínský nápis /1/

Obrázek 36 - Čínský nápis /2/

Obrázek 35 - Čínský nápis /1/ - rozpoznání – Bing: Like this? TL meet Agaw wrong, spring compliance po channeling civilization garden Mao Wanyi prohibition of climb...

Obrázek 36 - Čínský nápis /2/ – rozpoznáno více znaků než obsahuje - Bing: Dogs and dog Dragon Horse monkey Dragon – „vynechano“ – ox rabbitbandits hand hand in hand – „vynechano“.



Obrázek 37 - Čínský nápis /3/

Obrázek 38 - Čínský nápis /4/

Obrázek 37 - Čínský nápis /3/ – rozpoznáno s chybami – Bing: Visit tomato vulgar.

Obrázek 38 - Čínský nápis /4/ – rozpoznáno – Bing: Welcome to an ancient vessel for broomcorn millet.

5.2 Jazyky – Evropa

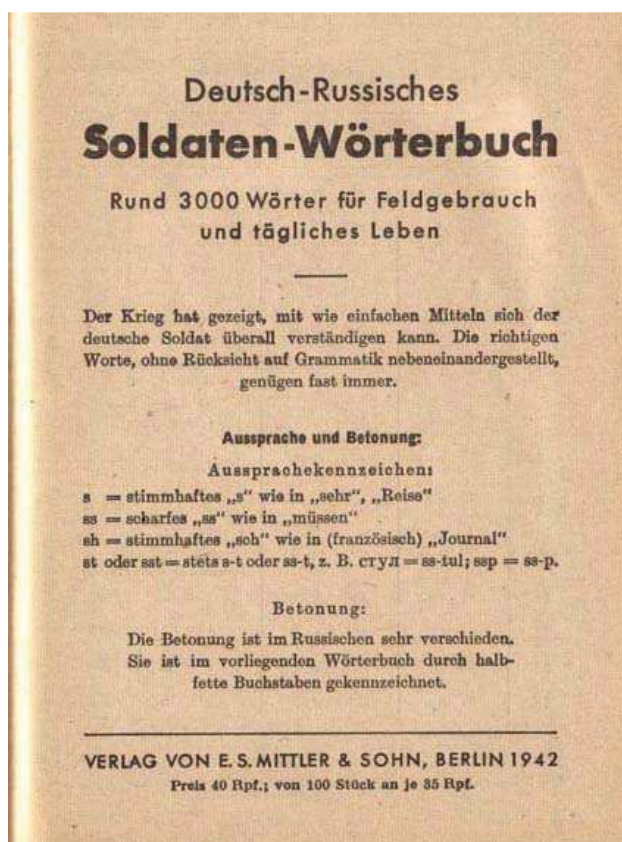


Obrázek 39 - Německý nápis /1/

Obrázek 40 - Německý nápis /2/

Chyba! Nenalezen zdroj odkazů. – rozpoznáno s chybami – Hetrelen der Baustelle verhunten.

Obrázek 40 - Německý nápis /2/ – rozpoznáno – Freies Saatgut für alle! Gegen die neue EU-Saatgut-Gesetzbung! – „vynechano“ – všv%u.suutgutkampugnz.org (zde by pomohlo zazoomování).



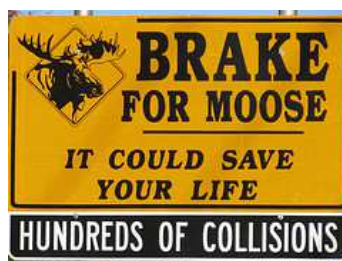
Obrázek 41 - Německý dokument /1/

Obrázek 41 - Německý dokument /1/ – bez zvětšení rozpozná téměř správně text velký písmem (horní řádky), po 4x zvětšení je patrné pouze malé zlepšení – problémem je nekvalitní obrázek.



Obrázek 42 - Anglický nápis /1,2,3/

Obrázek 42 - Anglický nápis /1,2,3/ – rozpoznáno správně pouze slovo Always a Limit. (Pravděpodobně je problém v tesseractu pro obrázky s malým obsahem textu.)



Obrázek 43 - Anglický nápis /4/

Obrázek 43 - Anglický nápis /4/ – rozpoznáno částečně – Brake Fon raosn IT COULD SAVE YOUR LIFE, po 2x zoomu je již rozpoznán správný text BRAKE FOR MOOSE IT COULD SAVE YOUR LIFE.



Obrázek 44 - Francouzský nápis /1/

Obrázek 45 - Francouzský nápis /2/

Obrázek 44 - Francouzský nápis /1/ – nerozpoznán, po 4x zoomu rozpoznáno nepřesně – sortIE Ecoua.

Obrázek 45 - Francouzský nápis /2/ – je nerozpoznatelný – zde by mohl pomoci ořez pozadí.



Obrázek 46 - Francouzský nápis /3/

Obrázek 46 - Francouzský nápis /3/ – rozpoznáno – Les Fouillesh som INTERDITES SUR LE TERRITOIRE DE LA COMMUNE DE RENNES LE CHATEAU VÿArre(é du 28.07.65.



Obrázek 47 - Italský nápis /1/

Obrázek 48 - Italský nápis /2/

Obrázek 47 - Italský nápis /1/ – rozpoznáno (drobná chyba) – FIRENZE NAPLES MRCMA POMPE II PERUGIA CINQUETERRE FLORENCE.

Obrázek 48 - Italský nápis /2/ – rozpoznání s chybami – po 2x zoomu ATTENZIONE sema o rall atore di velocita.



Obrázek 49 - Italský nápis /3,4/

Obrázek 49 - Italský nápis /3,4/ – nerozpoznatelné, nepomohlo zvětšování ani přidávání okrajů.



Obrázek 50 - Španělský nápis /1/

Obrázek 50 - Španělský nápis /1/ – rozpoznáno s drobnými chybami – SI USTED TIENE UN IVIATCAPASO O CUALQUIER...



Obrázek 51 - Španělský nápis /2,3,4/

Obrázek 51 - Španělský nápis /2,3,4/ – po 2x zvětšení rozpoznán s chybami text velkými písmeny – NO ES PERMITIDU TIRARSE DE CABEZA / NO IR EN PATINETA. / PARKING BY PERMIT ONLY ESTACIONAMIENTO NOC PEnMIso SOLAMENTE.

Neznámý útočník Jiřího Paroubka udeřil do obličeje ve středu zhruba kolem 17:30 na pražském Malostranském náměstí.

Informace o incidentu se ještě ve středu večer objevila na Paroubkově webu vasevec.cz s tím, že šlo o záměr. "Jednalo se nejspíše o zákeřný, ale také promyšlený útok. Ten, kdo jej plánoval, však zřejmě nepředpokládal, že se napadený nebude bránit," stojí na webu.

Poslanec a předseda národních socialistů byl napaden ve chvíli, kdy mířil na pracovní schůzku poblíž Poslanecké sněmovny. "Šel jsem po Malostranském náměstí a v jeho horní části jsem mluvil s americkými turistkami, kterým jsem ukazoval cestu k Pražskému hradu. V jedné ruce jsem držel tašku a přiznám se, moc jsem se nedíval kolem sebe. Najednou přišel nějaký opilec a udeřil mě," popsal pro iDNES.cz Paroubek.

Obrázek 52 - Český dokument /1/

Obrázek 52 - Český dokument /1/ – rozpoznáno včetně diakritiky, s minimem chyb (převážně velikost písmen).

5.3 Shrnutí rozpoznávání

Pokud je předloha dostatečně kvalitní, tak rozpoznání pomocí Tesseract OCR Engine ve všech testovaných jazycích umožní s pomocí strojového překladu získat představu o tom, jaké informace text obsahuje. Pokud by obrázky krátkých sdělení byly dostupné ve vyšší kvalitě, velmi pravděpodobně by stoupla přesnost rozpoznávání.

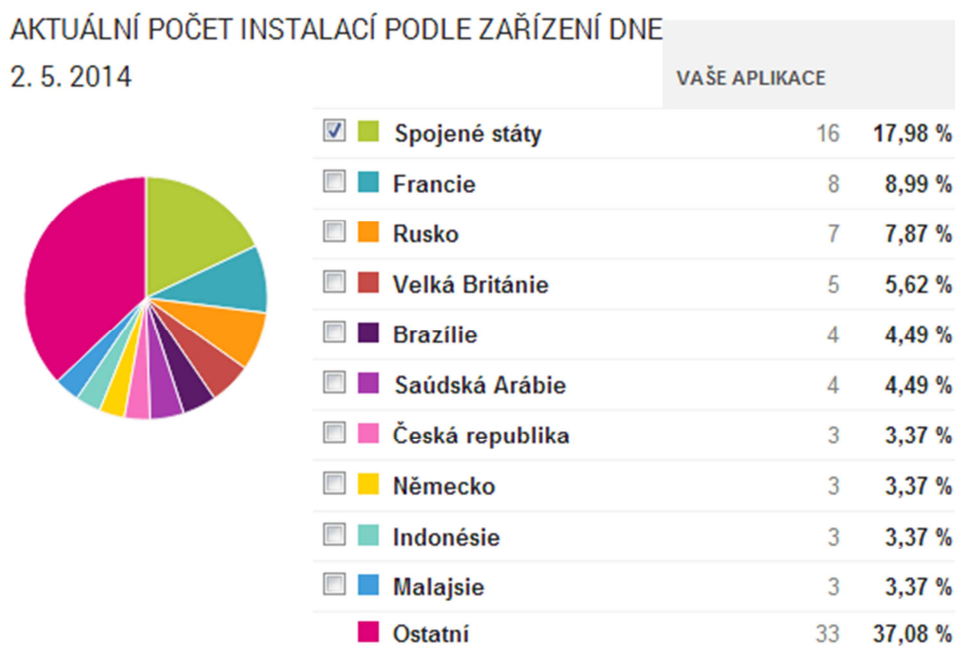
6 Závěr

V rámci této diplomové práce byl realizován CAT systém pro operační systém Android. V teoretické části se práce věnovala specifikům vývoje aplikace pro platformu Android a shrnula informace o knihovnách a komponentách, které byly použity pro realizaci aplikace. Dále obsahuje zpracované diagramy užití vyvíjené aplikace a činností, za které zodpovídají, spolu s popisem jejich implementace do jednotlivých aktivit.

V praktické části byla realizována aplikace podle požadavků na ni kladených. Vývoj probíhal v prostředí Eclipse (Indigo Service Release 1 a Kepler Service Release 1) s doplňkem Android Development Toolkit (16 - 22). Aplikace podporuje širokou škálu zařízení od verze SDK 8 - verze operačního systému Android 2.2.x – Froyo. Jako největší problém se jevílo vývojové prostředí a interakce s emulátorem, jehož chování nebylo zcela konzistentní.

V rámci této práce byly zpracovány výsledky OCR rozpoznávání pro několik významných světových jazyků, které ukazují, že kvalita rozpoznávání (v závislosti na předloze) je dostačující pro orientační překlad pomocí MT.

Implementovaná aplikace se svými vlastnostmi řadí mezi CAT nástroje, nicméně stále je několik možností, jak ji dále rozvíjet. Jedná se o případnou větší podporu formátů, použití více slovníků, možnosti ukládání vlastních překladů k opětovnému použití, integrace systému audio-vizuální syntézy řeči, který byl implementován jako desktopová aplikace v jazyce C#, atd. Aplikace je umístěna na play.google.com k bezplatnému stažení. Bez jakékoli propagace má aplikace k začátku května 2014 celkový počet instalací 689, z toho 89 aktivních po celém světě.



Obrázek 53 - Počet instalací aplikace

Literatura

1. **MURPHY, Mark L.** *Android 2: průvodce programováním mobilních aplikací*. Brno : Computer Press, 2011. str. 376. ISBN 978-80-251-3194-7.
2. **KOMATINEMI, S.** *Pro Android 3*. New York : Apress, 2011. str. 1175. ISBN 978-1-4302-3222-3.
3. **STEELE, J. a NELSON, T.** *The Android developer's cookbook*. Upper Saddle River : Addison-Wesley, 2011. str. 339. ISBN 0-321-74123-4.
4. **MORRIS, J.** *Android user interface development*. Birmingham : Packt Open Source, 2011. str. 287. ISBN 978-1-849514-48-4.
5. **ARLOW, Jim a NEUSTADT, Ila.** *UML 2 a unifikovaný proces vývoje aplikací*. Brno : Computer Press, 2007. str. 567. ISBN 978-80-251-1503-9.
6. **SHEN, Ethan.** Comparison of online machine translation tools. *Tcworld.info*. [Online] 2010. [Citace: 24. 04 2014.] <http://www.tcworld.info/e-magazine/translation-and-localization/article/comparison-of-online-machine-translation-tools/>.
7. **HEROUT, Pavel.** *Java a XML*. České Budějovice : Kopp, 2007. str. 313. ISBN 978-80-7232-307-4.
8. **HAROLD, Elliott Rusty a MEANS, W. Scott.** *XML v kostce*. Praha : Computer Press, 2002. str. 439. ISBN 80-7226-712-4.

A Rozpoznávání dokumentů

Jazyky - Asie

非木材植物 [編集]

紙の原料として使われた非木材植物には、次のものがある。いずれも、安定供給や品質の面から木材の代替にはならないとされており、現在では特別な用途で使われている。

アサ

アサやそのぼろは、中国で紙が発明されたときの主原料だった（リネンパルプ）。

カジノキ・ガンピ・コウゾ・マユミ・ミツマタ

カジノキ・ガンピ・コウゾ・マユミ・ミツマタはいずれもその樹皮が紙の原料として、中国・日本などで使われた。このうち栽培が比較的容易なコウゾは、現在和紙の主原料となっている。また、ミツマタは日本紙幣の原料として混ぜられている。

竹

竹紙は、中国で唐時代（7世紀）から作られ、宋時代（10世紀以降）には竹が紙の主原料となった。その後、藁や木材に取って代わられたが、近年ふたたび見直され、四川省などで大規模な工場も建設されている。

藁

藁（稲わらや麦わら）は、中国では唐時代から紙の原料として使われた。また、日本では1890年代頃は洋紙の主原料だった。藁には、繊維が細くて短すぎるため弱い紙しかできない、年に1回しか収穫できず腐りやすいため保管が難しい、などの問題点があるが、中国などでは、まだ原料として使用されている。

亜麻

亜麻やそのぼろは、イスラム世界で紙の主原料となった。ヨーロッパでも木材以前はよく使われた。

木綿

木綿のぼろ（ラグ）は、欧米で木材以前は紙の主原料であった。しかし、15世紀に印刷技術が確立して紙への需要が大きくなると供給不足になり、木材からの製紙方法が開発される契機となった。日本でも、製造開始直後の1880年代頃は洋紙の主原料だった。また、綿花の加工途中で生ずる地毛などの短繊維（リントー）を原料として紙を漉くこともできる。木綿のぼろから作られるパルプをラグパルプ、綿花の地毛などの短繊維から作られるパルプをリントーパルプという。なお、通常の木綿は、繊維が長過ぎるため、製紙には使いにくい。

サトウキビ

インド・中国や南米諸国では、製糖時に発生したサトウキビの絞りかす（バガス）からパルプを製造している。バガスパルプは多くの場合、製糖工場に隣接したパルプ工場生産される^[6]。

マニラアサ

マニラアサ（アバカ）は、フィリピンなどで栽培されているバショウ科の植物。アバカパルプは繊維が細長いため、しなやかで強い紙を作ることができる。現在、日本紙幣の主原料となっているほか、ティーバッグ、掃除機の紙パックの原料となっている。

Obrázek 54 - Japonský dokument

Tabulka 7 - Japonský dokument OCR+MT

<p>木材棺 𧄂 編集]</p> <p>紙の原料として使われた非木材植物には、次のものがある。いずれも、安定供給や品質の面から木材の代替にはならないとされており、現在では特別な用途で使われている。</p> <p>アサ</p> <p>アサやそのぼろは、中国で紙が発明されたときの主原料だった(リネンパルプ)。</p> <p>カジノキ・ガンピ・コウゾ・マユミ・ミツマタ</p> <p>カジノキ・ガンピ・コウブ・マユミ・ミツマタはいずれもその樹皮が紙の原料として、中国・日本などで使われた。このうち栽培が比較的容易なコウゾは、現在和紙の主原料となっている。また、ミツマタは日本紙幣の原料として混ぜられ氷ぜ亡いる。</p> <p>竹</p> <p>曼 竹紙は、中国で唐時代(升齡印に於られ、宋時代(川世紀以降)には竹が紙の主原料となった。その後、藁や木材に取って代わられたが、近年ふたたび見直され、四川省などで大規模な工場も建設されている。</p> <p>...atd</p>	<p>Non dřevo rakev 𧄂TM, úpravy]</p> <p>Rostliny nejsou ze dřeva byl použit jako surovina pro papír tam. Jednou je stabilní dodávky a produkty A kvalitní alternativu k dřevo a dnes je používán speciální</p> <p>Já o</p> <p>ASA</p> <p>ASA a jeho Sapporo, v čínské knize od TOMOYA byl hlavní složkou je přestala (リネンパルプ)。</p> <p>Večerní mulberry kniha ganpi kniha mulberry Mayumi Mitsumi</p> <p>Papíru mulberry, Gumby, Kolb, Mayumi a orientální paperbush jsou že jeho kůra je surovinou pro papír Byl použit v Číně a Japonsku. Kůra pěstování je poměrně snadné</p> <p>V současné době hlavní suroviny papíru a nebude mít. Také orientální paperbush mix jako složka NP banky</p> <p>Od ledu Ze smrti jsou.</p> <p>Bambus</p> <p>Adéla prací je dynastie Tchang v Číně (známky stárnutí Shou, bambus papír v období dynastie Song (od řeky), vyrobený z na</p> <p>Primárních surovin a člověk. Poté slámy nebo dřeva, aby se byla nahrazena v posledních letech znovu našel</p> <p>A také továrna v provincii s 'čchuan, včetně rozsáhlé stavby.</p> <p>...atd</p>
--	---

文字

也許，最主要以圖像上區別來分類的是**線條**。線條書寫系統是指字符由許多線條所組成，如拉丁字母和漢字。漢字如果是用原子筆、毛筆或鑄刻在青銅器上，則視為是線性的。類似地，埃及象形文字和瑪雅文字則通常是描繪線性外框，但在正式場合，他們雕刻在bas-relief中。另一方面，非線性系統則如盲文，不論是用什麼材質，都非由線條構成。最早的例子則是線條的：西元前 3300 年的楔形文字是線條的，雖然從它衍生的楔形文字不是。

楔形文字也許是最早的非線條文字。它的字形是由蘆葦筆尖所壓制在潮溼泥板而成，而非之前在泥板上用筆尖描繪線條。最後變成其文字的字根外觀上的變化。

盲文則是拉丁文字的非線條版本，它完全放棄拉丁文字的外形。字母是由凸塊所組成，其基底可以是皮革（路易斯·布萊葉原本採用的材質），堅硬塑膠或金屬。

也有拉丁文字的非線條的短暫性版本，包含摩爾斯電碼，手語的manual alphabet版本和藉由旗幟或布條放置在不同角度旗語。然而，如果文字是定義成永久紀錄資訊的能力的話，因為這些符號很快就會消失，則這些系統根本並非文字。

書寫方向

另見東亞文字排列方向、雙向文稿和鏡像文字

文稿的一項特徵是它們書寫的方向。埃及聖書體可以任一方向橫寫，只要動物或人的字形面向書寫它們的方向。早期的字母可以被以很多方向書寫，包含橫向（左至右或右至左）或縱向（上或下）。它一般是用牛耕式轉行書寫法。由一個（水平）方向開始，然後在一行結尾轉折以反方向書寫。

希臘字母和其後繼者則是左至右的模式，然後在一頁中由上往下寫。在 Timed Text (TT) Authoring Format 中，這個模式被簡稱為LRTB^[1]。其它文字，如阿拉伯文字和希伯來語則由右至左書寫。漢字傳統上是縱向書寫（上至下），接著在頁中由右至左排列。但近年來因為西方文化的影響，為了能使用羅馬字母的術語及電子文件的技術上限制，則逐漸出現由左至右，上而下的方式書寫，但傳統寫法在臺灣仍然常見，甚至遷就於無法直寫，以「一字一行」方式，由右向左橫寫的書寫方式也尚未完全消失（例如匾額）。回鶻文字母和其後繼者則是由上至下，左至右的文字；這種方向源自於將祖傳閃語方向90°逆時針方向以便在外觀上適合中文書寫。一些菲律賓和印度尼西亞的文字文字，如Hanunó'o，則是以遠離寫者方向，由下而上的方式書寫。

Obrázek 55 - Čínský dokument

Tabulka 8 - Čínský dokument - OCR+MT

<p>文字 也言午,最主要以围像上厘别束分类直的是辦燒 采泉條審焉系采充是指字符由言午多采泉條所粗 成 ,如拉丁字母和漢宅 漢字如果是用原子筆 毛筆或 金尊刻在青金同器上 則棍擻是腺性的 0 糞直似地 士矣及象形文字和瑠雅文字則通常是 描采含茅泉性外框 但在正式場合 他惘雕刻 在bas-relief中0 另一方面,非采泉性系梳劇如盲 文 不言命是用什顾材寔 都非由采泉條樟成0 最 早 的例子則是采泉條的: 西元前 3300 年的楔形文 字是采泉條的 虽佳然徙它衍生的楔形文字不是 楔形文字也言午是最早的非采泉條文宅 它的字 形是由董董筆尖所匪制在潮淫泥板而成, 而 非之前在泥板上用籒尖描擒腺侃 最後鱒成其文 字的字根外藿兒上白勺圃匕 盲文則是拉丁文字的非茅泉條版本 它完全放萊 拉丁文字的外开魏 字母是由凸瑰所采且成 其基底可以是皮革 潞易斯舫布萊萊原本揉用的 材固, 壁硬塑膠或金風 也有拉丁文字的非腺條的失豆暫性版本 包含摩 雨斯雷礪 手言吾的manual ...atd</p>	<p>Rozpoznávání znaků cak 'Light' (010t Cmtaotet t e n C 0 ohm Ca1 Ko Ko n t nos 0; 0C chrám) je chrám tým soubory tisknout a napsat dopis na win 7 Obraz videa může otrávit tahu ska stroj byl schopen číst charakter-je převést. Černobílé záběry dokumentů, skenování obrázků se může dostat do počítače upravitelný kód znaku apod. Hung-Sik převést software, běžně označované jako 0 C 1 ucho o 0CI (exhibition, je umělá inteligence jsem Stroje A 1 I ' Park (m C S nosem "nos pocit ne 11/0n) oblastech výzkumu: 7 Schön byl I: ~. Zrcadlo nebo čočky, optické rozpoznávání znaků pomocí optické technologie, l skeneru a vědět Lee, jeden z žlučovody, jako například Rozpoznávání digitální znaků jsou myšlenka k jiné oblasti se žádná ~ nyní je optické rub uznání řekněme Digitální rozeznávání a čtyři síly-byl považován za d rez Původní systém je že konkrétní písmo pro čtení je čtení ukázkou písma předem znamená "Tre 0 Ning" Tento Perl by stálo Jimin Gibb teď nejvíce změnil písmo až k oceánu, jako vysoká míra Quark-ye ~ d ~, některé ...atd</p>
---	--

광학 문자 인식(Optical character recognition; OCR)은 사람이 쓰거나 기계로 인쇄한 문자의 영상을 이미지 스캐너로 획득하여 기계가 읽을 수 있는 문자로 변환하는 것이다.

이미지 스캔으로 얻을 수 있는 문서의 활자 영상을 컴퓨터가 편집 가능한 문자코드 등의 형식으로 변환하는 소프트웨어로써 일반적으로 OCR이라고 하며, OCR은 인공지능이나 기계 시각(machine vision)의 연구분야로 시작되었다.

거울이나 렌즈 등의 광학 기술을 이용한 광학 문자 인식과 스캐너 및 알고리즘에 의한 디지털 문자 인식은 다른 영역으로 생각되었으나 이제는 광학 문자 인식이라는 말이 디지털 문자 인식을 포함하는 것으로 간주되었다.

초기 시스템은 특정한 서체를 읽기 위해 미리 해당 서체의 샘플을 읽는 것을 뜻하는 "트레이닝"이 필요했지만, 지금은 대부분의 서체를 높은 확률로 변환이 가능하다. 몇몇 시스템에서는 읽어들이는 이미지에서 그것과 거의 일치하는 워드 프로세서 파일과 같은 문서 포맷으로 된 출력 파일을 생성할 수 있으며, 그 중에는 이미지처럼 문서 이외의 부분이 포함되어있어도 제대로 인식하는 것도 있다.

Obrázek 56 - Korejský dokument

Tabulka 9 - Korejský dokument - OCR+MT

<p>'광' 각 문자 인식(010t코Ca1 Cmtaotet teC0음n코t 코0n; 0C묘)은 사뮌파이 쓰거나 7 계 로 인쇄 한 문자의 영상을 이미지 스캐너로 획득하여 기계가 읽을 수 있는 문자-로 변환하는 것이다. 이미지 스캔으로 얻을 수 있는 문서의 활자 영상을 컴퓨터가 편집 가능한 문자-코드 등의 형식으로 변환하는 소프트웨어로써 일반적으로 OCR이라고 하며, OCR은 인공지능이나 기계 시각(machine vision)의 연구분야로 시작되었다. 거울이나 렌즈 등의 광학 기술을 이용한 광학 문자 인식과 스캐너 및 알고리즘에 의한 디지털 문자 인식은 다른 영역으로 생각되었으나 이제는 광학 문자- 인식이라는 말이 디지털 문자 인식을 포함-하는 것으로 간주되었다. 초기 시스템은 특정한 서체를 읽기 위해 미리 해당 서체의 샘플을 읽는 것을 뜻하는 "트레이닝"이 필요했지만 지금은 대부분의 서체를 높은 확률로 변환이 가능-하다, 몇몇 시스템에서는 물-줄어들인 이미지에서 그것과 거의 일치-하는 워드 프로세서 파일과 같은 문서 포맷으로 된 출력 파-일을 생성할 수 있으며, 그 중에는 이미지처럼 문서 이외의 부분이 포함되어 있-어도 제대로 인식하-는 것도 못-처다.</p>	<p>Rozpoznávání znaků jak 'Light' (010t Cmtaotet te C0 ohm Ca1 Ko Ko n t nos 0; 0C chrám) je chrám tým soubory tisknout a napsat dopis na win 7 Obraz videa může otrávit tahu ska stroj byl schopen číst charakter-je převést. Černobílé záběry dokumentů, skenování obrázků se může dostat do počítače upravitelný kód znaku apod. Hung-Sik převést software, běžně označované jako 0 C 1 ucho o 0CI <exhibition, je umělá inteligence jsem Stroje A 1 I ' Park (m C S nosem "nos pocit ne 11/0n) oblastech výzkumu: 7 Schön byl I: ~. Zrcadlo nebo čočky, optické rozpoznávání znaků pomocí optické technologie, 1 skeneru a vědět Lee, jeden z žlučovody, jako například Rozpoznávání digitální znaků jsou myšlenka k jiné oblasti se žádá ~ nyní je optické rub uznání řekněme Digitální rozeznávání a čtyři síly-byly považován za d rez Původní systém je že konkrétní písmo pro čtení je čtení ukázkou písma předem znamená "Tre 0 Ning" Tento Perl by stálo Jimin Gibb teď nejvíce změnil písmo až k oceánu, jako vysoká míra Quark-ye ~ d ~, některé Systém využívá výkres - Jules byl stejně krotčí Edie 7 Deere Téměř odpovídá dolní a textový editor souborů jako například Výstupní formát dokumentu byl schopen generovat krve práce, než dokument jako obrázek, z nich Část 0 Rozpoznán, záznamník/horký vzduch také součástí DNA-je také pohledem na palci.</p>
---	---

تصنيع الورق آليا : هناك طريقتان لصناعة العجينة الورقية : ميكانيكية وكيميائية.

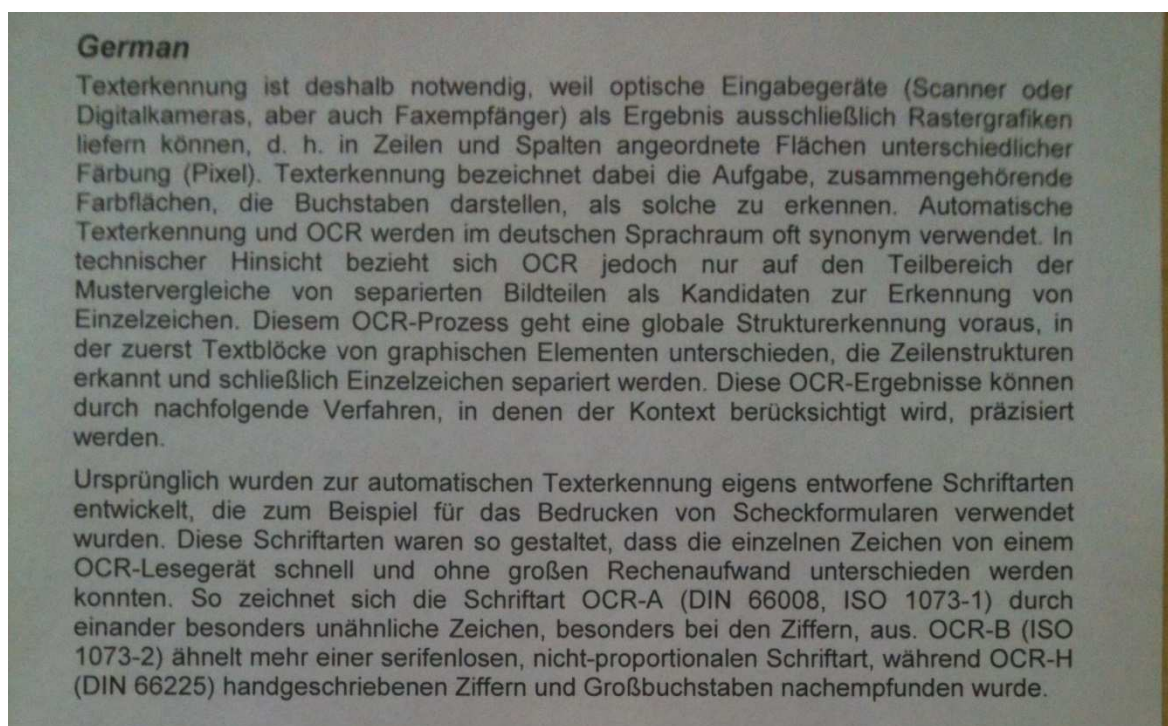
- عند صناعة الورق آليا ينظف اللحاء المستخدم باستخدام الماكينة من أجل التخلص من الغبار أو الرماد والمواد الغريبة.
- وبعد عملية التنظيف هذه، يوضع اللحاء في غلاية دائرية كبيرة حيث يغلي اللحاء والجير تحت ضغط البخار لمدة تصل إلى عدة ساعات.
- يتحد الجير مع الدهون والمواد الغريبة الأخرى الموجودة في اللحاء ليكون صابونا غير قابل للذوبان، ويمكن التخلص من هذا الصابون فيما بعد، كما أن الجير يقلل أية صبغة ملونة موجودة في المركبات الملونة.
- ثم يحول اللحاء إلى ماكينة تسمى هولاندر بحيث تشكل سلسلة متصلة.
- ويمر الخليط المكون من اللحاء والمياه بين الأسطوانة ولوح القاعدة ويتحول اللحاء إلى ألياف، بحيث تمتص المياه من الحوض تاركة اللحاء والألياف خلفها.
- وأثناء تدفق خليط اللحاء والمياه حول الهولاندر، يتم التخلص من القاذورات وينقع اللحاء تدريجيا حتى يتحلل تماما إلى ألياف مفردة.
- وبعد ذلك يتم إدخال اللحاء المبتل في ماكينة هولاندر فرعية من أجل فصل الألياف مرة أخرى.
- وعند هذه النقطة، تضاف مواد تلوين ومواد غراء كالصمغ ومواد حشو مثل كبريتات الجير، وذلك لزيادة حجم ووزن الورق.

Obrázek 57 - Arabský dokument

Tabulka 10 - Arabský dokument - OCR+MT

<p>م الورقية العهنة لصناعة طريقتان هناك ب آليا الورق تصنيع وكيمهمانية ميكانيكية</p> <ul style="list-style-type: none"> • الماكينة باستخدام المستخدم اللحاء ينظف اليا الورق صناعة عند الغريبة والمواد الرماد أو الغيار من التخلص أجل من كبيرة دائرية غلاية في اللحاء يوضع هذه، التنظيف عملية وبعد . إلى تصل لمدة البخار ضغط تحت والقهر اللحاء يغلي حيث ساعات عدة • في الموجودة الأخببي الغربية والمواد للدهون مع الجهر ليتحد . من التخلص ويمكن للذوبان، قابل تجر صالونا ليكون اللحاء موجودة ملونة صهفة اية يقلل الجهر ان كما بعد، فهما الصابون هذا الملونة المركبات في • سلسلة تشكل بحيث هولاندر تسمى ماكينة إلى اللحاء يحول ثم متصلة • القاعدة ولوج الانطواين بهن والمياه اللحاء من المكون الخليط ويمر . من المياه تمتص بحيث الهاف، إلى اللحاء ويتحول خلفها والالياف اللحاء نناركة الحوهس • من التخلص يتم الهولاندر، حول والمياه اللحاء خليط تدفق وأثناء . ألياف إلى تماما يتحلل حلى تدريجيا اللحاء ولينقع القاذورات مفردة • من فرعية هولاندر ماكينة في المهتل اللحاء إدخال ليتم ذلهم وبعد . أخرى مرة الألياف فصل اجل • ومولد كظصمغ غراء ومراد تلوين مولد تضاف النقطة، هذه وعند . الورق ووزن حجم لزيادة وذلك الجهر كبريتات مثل حشر 	<p>Papírové dělat stroj b tam jsou dva způsoby, jak vyrobit papír, mechanické alahhnh wekimhmanih.</p> <p>0 když papír automaticky čistí, uživatel kůra přístrojem zbavit náhradní nebo popela a západní materiálů</p> <p>. Po vyčištění umístěte kůra do konvice velké kruhové kde vroucí kůry a útlaku pod tlak vodních par do</p> <p>Několik hodin.</p> <p>. Kombinovat hlas tuku a západní alakhbbi materiálem v mozkové kůry Salon je llzaweyan, a mohou být odstraněny</p> <p>Jsou to mýdlo, a hlas snižuje libovolnou barvu v shghah barevné sloučeniny</p> <p>0 pak převede kůra na stroj s názvem Hollander tvořit kontinuum.</p> <p>. Předávání směs kůry a voda je alantawain do Al-Kajdy zapne allkhaa do alhaf, který nasává vodu z Alhohs nmarkh lýková vlákna.</p> <p>. A zatímco tok vody po celém alholandz směs kůry se zbavit špíny a namočí lýkových postupně odbourává helli vlákna</p> <p>Sám.</p> <p>. Po zellhm vstoupit cortex almhtl holandz sub stroje k oddělení vlákno znovu.</p> <p>0 v tomto okamžiku přidat barvy a Murat generátor kzsamgh generátor napěchované lepidlo jako síran hlas ke zvýšení velikosti a hmotnosti papíru.</p>
---	--

Jazyky – Evropa



Obrázek 58 - Německý dokument /2/

Tabulka 11 - Německý dokument /2/ - OCR+MT

<p>German g</p> <p>Texterkennung ist deshalb notwendig, weil optische Eingabegeräte (Scanner oder Digitalkameras, aber auch Faxempfänger) als Ergebnis ausschließlich Rastergrafiken liefern können, d h in Zeilen und Spalten angeordnete Flächen unterschiedlicher Färbung (Pixel), Texterkennung bezeichnet dabei die Aufgabe, zusammengehörende Farbflächen, die Buchstaben darstellen, als solche zu erkennen. Automatische Texterkennung und OCR werden im deutschen Sprachraum oft synonym verwendet In technischer Hinsicht bezieht sich OCR jedoch nur auf den Teilbereich der Mustervergleiche von separierten Bildteilen als Kandidaten zur Erkennung von Einzelzeichen. Diesem OCR-Prozess geht eine globale Strukturerkennung voraus, in der zuerst Textblöcke von graphischen Elementen unterschieden, die Zeilenstrukturen erkannt und schließlich Einzelzeichen separiert werden, Diese OCR-Ergebnisse können durch nachfolgende Verfahren, in denen der Kontext berücksichtigt wird, präzisiert werden Ursprünglich wurden zur automatischen Texterkennung eigens entworfene Schriftarten entwickelt, die zum Beispiel für das Bedrucken von Scheckformularen verwendet wurden. Diese Schriftarten waren so gestaltet, dass die einzelnen Zeichen von einem OCR-Lesegerät schnell und ohne großen Rechenaufwand ...atd.</p>	<p>Německý g</p> <p>Rozpoznávání textu je nezbytná. Protože optické vstupní zařízení (skenery nebo digitální fotoaparáty, ale i Faxempfänger) z důvodu zásobování může jen Rastergrafiken, d h v řádcích a sloupcích na různé povrchy</p> <p>Toto, barvivo (pixely), prediktivní psaní textu označuje úkol zjistit zusammengehörende Farbflächen představují písmena, jako takové. Automatický text uznání a OCR jsou často používány zaměnitelně v německy mluvícím světě</p> <p>In</p> <p>OCR technicky, ale jen odkazuje na část vzoru odpovídajících částí oddělených obraz jako kandidáty pro detekci jednotlivých znaků. Celosvětové uznání struktury je před OCR procesem, kde je lze nejprve rozlišit bloků textu od grafických prvků, poznal liniových staveb a nakonec odděleny znak, tato OCR jsou výsledky prostřednictvím následujících postupů, která berou v úvahu kontext, objasňuje původně speciálně navržený písma pro rozpoznávání textu automatické, používané například pro tisk šekové tiskopisy byli. Tato písma jsou navrženy tak, aby každý znak OCR-Reader by mohla být rychle a nterschieden bez výpočetně složitý. Tak je charakterizována písma OCR-A (DIN 66008, ISO 1073-1)</p> <p>atd.</p>
--	---

English

In 1929 Gustav Tauschek obtained a patent on OCR in Germany, followed by Handel who obtained a US patent on OCR in USA in 1933 (U.S. Patent 1,915,993). In 1935 Tauschek was also granted a US patent on his method (U.S. Patent 2,026,329). Tauschek's machine was a mechanical device that used templates and a photodetector.

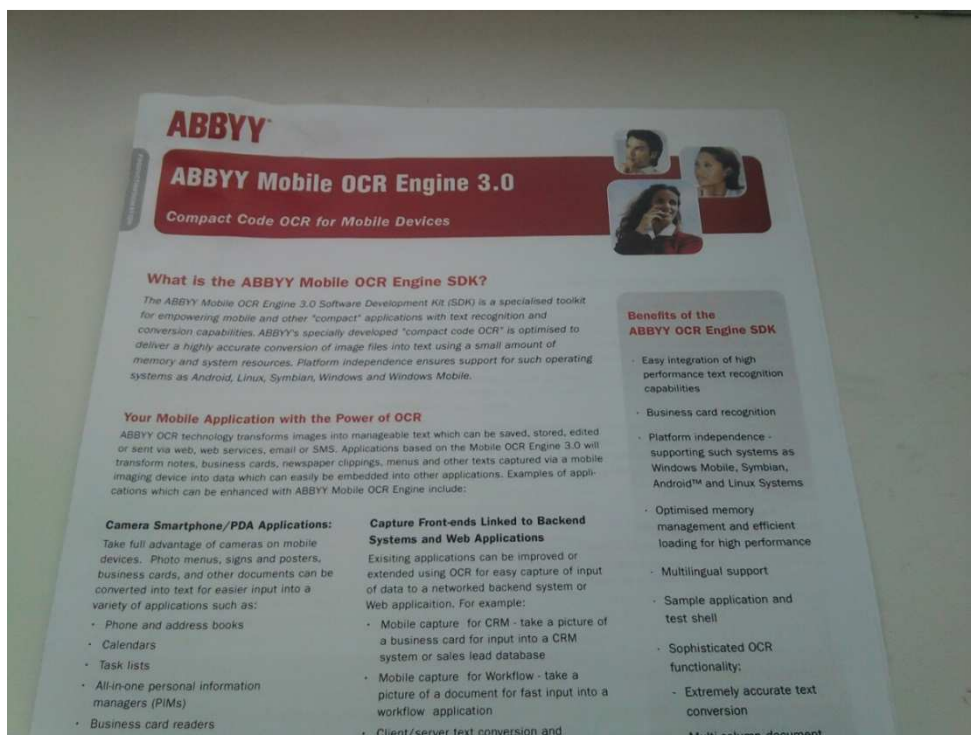
In 1950, David H. Shepard, a cryptanalyst at the Armed Forces Security Agency in the United States, addressed the problem of converting printed messages into machine language for computer processing and built a machine to do this, reported in the Washington Daily News on 27 April 1951 and in the New York Times on 26 December 1953 after his U.S. Patent 2,663,758 was issued. Shepard then founded Intelligent Machines Research Corporation (IMR), which went on to deliver the world's first several OCR systems used in commercial operation.

The first commercial system was installed at the Reader's Digest in 1955. The second system was sold to the Standard Oil Company for reading credit card imprints for billing purposes. Other systems sold by IMR during the late 1950s included a bill stub reader to the Ohio Bell Telephone Company and a page scanner to the United States Air Force for reading and transmitting by teletype typewritten messages. IBM and others were later licensed on Shepard's OCR patents.

Obrázek 59 - Anglický dokument /1/

Tabulka 12 - Anglický dokument /1/ - OCR+MT

<p>English</p> <p>In 1929 Gustav Tauschek obtained a patent on OCR in Germany, followed by Handel who obtained a US patent on OCR in USA in 1933 (U.S. Patent 1,915,993). In 1935 Tauschek was also granted a US patent on his method (U.S. Patent 2.026.329). Tauschek's machine was a mechanical device that used templates and a photodetector. in 1950, David H. Shepard, a cryptanalyst at the Armed Forces Security Agency in the United States, addressed the problem of converting printed messages into machine language for computer processing and built a machine to do this, reported in the Washington Daily News on 27 April 1951 and in the New York Times on 26 December 1953 after his U.S. Patent 2,663,758 was issued. Shepard then founded Intelligent Machines Research Corporation (IMR), which went on to deliver the world's first several OCR systems used in commercial operation. The first commercial system was installed at the Reader's Digest in 1955. The second system was sold to the Standard Oil Company for reading credit card imprints for billing purposes. Other systems sold by IMR during the late 1950s included a bill stub reader to the Ohio Bell Telephone Company and a page scanner to the United States Air Force for reading and transmitting by teletype typewritten messages. IBM and others were later licensed on Shepard's OCR patents.</p>	<p>Angličtina</p> <p>V roce 1929 Gustav Tauschek získal patent na OCR v Německu, následovaný Handel, který získal americký patent na OCR v USA v roce 1933 (US Patent 1,915,993). V roce 1935 Tauschek byl také udělen US patent na jeho metodu (US Patent 2.026.329). Mechanické zařízení, které používají šablony a fotosnímač byl stroj je Tauschek. v roce 1950, David H. Shepard, kryptoanalytik v bezpečnostní agentuře ozbrojených sil ve Spojených státech, řeší problém převodu tištěných zpráv do strojového jazyka pro počítačové zpracování a stroj k tomu, oznámil ve Washingtonu Daily News, 27. dubna 1951 a v New York Times 26 prosince 1953 po jeho US Patent 2,663,758 byl vydán. Shepard pak založil inteligentní stroje Research Corporation (IMR), který pokračoval doručit na světě první několik OCR systémy používané v komerčním provozu. První komerční systém byl instalován na Reader's Digest v roce 1955. Druhý systém byl prodán společnosti Standard Oil pro čtení kreditní kartu otisky pro účely fakturace. Ostatní systémy nabízené IMR během pozdních padesátých let zahrnoval čtečku zákona inzerováním Ohio Bell telefonní společnosti a stránky skeneru pro letectvo Spojených států pro čtení a přenos zpráv dálnopis stroji. IBM a další byli později licence na patenty OCR Shepardovi.</p>
--	--



Obrázek 60 - Anglický dokument /2/

Tabulka 13 - Anglický dokument /2/ - OCR+MT

<p>.5 i i .>), "7i3,~:-i'-.-l:=<Li<i 1: >'),,i. 1.3 . . ; 1 7; 3;~:.' i} "l:..ll:'£'ial,"'flvifil-?'V7?'uiç'3'T ~,~*':.'~'j:=-'»j i=l=:::=-F: i ' ~ . ~ - 7' ii ~ :- "I, 'b"l'€ill"" 1. l. ' l - ssvv —*<~:,A, 7 =55. *1 - i e 'll, ' ~.~ . - i. I ' '§ ,{ W "—,"'~'~{n:'.~,'i;,'fi\$%l-Tr= =:."i.i'j:.*' V I ' ' as "" ,5; 5;: ABBYY Mfhl I 8 OCR Engine 3 . 0 i : i * ' .« ~ ' i ' y'£"4": 4Z::'v ?V" . i: i r 5 i Compact c - . 2 . ii ode OCR for Mobile Devices ' "</p>	<p>.5 já i .»Já, "7i3, ~:-jsem '--okýnka: = <.Li «já 1.:>'),, i. 1.3...; 1 7; 3, '~:.' i} "l:..LL: '£' rú, ""flvifil- ?"V7?" 'UI ç'3 'T ~, ~*':."~'j:=-'»j i=l=:::=-F: i ' ~ . . ~-7' ii ~:-"já," b "l' €nemocný" 1. l. l - ssvv — * < ~:, A, 7 = 55. *1 - já ti e, ' ~. ~.-já. I ' '§ ,{ W "—,"'~'~{n:'.~,'i;,'fi\$ %l-Tr=: "i.i'j:.*' V I ' ' jako" "" ,5; 5;: ABBYY Mfhl I 8 OCR Engine 3 . 0 i : i * ' «~ ' i ' y' £"4" " : 4Z::'v?V" . i: jsem r 5 komprimace c - . 2. II ode OCR pro mobilní zařízení</p>
<p>i l s i What is the ABBYY Mobile OCR Engine SDK? i The ASBH' Mobile OCR Engine 3.0 Software Development Kit (SDK) is a specialised toolkit > V ' ' ,7 fo' e'77130W5flllg mobile and other "coiiipact" applications with text recognition and senengs of the I V ' i / conversion capabilities. ABBYY's specially developed "compact code OCR" is optimised to ABBYY OCR Engne SDK . ' 3 deliver a highly accurate conversion of image files into text using a small amount of ' I , "1;""~""0/y and system resources. Platform independence nsures support for such operating . Easy Imegrallon M "lg, -l' ,1: Systems as Android, Linux, Symbian, Windows and Windows Mobile. performance text reoogitltlon ml ,i,___ .54?capabilities ' _ Your Mobile Application with the Power of OCR - Business card recognition 9 * '5ABBYY OCR technologie transforms images into manageable text which can be saved, stored. ediied _ ...atd.</p>	<p>Já l s co je ABBYY Mobile OCR Engine SDK? jsem The ASBH' Mobile OCR Engine 3.0 Software Development Kit (SDK) je specializované toolkit > V ' ' ,7 fo' e'77130W5flllg mobilních a jiných "coiiipact" aplikace s rozpoznávání textu a senengs i V ' já / možnosti převodu. Společnost ABBYY je speciálně vyvinutý "kompaktní OCR kódu" je optimalizována na ABBYY OCR Engne SDK. "3 dodat vysoce přesné konverzi obrazových souborů do textu pomocí malé množství", "1;"" ~""0/y a systémových prostředků. Platformní nezávislosti nsures podpora pro takový provoz. Snadné Imegrallon M "lg,-l' ,1: systémy jako Android, Symbian, Windows Linux a Windows Mobile. výkon textu reoogitltlon ml, já, ___ .54? _ schopnosti Your mobilní aplikaci s moc OCR - vizitka uznání 9 * '5ABBYY OCR technologie transformace obrazů do spravovatelných text, který může být uložen, uložen. ediied _ ...atd.</p>

French

La première machine de ROC fut créée par Gustav Tauschek, un ingénieur allemand, en 1929. Elle contenait un détecteur photosensible qui pointait une lumière sur un mot lorsqu'il correspondait à un gabarit contenu dans sa mémoire.

En 1950, Frank Rowlett, qui avait cassé le code diplomatique japonais PURPLE, demanda à David Shepard, un cryptanalyste de l'AFSA (prédécesseur de la NSA américaine), de travailler avec Louis Tordella pour faire à l'agence des propositions de procédures d'automatisation des données. La question incluait le problème de la conversion de messages imprimés en langage machine pour le traitement informatique. Shepard décida qu'il devait être possible de construire une machine pour le faire, et, avec l'aide de Harvey Cook, un ami, construisit « Gismo » dans son grenier pendant ses soirées et ses week-ends. Le fait fut rapporté dans le Washington Daily News du 27 avril 1951 et dans le New York Times du 26 décembre 1953 après le dépôt du brevet numéro 2 663 758. Shepard fonda alors Intelligent Machines Research Corporation (IMR), qui livra les premiers systèmes de ROC au monde exploités par des sociétés privées. Le premier système privé fut installé au Reader's Digest en 1955, et, de nombreuses années plus tard, fut offert par le Readers Digest au Smithsonian, où il fut mis en exposition. Les autres systèmes vendus par IMR à la fin des années 1950 comprenaient un lecteur de bordereau de facturation à l'Ohio Bell Telephone Company et un numériseur (scanner de documents) à l'US Air Force pour la lecture et la transmission par télex de messages dactylographiés. IBM et d'autres utilisèrent plus tard les brevets de Shepard.

Obrázek 61 - Francouzský dokument

Tabulka 14 - Francouzský dokument - OCR+MT

<p>French</p> <p>La première machine de ROC fut créée par Gustav Tauschek, un ingénieur allemand, en 1929, Elle contenait un détecteur photosensible qui pointait une lumière sur un mot lorsqu'il correspondait à un gabarit contenu dans sa mémoire. En 1950, Frank Rowlett, qui avait cassé le code diplomatique japonais PURPLE, demanda à David Shepard, un cryptanalyste de VAFSA (prédécesseur de la NSA américaine), de travailler avec Louis Tordella pour faire à l'agence des propositions de procédures d'automatisation des données. La question incluait le problème de la conversion de messages imprimés en langage machine pour le traitement informatique. Shepard décida qu'il devait être possible de construire une machine pour le faire, et, avec l'aide de Harvey Cook, un ami, construisit « Gismo » dans son grenier pendant ses soirées et ses week-ends. Le fait fut rapporté dans le Washington Daily News du 27 avril 1951 et dans le New York Times du 26 décembre 1953 après le dépôt du brevet numéro 2 663 758. Shepard fonda alors Intelligent Machines Research Corporation (IMR), qui livra les premiers systèmes de ROC au monde exploités par des sociétés privées. Le premier système privé fut installé au Reader's Digest en 1955, et, de nombreuses années plus tard, fut offert par le Readers Digest au Smithsonian, où il fut mis en exposition. Les autres systèmes vendus par IMR à la fin des années 1950 comprenaient un lecteur atd.</p>	<p>Francouzština</p> <p>První rocková stroj vytvořil Gustav Tauschek, německý inženýr, v roce 1929, obsahoval fotosenzitivní detektor, který ukázala světlo na lorsquii, kterou slovo odpovídal na šabloně obsažené v jeho paměti. V roce 1950, Frank Rowlett, který se vloupal japonské diplomatické kód fialová, zeptal se David Shepard, kryptoanalytik VAFSA (předchůdce americká NSA), práce s Louis Tordella předkládat návrhy na postupy Informationen údaje agentů. Otázka součástí problému převodu zpráv v jazyka stroje pro počítačové zpracování. Shepard se rozhodl, že by mělo být možné sestavit stroj, a, s pomocí Harvey Cook, přítel, postavený "věcička » v podkrovní během večery a víkendy. Skutečnost byla hlášena ve Washingtonu Daily News 27 dubna 1951 a New York Times 26. prosince 1953, po podání patentové číslo 663 2758. Shepard založil pak inteligentní stroje Research Corporation (IMR), který vydal první ROC systémů na světě provozovány soukromými společnostmi. První soukromý systém byl nainstalován v Reader's Digest v roce 1955, a. O mnoho let později, byl předložen čtenáři Digest v Smithsonian, kde byl kladen na displeji. Ostatní systémy nabízené IMR během pozdních padesátých let zahrnutý skluzu jsem vyúčtování čtenář ' Ohio Bell telefonní společnost a skener (skener) do amerického letectva pro čtení a přenos zpráv dálnopis napsaný. IBM atd.</p>
---	--

Italian

Il sistema postale degli Stati Uniti d'America utilizza sistemi di OCR fin dal 1965. La necessità di riconoscere le destinazioni delle missive e di organizzarle in modo automatico ha spinto la ricerca nel settore dell'OCR. I sistemi OCR leggono il codice postale scritto sulle lettere e provvedono a stampare sulle missive un codice a barre che rappresenta la destinazione della lettera. Per impedire che il codice a barre disturbi la lettura dell'indirizzo e quindi complichino il lavoro dei postini il codice a barre viene stampato con un inchiostro visibile solo se illuminato da una luce con lunghezza d'onda nell'ultravioletto. Il codice a barre viene utilizzato da macchine smistatrici per indirizzare la corrispondenza all'ufficio postale corrispondente che si occuperà di recapitarlo al destinatario. Un metodo analogo è in uso dalle Poste Italiane per la gestione della corrispondenza.

Obrázek 62 - Italský dokument

Tabulka 15 - Italský dokument - OCR+MT

Italian	Italština
<p>Il sistema postale degli Stati Uniti d'America utilizza sistemi di OCR fin dal 1965. La necessità di riconoscere le destinazioni delle missive e di organizzarle in modo automatico ha spinto la ricerca nel settore dell'OCR. I sistemi OCR leggono il codice postale scritto sulle lettere e provvedono a stampare sulle missive un codice a barre</p> <p>che rappresenta la destinazione della lettera. Per impedire che il codice a barre disturbi</p> <p>la lettura dell'indirizzo e quindi complichino il lavoro dei postini il codice a barre viene stampato con un inchiostro visibile solo se illuminato da una luce con lunghezza d'onda nell'ultravioletto. Il codice a barre viene utilizzato da macchine smistatrici per indirizzare</p> <p>la corrispondenza all'ufficio postale corrispondente che si occuperà di recapitarlo al destinatario. Un metodo analogo è in uso dalle Poste Italiane per la gestione della corrispondenza.</p>	<p>poštovní systém Spojených států amerických, pomocí OCR systémy od roku 1965. V Potřebujete identifikovat cíle písmen a uspořádat tak automaticky tlačeného výzkumu v deliOCR. OCR systémy číst kód Písenná na poštovní dopisy a tisk na dopis medikovanými krmení čárový kód</p> <p>To je cíl, dopisu. Chcete-li zabránit čárový kód poruchy</p> <p>čtení delfindinzzo a tudíž komplikují práci pošťáků, čárový kód je potišťené inkoustem, který je viditelný, když se rozsvítí světlo s vlnovou délkou pouze Neli ' ultravioletto. Čárový kód používá smistatrici do cílových počítačů</p> <p>poštovní aliufficio odpovídající shody, která se bude bát z recapitario na příjemce. Podobná metoda se používá v Poste italiane pro řízení korespondence.</p>

Spanish

Todos los algoritmos de Reconocimiento Óptico de Caracteres tienen la finalidad de poder diferenciar un texto de una imagen cualquiera. Para hacerlo se basan en 4 etapas: Binarización, Fragmentación o segmentación de la imagen, Adelgazamiento de las componentes y Comparación con patrones.

Binarización

La mayor parte de algoritmos de OCR parten como base de una imagen binaria (dos colores) por lo tanto es conveniente convertir una imagen de escala de grises, o una de color, en una imagen en blanco y negro, de tal forma que se preserven las propiedades esenciales de la imagen. Una forma de hacerlo es mediante el histograma de la imagen donde se muestra el número de píxeles para cada nivel de grises que aparece a la imagen. Para binarizarla tenemos que escoger un umbral adecuado, a partir del cual todos los píxeles que no lo superen se convertirán en negro y el resto en blanco.

Obrázek 63 - Španělský dokument

Tabulka 16 - Španělský dokument – OCR+MT

<p>Spanish</p> <p>Todos los algoritmos de Reconocimiento Óptico de Caracteres tienen la finalidad de poder diferenciar un texto de una imagen cualquiera. Para hacerlo se basan en 4 etapas: Binarización, Fragmentación o segmentación de la imagen, Adelgazamiento de las componentes y Comparación con patrones.</p> <p>Binarización</p> <p>La mayor parte de algoritmos de OCR parten como base de una imagen binaria (dos colores) por lo tanto es conveniente convertir una imagen de escala de grises, o una de color, en una imagen en blanco y negro, de tal forma que se preserven las propiedades esenciales de la imagen. Una forma de hacerlo es mediante el histograma de la imagen donde se muestra el número de píxeles para cada nivel de grises que aparece a la imagen. Para binarizarla tenemos que escoger un umbral adecuado, a partir del cual todos los píxeles que no lo superen se convertirán en negro y el resto en blanco.</p>	<p>Španělština</p> <p>Všechny algoritmy pro optické rozpoznávání znaků naplnit účel k odlišení textu z libovolného obrázku k tomu vycházejí 4 fáze: Binarizace, Fragmentace nebo segmentace obrazu, řídící z komponenty a vzory ve srovnání.</p> <p>Binarizace</p> <p>Většina OCR algoritmy odjíždí jako obrázek binaria (dvě základní barvy) je tedy vhodné převést obraz ve stupních šedi, nebo jeden z barva v černé a bílé image. Taková forma, která jsou zachovalé vlastnosti základní obrázek. Způsob, jak to provést je pomocí histogramu obrázku. Zobrazuje počet pixelů pro jednotlivé úrovně šedi, která se zobrazí a obrázek. Pro binarizaci budeme muset zvolit vhodnou prahovou hodnotu, z něhož všechny pixely nepřekračující to stane černá a se odpočívá v bílém.</p>
---	--

Komplex je zpravodajská jednotka firmy. Právě jim jsem volal svou víceméně planou výhrůžku. Bylo mi jasné, že mé poznámky si vyloží jako informace, jimiž může disponovat jedině jejich bývalý zaměstnanec, a rychle si zjistí jeho základní data. Uspokojovalo mě, že je mohu oslovovat jmény, že vím, jak se jmenovala něčí matka za svobodna, že jeho mohu takto omráčit, do detailu jim popsat jejich postupy. Měli mě v hlavě, spojení bylo navázáno. Nemusel jsem to břímě nést sám.

Obrázek 64 - Český dokument /2/

Tabulka 17 - Český dokument /2/ - OCR

<p>Komplvx je Zpravodijská jednotka firmy. Právě jim jsem volal Svou víreméně planou výhrůžku. Bylo mi jasné, že mé poznánikj Si vyloží jalxo informace, jimiž může disponovat jedině jejich bývalý Zaměstnanoo, a ryřř-lilv si Zjistíjeho Základní (lata. Uspokojovalo mě, že je mohu oslovovat jmény, že vím, se jmenovala něčí matlša Za svobodou. že jeho mohu takto omráčit, do (lctaílu jim jiripsatjejich postupy. Älěli mě v hlavě, Spojøní bylo navázáno. \\(nnuS<l jsem to liřřinu“ nést sám.</p>

Пример текста. Язык документа –русский. | 2011

Китайские летописи сообщают, что бумага была изобретена в 105 году н. э. Цай Лунем (см. четыре великих изобретения)^[1]. Однако в 1957 году в пещере Баоця северной провинции Китая Шаньси обнаружена гробница, где были найдены обрывки листов бумаги. Бумагу исследовали и установили, что она была изготовлена во II веке до нашей эры.

До Цай Луня бумагу в Китае делали из пеньки, а ещё раньше из шёлка, который изготавливали из бракованных коконов шелкопряда.

Цай Лунь растолок волокна шелковицы, древесную золу, тряпки и пеньку. Всё это он смешал с водой и получившуюся массу выложил на форму (деревянная рама и сито из бамбука). После сушки на солнце, он эту массу разгладил с помощью камней. В результате получились прочные листы бумаги.

Obrázek 65 - Ruský dokument

Tabulka 18 - Ruský dokument - OCR+MT

<p>Пример текста. Язык документа -русский. 201 ĭ</p> <p>Китайские летописи сообщают, что бумага была изобретена в 105 году н. э. Ц ай Лунем (см. четыре великих изобретенищ. Однако в 1957 го ду в пещере Баоця северной провинции Китая Шаньси обнаружена гробница, где были найдены обрывки листов бумаги. Бумагу исследовали и установили, что она была изготовлена во II веке до нашей эры.</p> <p>До Цай Луня бумагу в Китае делали из пеньки, а ещё раньше из шёлка, который изготавливали из бракованны1х коконов шелкопряда.</p> <p>Цай Лунь растолок волокна шелковицы, древесную Ёу, тряпки и пеньку. Всё это он смешал с водой и получившуюся массу выложил на форму (деревянная рама и сито из бамбука). После сушки на солнце, он эту массу разгладил с помощью камней. В результате получились прочные листы бумаги.</p>	<p>Přříklad textu. Dokument jazyk Ruština. 201-</p> <p>Čínské kroniky hlásí, že papír byl vynalezen v 105 a.d. u Ay Lunem (viz čtyři skvělé izobretenišš., v roce 1957 v jeskyni na severním Baocâ Čínská provincie Shanxi byla nalezena v hrobce, kde našli kousky papíru. Papír prošetřena a zjistil, že byl vyroben v p století př.</p> <p>Před Cai Lun papír v Číně byly vyrobeny z konopí a hedvábí, který býval izgotavlivali zámotků bource brakovann'lh.</p> <p>Cai Lun rastolok Mulberry vlákná, dřevo, Ěu, hadry a konopí. To vše se smíchaný s vodou a výsledná hmotnost transparent na formuláři (dřevěný rám a síto z bambus). Po sušení na slunci, uhladil ji přes tuto hmotnost kamenů. Dopad byly to pevné listy papíru.</p>
--	--

B Obsah příloženého DVD

- Text diplomové práce ve formátu PDF
- Implementovaná aplikace
- Zdrojové kódy
- Obrázky pro rozpoznávání textu