

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MOBILNÁ APLIKÁCIA NA GENEROVANIE PECS
KARTIČIEK
BAKALÁRSKA PRÁCA

2023

JAKUB KRIŠTOF

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MOBILNÁ APLIKÁCIA NA GENEROVANIE PECS
KARTIČIEK
BAKALÁRSKA PRÁCA

Študijný program: Aplikovaná informatika
Študijný odbor: Informatika
Školiteľ: RNDr. Zuzana Berger Haladová, PhD.

Bratislava, 2023
Jakub Krištof



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Jakub Krištof
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Mobilná aplikácia na generovanie PECS kartičiek
Mobile app for PECS cards generation

Anotácia: Deti s poruchou autistického spektra majú problém s porozumením a verbálnou komunikáciou. Systém PECS kartičiek im pomáha lepšie porozumieť a dohodnúť sa. V tejto práci pojde o vytvorenie aplikácie, ktorá umožní rodičom zadávať obrázky a popisky a generovať pdf s kartičkami pripravenými na tlač a laminovanie. Aplikácia môže byť pre Android alebo IOS a bude využívať algoritmy počítačového videnia pre efektívny design kartičiek.

Vedúci: RNDr. Zuzana Berger Haladová, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 25.05.2022

Dátum schválenia: 17.10.2022

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestné prehlásenie: Čestne prehlasujem, že som túto prácu vypracovala samostatne iba s použitím uvedených zdrojov a na základe konzultácií so školiteľom a konzultantkou.

.....

Pod'akovanie: TBA

.....

Abstrakt

Hlavný cieľ bakalárskej práce spočíval vo vytvorení Android aplikácie na generovanie PECS kartičiek, slúžiacich na zjednodušenie, zefektívnenie a spríjemnenie komunikácie s ľuďmi s či už poruchami autistického spektra, alebo aj s poruchami schopnosti komunikácie. Výsledná aplikácia ponúka viacero funkcionalít. Okrem samotného jadra aplikácie, a to generovania samotných kartičiek, ponúka aj možnosť kartičky používať v rámci mobilného telefónu, možnosť ich exportovať do formátu PDF pre následnú tlač, alebo automatické orezanie vloženého obrázku na základe jeho významných oblastí. Kartičky sa ukladajú priamo do pamäte mobilného zariadenia, a v aplikácii ich bude možné vymazať, prípadne zazdieľať iným užívateľom pomocou sociálnych sietí/Bluetooth. V práci sú opísané využité technológie a algoritmy, no aj samotný proces získania funkcionality na spracovanie obrazu v jazyku Kotlin.

Kľúčové slová: mobilná aplikácia, PECS systém, porucha schopnosti komunikácie, významné oblasti obrázku

Abstract

The main goal of the bachelor's thesis was to create an Android application for generating PECS cards, which serve to simplify, and streamline streamline communication with people with either autism spectrum disorders or communication disorders. The resulting application offers several functionalities. In addition to the very core of the application, namely the generation of the cards themselves, it also offers the possibility of using the cards within a mobile phone, the possibility of exporting them to PDF format for subsequent printing, or automatic cropping of the inserted image based on computed saliency. The cards are stored directly in the memory of the mobile device, and it will be possible to delete them within the application, or share them with other users using social networks/Bluetooth. The work describes the technologies and algorithms used, as well as the actual process of acquiring functionality for image processing in the Kotlin language.

Keywords: mobile application, PECS system, communication impairment, image saliency

Obsah

Úvod	1
1 Úvod do problematiky	3
1.1 Teoretické východiská	3
1.1.1 Poruchy autistického spektra	3
1.1.2 Systémy na učenie komunikácie u ľudí s poruchami autistického spektra	5
1.1.3 Picture Exchange Communication System (PECS)	6
1.1.4 Významné oblasti fotografie	6
1.1.5 Algoritmy na detekciu významných oblastí fotografie	7
1.1.6 Viola-Jones algoritmus, Haarove kaskády	8
1.2 Motivácia	9
1.2.1 Aplikácia pre rodičov	9
1.2.2 Požiadavky pre aplikáciu	10
2 Prehľad existujúcich aplikácií	11
2.0.1 Card Talk	11
2.0.2 Talk in Pictures X - AAC Speak	12
2.0.3 SymboTalk - AAC Talker	13
3 Technológie	15
3.1 Použité knižnice	15
3.1.1 CameraX	15
3.1.2 Room	15
3.1.3 OpenCV	15
3.1.4 Chaquopy	16
3.2 Použité programy	16
3.2.1 Android Studio	16
3.2.2 IntelliJ IDEA	16

4	Návrh riešenia	17
4.1	Užívateľské rozhranie	17
4.1.1	Úvodná obrazovka	17
4.1.2	Návod na použitie aplikácie	18
4.1.3	Domovská stránka	18
4.1.4	Úprava novej kartičky	20
4.1.5	Export kartičiek do formátu PDF	21
4.2	Dátový model	21
4.2.1	Pôvodne zamýšľaný dátový model	22
4.2.2	Výsledný zamýšľaný dátový model	22
4.3	Porovnanie algoritmov na spracovanie obrazu	22
4.4	Material Design 3	23
5	Implementácia	24
5.1	Vytváranie nových kartičiek	24
5.1.1	Použitie kamery mobilného zariadenia	24
5.1.2	Vybratie fotografie z galérie	24
5.2	Automatické orezanie fotografie na báze významných oblastí	25
5.2.1	Natívne implementovanie OpenCV	25
5.2.2	Implementovanie knižnice OpenCV pomocou knižnice JavaCPP	25
5.2.3	Implementovanie knižnice OpenCV pomocou knižnice chaquopy	26
5.2.4	Spracovanie významných oblastí obrázka	26
5.2.5	Rozpoznávanie tvárí v obrázku	28
5.2.6	Zjednotenie výsledkov spracovania obrazu	29
5.3	Export kartičiek do formátu PDF	29
5.3.1	Vytvorenie PDF súboru	29
5.3.2	Uloženie obrázkov do PDF súboru	29
5.4	Testovanie	30
5.4.1	Testovací scenár	30
5.4.2	Výsledky testovania	30
	Záver	32
	Literatúra	33

Zoznam obrázkov

1.1	Gangliové bunky a ich aproximácie používané v modeloch an spracoanie obrazu	8
1.2	Haarove prvky	9
2.1	Ukážka aplikácie Card Talk	12
2.2	Ukážka aplikácie Talk in Pictures X - AAC Speak	13
2.3	Ukážka aplikácie Symbotalk - AAC Talker	14
4.1	"Splash screen" aplikácie	17
4.2	Návod na používanie aplikácie	18
4.3	Domovská stránka aplikácie	19
4.4	Úprava novej kartičky	20
4.5	Vytvorenie a zdieľanie dokumentu	21
5.1	Operácia "Opening"	27
5.2	Operácia "Dilation"	27
5.3	Vyhodnotenie významných oblastí obrázku	28
5.4	Rozpoznávanie tvárí	28

Zoznam tabuliek

4.1	Úspešnosť algoritmov na spracovanie obrazu	23
-----	--	----

Úvod

Komunikácia je jednou z najstarších vlastností človeka a je kľúčom k jeho prežitiu. Vyvíjala sa spolu s ľudstvom, od jednoduchých zvukov a nástenných malieb, cez prvé piktogramy a telegraf až po súčasnosť - komplexné abecedné systémy, počítače a schopnosť komunikovať s človekom na druhej strane zemegule v priebehu sekúnd. Zjednodušuje nám každodenný život, pomáha nám vyriešiť problémy a posúva nás vpred.

Nie každý je ale schopný komunikovať plnohodnotne - ľudia s poruchami autistického spektra, sociálno-komunikačnými poruchami a inými poruchami, ktoré im znemožňujú plnohodnotne komunikovať mali v minulosti problém vyjadriť svoje názory a potreby. V dnešnej dobe, kedy sa technológie posúvajú míľovými krokmi existuje veľmi veľa možností, ako takýmto ľudom pomôcť a zjednodušiť im ich každodenný život.

Najjednoduchšie riešenie je pomocou obrázkov, ktoré vyjadrujú nejakú činnosť, predmet alebo stav. V 90. rokoch minulého storočia bol v USA vyvinutý systém PECS - *Picture Exchange Communication System*, ktorý je založený na knihe *Verbal Behavior* od B. F. Skinnera. Systém umožňuje človekovi s poruchou komunikácie vyjadriť svoje potreby a názory pomocou obrázkov, ktoré skladá do viet.

V dobe, kedy sú mobilné zariadenia skoro až rozšírením našich tiel je priam triviálne aby pomáhali takýmto ľuďom. Práve vývoj aplikácie pre vytvorenie kartičiek pre systém PECS a možnosť ich využívať v mobile, a zároveň možnosť ich vytlačiť do väčšieho formátu pre potreby naučenia komunikácie už v ranom veku detí je veľmi dôležité pre spoločnosť.

Rapídny vývoj technológií taktiež umožnil prácu s mnoho nástrojmi, ktoré boli vyvinuté ako open-source knižnice zamerané na špecifické využitie v danej oblasti. Jedna z takých knižníc ponúka nástroje na spracovanie obrazu pomocou počítačového videnia. Táto knižnica nám pomôže zjednodušiť vytváranie kartičiek, a to možnosťou automatického orezania odfotenej fotografie na základe vypočítania jej významných oblastí.

V úvodnej kapitole práce sa nachádzajú teoretické východiská súvisiace s poruchami autistického spektra, poruchami komunikácie a spracovania obrazu. V jej rámci je podrobnejšie definovaná motivácia pre vývoj našej aplikácie, spolu s analýzou využitých technológií. Úvodná kapitola taktiež zahŕňa analýzu podobných aplikácií a prehľad

nami použitých programov a technológií.

Druhá kapitola obsahuje návrh riešenia našej aplikácie a jej užívateľské rozhranie.

V tretej kapitole sú obsiahnuté konkrétne riešenia, ktoré boli potrebné pre konečnú verziu aplikácie, spolu s opisom pripomienok z testovania aplikácie, ktoré bolo vykonané počas vývoja.

Kapitola 1

Úvod do problematiky

1.1 Teoretické východiská

V tejto kapitole sú opísané teoretické východiská, ktoré je vhodné ovládať pri vývoji softvéru v oblasti spracovania obrazu a komunikácie s ľuďmi s poruchami autistického spektra alebo s poruchami komunikácie.

1.1.1 Poruchy autistického spektra

Vymedzenie poruchy autistického spektra

Odborná literatúra rozlišuje dva pojmy, a to *autizmus* a *poruchu autistického spektra* (PAS), ktoré sa často používajú ako synonymá. Ich definícia môže byť problematická, pretože autizmus je zložitý stav, ktorý zahŕňa veľké množstvo aspektov správania sa. [14] Jeho prejavy vedia byť premenlivé, a teda nezáleží len od jedinca, ale aj od jednotlivých etáp jeho vývoja.

Autizmus je skrátený starší pojem, ktorý označuje širšiu skupinu diagnóz, ktoré sa vymedzujú ako pervazívne vývojové poruchy. [10]. To znamená, že ide o poruchy, ktoré postihujú kognitívne, emočné a neurobehaviorálne funkcie, a tým pádom zasahujú do všetkých oblastí osobnosti jedinca. [29]

Na druhej strane *autistické spektrum* je definované ako bohatá kvantitatívna aj kvalitatívna škála kombinovaného postihnutia. Je to ochorenie, na ktorom sa podieľa viac faktorov naraz - napríklad hormonálne a epigenetické faktory, pričom za najvýznamnejší označujú genetický, a ďalej naň majú vplyvy aj črevná mikroflóra, imunitné mechanizmy či iné vplyvy vnútorného prostredia organizmu. Je to porucha, s ktorou sa jedinec musí vysporadúvať celý život. [10]

Porucha autistického spektra sa stala termínom, ktorý zastrešuje viacero rozdielnych typov autistického ochorenia, presnejšie neurovývojové poruchy, ktoré charakterizuje deficit v sociálnej komunikácii a prítomnosť obmedzených záujmov a opakovaného

správania. [11]

Pre potreby tejto bakalárskej práce budeme používať konkrétnejší a aktuálnejší pojem, a to porucha autistického spektra.

Diagnostika PAS

Porucha autistického spektra sa začína prejavovať už v ranom veku, no odhalená môže byť až v priebehu vývoja jedinca, a takisto môže byť aj čiastočne skrytá vďaka naučným stratégiám. [11] Kritéria na jej diagnostiku sú zavedené v svetových klasifikačných systémoch ochorení. Tými sú manuály s názvami *Medzinárodná klasifikácia chorôb*, ktorý je používaný v Európe a vydáva ho Svetová zdravotnícka organizácia, a *Diagnostický a štatistický manuál duševných porúch*, vydávaný Americkou psychiatrickou spoločnosťou.

Pojem porucha autistického spektra sa používa ako zastrešujúca kategória, ktorá sa následne rozdeľuje na konkrétne diagnózy (atypický autizmus, Aspergerov syndróm, detský autizmus či iné). Obecnejšie sa radí medzi pervazívne vývojové poruchy, ktoré sme spomínali vyššie. Ku zlúčeniu diagnóz do zastrešujúceho termínu autistické spektrum došlo najmä preto, že medzi diagnózami chýbali jednoznačne vymedzené hranice a dochádzalo k prekryvaniu symptómov. [25]

Prejavy PAS

K stanoveniu diagnózy je potrebné rozlíšiť niekoľko príznakov, ktoré sú definované aj ako triády autizmu. Tieto triády opisujú príznaky spoločné pre jedincov s poruchou autistického spektra - sociálna interakcia, sociálne správanie a sociálna predstavivosť. [27] Vymenované oblasti patria medzi špecifické rysy, ktoré sú spoločné pre všetkých jedincov s poruchou autistického spektra, avšak platí, že miera deficitu je u každého jedinca individuálna. [11]

Porucha autistického spektra výrazne zasahuje do sociálnych vzťahov jedincov, ktorí majú sťažené doslovné chápanie reči a malú či žiadnu schopnosť porozumieť aspektom neverbálnej komunikácie. [25] Prejavmi taktiež môžu byť aj nezáujem o svoje okolie, uzavretosť či vyhýbanie sa očnému kontaktu [21]. Neporozumenie sociálnej situácii a komunikácii medzi osobami má za následok zlyhávajúce v obojstrannej konverzácii a zdieľaní emócií a záujmov, a zároveň sa jedinci aj ťažšie adaptujú na novú situáciu. [18]

Narušenie vývoja reči postihuje väčšinovú časť detí s diagnostikovanou poruchou autistického spektra a okolo polovice detí sa tak nikdy nerozvinie do dostatočnej formy na komunikáciu s okolím. Typickou je monotónna a neemotívna reč, chýba gestikulácia, a deti síce vedia rozprávať, no slová nepoužívajú na dorozumievanie sa s okolím. Typickým prejavom je aj volenie nevhodnej hlasitosti svojho prejavu či chybné využívanie

zámen. [21]

Poslednou súčasťou je predstavivosť, ktorá je u detí s poruchami autistického spektra obmedzená. [21] Ďalším prejavom sú opakujúce sa aktivity a stereotypné správanie a prejavy. [25] Takisto ťažšie reagujú na zmeny a niektoré činnosti môžu mať až upnutý charakter, ktorý niekedy ide ťažko prerušiť. [16]

Kvôli týmto prejavom môžu ľudia s poruchou autistického spektra naraziť na sociálne vylúčenie, čo je spôsobené neschopnosťou reagovať na ostatných ľudí a ich vnímanie. [17] Pridruženým prejavom je aj rutinný rituál, ako je napríklad chodenie tou istou cestou. [12]

Naproti špecifickým rysom stoja nešpecifické rysy, ktoré nie sú súčasťou diagnostických kritérií, teda neplatia pre všetkých jedincov. Patria medzi ne emočná reaktivita, problémové správanie, percepčné poruchy, odlišnosť v motorickom vývoji, či problémová adaptácia na zmeny. Príčiny vzniku poruchy autistického spektra môžu byť rôzne, avšak jedná sa o prenatálne príčiny. Vonkajšie faktory ako výchova či prostredie môžu ovplyvniť nasledujúci vývoj jedinca, ale ako súčasť terapie. [25]

1.1.2 Systémy na učenie komunikácie u ľudí s poruchami autistického spektra

V priebehu histórie sa terapeuti na učenie komunikácie u detí snažili použiť rôzne systémy. Väčšina z nich boli založené na imitácii zvukov - terapeut vyslovil slovo alebo zvuk a dieťa ho muselo zopakovať. Tento systém ale vyžadoval pozornosť dieťaťa, ktorá sa nie vždy dala zaručiť. Preto sa začali skúmať iné, alternatívne spôsoby výuky komunikácie u detí s poruchou autistického spektra, prípadne s poruchou komunikácie.

Ako alternatíva sa zdala byť posunková reč. Tá nevyžaduje žiadne dodatočné nástroje a postačia na ňu ruky. Bohužiaľ, rýchlo sa ukázali rôzne limitácie, a to hlavne nedostatok ľudí, ktorí ovládajú posunkovú reč. [24] Taktiež bol problém vybudovať slovnú zásobu a niektoré znaky boli pre deti príliš komplikované. Pri komunikácii s ľuďmi s poruchami autistického spektra/poruchami komunikácie sa do určitej miery posunková reč využíva, no nikdy nie ako plnohodnotný nástroj.

Za ďalšiu alternatívu sa považovali tzv. *"picture-point" systémy*, pri ktorých človek s poruchou mal komunikačnú tabuľku, na ktorej sa nachádzali rôzne obrázky vyjadrujúce rôzne potreby, situácie a predmety. Pri tomto systéme sa ukázalo, že niektoré deti s motorickými poruchami mali problém exaktne ukázať na jeden obrázok. Taktiež boli limitované skutočnosťou, že "poslucháč" musí byť v dostatočnej blízkosti, aby videl na tabuľku, čo viedlo k závislosti na výzvu (*prompt dependency*). [13]

Ako schopným systémom sa preukázal PECS - *Picture Exchange Communication System*

1.1.3 Picture Exchange Communication System (PECS)

PECS vznikol v USA v roku 1985, a pracovali na ňom Andy Bondy, PhD. a Lori Frost, MS, CCC-SLP. Pilotný program bol spustený so škôlkarmi diagnostikovanými poruchou autistického spektra v Delaware, a odvtedy bol implementovaný po celom svete. [6]

Je to výučbový systém vytvorený s cieľom naučenia detí s poruchou autistického spektra komunikácie, ktorá je najmä samo-inicializovateľná. Celý tréning pozostáva z 6 fáz, ktoré začínajú učením diet'at'a komunikovať pomocou obrázkov, a končia učením skladania obrázkov do viet.

V prvej fáze, ktorá väčšinou prebieha medzi 6 a 8 mesiacom veku diet'at'a sa učí základ celej komunikácie - oslovenie, získanie pozornosti rodiča, následne poukázanie na nejaký objekt alebo udalosť, a končí to nejakým výsledkom, čiže reakciou od rodiča.

V druhej fáze sa diet'a učí, že obrázky nie sú vždy automaticky dostupné a potrebuje ich najprv individuálne, bez pomoci ostatných nájsť.

V tretej fáze sa diet'a učí rozlíšiť rôzne obrázky a rôzne kartičky.

V štvrtej fáze sa diet'a učí používať tzv. štartér viet (frázy typu "*Chcem*", "*Počujem*", "*Vidím*"). Tieto štartéry majú vlastné obrázky, čiže diet'a už v štvrtej fáze "skladá" krátke vety.

Piata fáza učí odpovedať na priamu otázku - tréner položí jednoduchú otázku typu "*Čo chceš?*", a po krátkej pauze poukáže na kartičku zodpovedajúcu odpovedi "*Chcem*". Cieľom je, aby diet'a začalo odpovedať predtým, ako tréner ukáže na daný obrázok.

V poslednej fáze tréner pridá viac kartičiek, ktoré začínajú vetu a učí rozdiel medzi nimi, a namiesto ponúknutia daného objektu na podnet "*Vidím*" ponúkne iba slovnú odpoveď.

Po úspešnom ukončení prvých piatich fáz, popri učení šiestej dokážeme postupne rozširovať slovník kartičiek.[13]

1.1.4 Významné oblasti fotografie

Ľudské vnímanie dostáva obrovské možno dáť každú sekundu, a ich spracovanie a upriamenie pozornosti sa deje automaticky, no z hľadiska vývoja a výskumu je potrebné identifikovať oblasti scény, na ktoré sa oči zameriavajú - tieto oblasti označujeme ako významné. Ľudské vizuálne vnímanie závisí od množstva faktorov - farba, textúra, intenzita, náhle zmeny, pohyb.

Pri vnímaní scény rozlišujeme dva hlavné prístupy: bottom-up (*zdola-nahor*), ktorý využíva exogénnu pozornosť, a top-down (*zhora-nadol*), ktorý využíva pozornosť endogénnu. Exogénna pozornosť sa zameriava na vonkajšie podnety, upozorňuje na významné časti automaticky a má rýchly časový priebeh. Na druhú stranu, endogénna pozornosť je riadená človekom, je na ňom závislá a má pomalý časový priebeh.

Na určovanie významných oblastí sa využívajú dva hlavné prístupy - sledovanie pohybu očí a výpočtové modely na detekciu. Sledovanie pohybu očí je zatiaľ finančne náročnejšie, pre presnosť vyžaduje špeciálny hardvér, a preto je používané najmä pri výskumoch. V posledných rokoch sa tento hardvér dostáva aj do komerčného sveta, a to napríklad do vyšších modelov okuliarov pre virtuálnu realitu. Modely na detekciu významných oblastí sú predmetom výskumov už vyše 30 rokov, a bolo vytvorené veľké množstvo modelov na báze rôznych prístupov. [23]

1.1.5 Algoritmy na detekciu významných oblastí fotografie

V rámci našej práce sme sa stretli s dvoma algoritmami na detekciu významných oblastí fotografie - princíp Spektrálneho zvyšku (*Spectral Residual approach*) a Jemne zrnitý princíp (*Fine Grained approach*). V nasledujúcich podkapitolách popíšeme ich fungovanie, v kapitole č. 2 porovnáme ich presnosť a vyhodnotíme, ktorý je pre našu aplikáciu lepší.

Spectral Residual Saliency

Spektrálny reziduálny model je algoritmus na detekciu významných oblastí založený na myšlienke efektívneho kódovania. Využíva skutočnosť, že fotografie sa riadia predvídateľnými distribúciami, a tieto redundancie potláča, aby sa zameral len na neočakávané signály, ktoré možno doručiť do neskorších fáz spracovania.

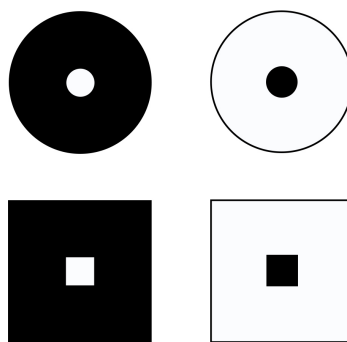
Prvým krokom v modeli je reprezentácia obrazu v logaritmickom spektre, ktoré možno získať logaritmom Fourierovho spektra obrazu. Na rozdiel od log-log reprezentácie spektra, ktorá nie je vhodná na analýzu jednotlivých obrazov z dôvodu zlého pomeru vzorkovacích bodov a nerovnosti stupnice.

Druhým krokom v modeli spektrálneho rezídua je výpočet spektrálneho rozdielu, čo je rozdiel medzi pôvodným logaritmickým spektrom a vyhladenou verziou priemerného logaritmického spektra. Vyhladená verzia logaritmického spektra sa získa zjednotením logaritmického spektra s Gaussovým filtrom. Spektrálny zvyšok predstavuje anomáliu v logaritmickom spektre a používa sa na vytvorenie mapy významných oblastí použitím funkcie *threshold* a normalizáciou. [19]

Fine Grained Saliency

Princíp "jemnej zrnitosti" je založený na tom, ako človek vníma obraz - princíp centrum-okolie. Gangliové bunky v oku sú bunky, ktoré sú citlivé na intenzitu svetla. Vieme rozdeliť na dva typy - *off-center* (citlivé na zníženie intenzity svetla v stredovej časti) a *on-center* (citlivé na zvýšenú intenzitu svetla v stredovej časti). Ak sú svetlom zasiahnuté obe časti (t.j. stredová i okrajová), rapídne sa znižuje schopnosť rozpoznávať

intenzitu.



Obr. 1.1: Hore, zľava doprava: gangliová bunka on-center, gangliová bunka off-center. Dole, zľava doprava: on-center aproximácia, off-center aproximácia

Modely pre spracovanie obrazu sa snažili napodobniť princíp gangliových buniek. Najprv sa obraz prekonvertuje do čiernobiela, a následne sa z neho vytvorí tzv. integrány obraz. Predstavme si, že máme funkciu $f(x,y)$ nad určitým výrezom fotografie, ktorá z pixelov získava reálne čísla (napríklad intenzitu pixelov) - naivný princíp je pre každý pixel vypočítať hodnotu až po zadaní súradníc výrezu. Integrovaný obraz vypočíta hodnoty predom, a do každého pixelu uloží súčet všetkých hodnôt pixelov, nachádzajúcich sa smerom doľava a nahor, čo rapídne zrýchli čas spracovania obrazu. [15]

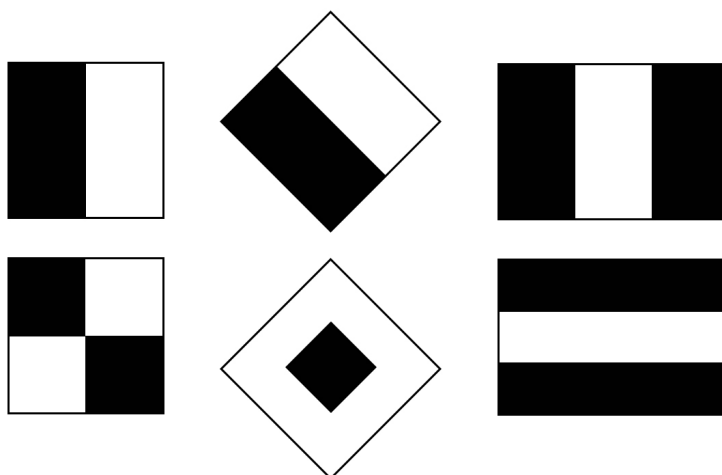
Na získanie mapy významných oblastí algoritmus využíva tzv. *Ittiho model*, ktorý obrázky rozdelí na tri časti - Gaussove pyramídy farieb, intenzít a orientácií. Tieto pyramídy sa spracujú a aplikuje sa na ne princíp centrum-okolie, a následne sa z nich vytvorí spolu 42 máp (6 pre intenzitu, 12 pre farbu a 24 pre orientáciu). Mapy sa pre každý príznak sčítajú, a výsledné 3 mapy sa skombinujú, aby sme získali mapu významných oblastí obrázka. [20]

1.1.6 Viola-Jones algoritmus, Haarove kaskády

Viola-Jones algoritmus je pomenovaný podľa svojich tvorcov Paula Violu a Michaela Jonesa, ktorí ho vyvinuli v roku 2001. Algoritmus je založený na postupnom skenovaní obrazu zľava-doprava, zhora-nadol a hľadá prvky, ktoré sú podobné vopred definovanému súboru vzorov, známemu ako Haarove prvky. [26]

Haarove prvky sú jednoduché binárne vzory, ktoré sú pomocou Viola-Jones algoritmu aplikované na obraz, s cieľom definovať časti obrazu, ktoré sú podstatné. Pri každom použití filtra sa vypočíta hodnota na základe kontrastu a jasnosti pixelov. Ak takto definujeme mnoho podstatných častí na niekoľko rôznych obrázkoch, získame databázu prvkov, tzv. klasifikátor, ktorý vieme následne použiť na detekciu objektov. [22] Výhoda Haarových prvkov je, že vieme si preddefinovať významné prvky nielen tvárí, ale

aj iných objektov.



Obr. 1.2: Príklady Haarových prvkov, ktoré sú následne aplikované na fotografiu

Keby sme aplikovali Haarov klasifikátor na obrázok o veľkosti 16x16 pixelov, získali by sme 160000 prvkov [22], a preto sa spolu s Viola-Jones algoritmom a Haarovými klasifikátormi používa algoritmus AdaBoost. Myšlienka AdaBoost spočíva v iteratívnom tréňovaní série slabých klasifikátorov na rôznych podmnožinách tréňovaných údajov. Slabý klasifikátor je jednoduchý model, ktorý je len o niečo lepší ako náhodné hádanie.

Počas každej iterácie algoritmus priradzuje váhy tréňovaným vzorkám na základe ich predchádzajúcej klasifikačnej presnosti. Vzorkám, ktoré boli v predchádzajúcej iterácii nesprávne klasifikované, sa priradia vyššie váhy, zatiaľ čo správne klasifikovaným vzorkám sa priradia nižšie váhy.

Ďalší slabý klasifikátor sa potom trénuje pomocou aktualizovaných váh, takže sa viac zameriava na nesprávne klasifikované vzorky. Konečný silný klasifikátor sa vytvorí kombináciou slabých klasifikátorov.

Pri iteratívnom skenovaní obrazu pomocou Viola-Jones algoritmu sa teda najprv použije AdaBoost - zistí sa, či vôbec v danom okne je tvár, ak nie, tak ide ďalej, a iba ak áno, tak aplikuje Haarove klasifikátory.

1.2 Motivácia

1.2.1 Aplikácia pre rodičov

Rodičia s deťmi s poruchami autistického spektra, prípadne s poruchami komunikácie, denne vyžívajú nejaký systém pre komunikáciu s ich deťmi. Keďže jedno zo 100 detí narodených dnes má diagnostikovanú nejakú formu poruchy autistického spektra [28], je potrebné vytvoriť voľne dostupné a jednoduché nástroje pre vytváranie a generovanie materiálov na výuku týchto systémov.

S neustále napredujúcim softvérom a hardvérom, ktorý sa nachádza v našich vreckách vzniká aj dopyt po aplikáciách, ktoré by túto funkcionality poskytovali. Z prieskumu trhu sme zistili, že na trhu sa nachádza hneď niekoľko aplikácií zameraných na túto problematiku, no ani jedna kompletne nespĺňala všetky požiadavky - či už boli nekompletné, neobsahovali všetku funkcionality, neboli aktualizované niekoľko rokov alebo boli nepoužiteľné z hľadiska užívateľa.

Žiadna aplikácia zameraná na túto problematiku nie je vyvinutá priamo pre ľudí s poruchami autistického spektra, čo je námet pre ďalší výskum.

1.2.2 Požiadavky pre aplikáciu

Hlavnou podmienkou pri vývoji aplikácie bude obzvlášť výber správneho užívateľského prostredia. Keďže je aplikácia mierená na ľudí, ktorí nemusia byť s technológiami úplne zladení, je dôležité, aby aplikácia bola čo najjednoduchšia na používanie. Užívateľské rozhranie preto musí byť prehľadné, nesmie byť presaturované zbytočnými funkciami a malo by obsahovať vysvetlivky.

Kapitola 2

Prehľad existujúcich aplikácií

Na Android trhu sme našli viacero aplikácií zameraných na komunikáciu pomocou PECS systému, no ani jedna nie je orientovaná na export kartičiek do PDF formátu pre tlač a nedisponuje touto funkcionalitou. Väčšina je vyvinutá na použitie a generovanie kartičiek priamo v mobile, bez možnosti zdieľania kartičiek.

V tejto kapitole uvádzame prehľad a opis niektorých aplikácií, ktoré majú podobné alebo rovnaké vlastnosti ako tá naša a sú k dispozícii na stiahnutie pre mobilné zariadenia so systémom Android cez Google Play Store. Vybrali sme také aplikácie, ktoré boli lokalizované na celosvetový trh (anglický jazyk), mali dobré užívateľské hodnotenia a mali adekvátny počet stiahnutí.

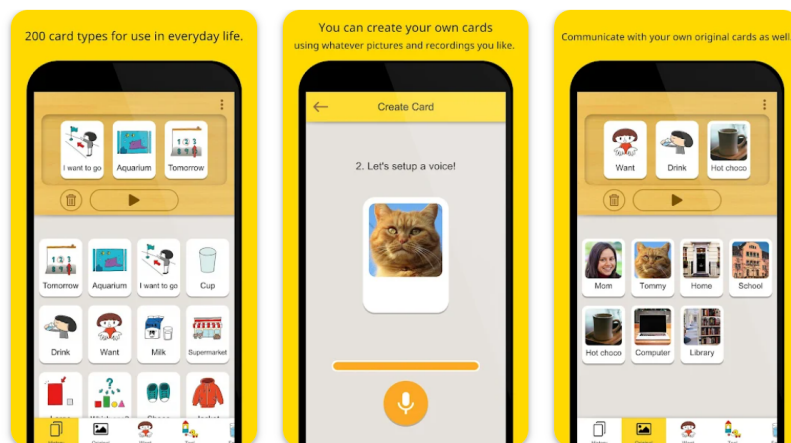
2.0.1 Card Talk

Vo vyhľadávaní, recenziách a funkcionalite dominuje aplikácia Card Talk, s viac ako 100 tisíc stiahnutiami a hodnotením 4.6 z 5 na Google Play Store, no naposledy bola aktualizovaná v júli 2022. [3]

Aplikácia po stiahnutí obsahuje dataset predom vytvorených kartičiek, spolu s kresleným obrázkom a nahraným zvukom, ktorý vyslovuje popis kartičky, no tieto kartičky sa nedajú upravovať. Hlavnou funkcionalitou aplikácie je zoradovanie kartičiek do "viet", kde sa po stlačení tlačidla prehrávajú slová z kartičiek, ktoré sa vizuálne zvýraznia pri každom slove a sformujú tak jednoduchú vetu. Jedna kartička sa dá v jednej vete využiť viackrát.

Predom vytvorené kartičky sú usporiadané do priečinkov, a nedajú sa upravovať. Užívateľ má možnosť zvoliť si z dvoch módov aplikácie - všetky kartičky, alebo balíček. Do balíčka si vie vložiť maximálne osem kartičiek, a to i vlastné, no keď je v tomto móde, nemá prístup k žiadnym iným kartičkám. Zmena týchto módov sa vykonáva v menu.

Ponúka taktiež možnosť vytvoriť si vlastné kartičky - fotografia sa dá buď odfotiť, alebo vybrať z galérie, následne sa nahrá audio a napíše sa popis. Kartička je potom



Obr. 2.1: Ukážka hlavných používateľských rozhraní aplikácie Card Talk, zľava: hlavné zobrazenie predpripravených kartičiek, tvorba vlastnej kartičky, použitie vlastných kartičiek.

uložená v priečinku "Vlastné" a neexistuje možnosť kategorizovať si, alebo zdieľať nami vytvorené kartičky.

V našom testovaní dopadla táto aplikácia najlepšie a vyvolávala pocit dokončenosti (v porovnaní s ostatnými). Napriek zbytočne dlhej poistke tlačidla menu a neumožneniu upravovania existujúcich kartičiek si vieme predstaviť túto aplikáciu používať.

Používateľské rozhranie je viditeľne mierené na rodičov s malými deťmi, a to veľkými obrázkami, obľými hranami a veľa farbami. Aby sme sa dostali do menu, je potrebné na 5 sekúnd podržať tlačidlo, čo je myslené ako poistka pri používaní.

2.0.2 Talk in Pictures X - AAC Speak

Táto aplikácia je menej známa, má len 10 000 stiahnutí na Google Play Store, hodnotenie 3.3 z 5 of 113 užívateľov, a naposledy bola aktualizovaná viac než 3 roky dozadu. [9]

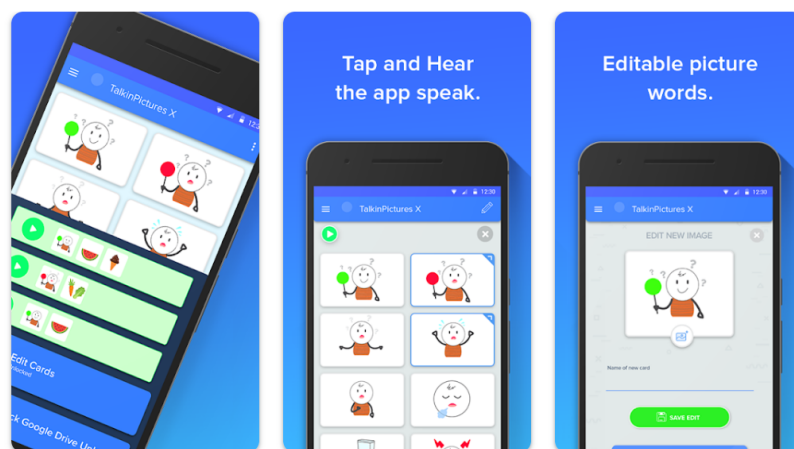
Pri prvom otvorení aplikácie sa užívateľovi zobrazí jednorázové okno, kde je minimalisticky vysvetlená hlavná funkcionálna pomocou obrázkov.

Rovnako ako Card Talk, aplikácia obsahuje dataset predom vytvorených kartičiek, s kresleným obrázkom a nahraným zvukom. Kartičky sa dajú usporadúvať do viet, no dajú sa použiť aj samostatne.

Aplikácia ponúka vytváranie nových kartičiek, no táto funkcia je nezmyselne dostupná iba pri upravovaní už existujúcej kartičky. Pri úprave už existujúcich kartičiek je dostupné prepísanie textu a zvolenie obrázka z kamery, galérie, z databázy preddefinovaných ikoniek a z internetu. Obe funkcionality sú spoplatnené mesačným poplatkom. Aplikácia disponuje skúšobnou dobou zadarmo pre otestovanie funkcií na prvých 7 dní.

Aplikácia taktiež ponúka históriu posledných 3 použitých kartičiek, no aby sa k

nej užívateľ dostal, musí kliknúť na menu vpravi hore a následne na "Show History". História sa zobrazí ako dialógové okno ponad kartičky, čo nie je ideálne riešenie.



Obr. 2.2: Ukážka hlavných používateľských rozhraní aplikácie Talk in Pictures X - AAC Speak, zľava: hlavné zobrazenie kartičiek, použitie kartičky samostatne, tvorba vlastnej kartičky.

Pri našom testovaní aplikácie sme narazili na viacero problémov. Predom vytvorené kartičky boli usporiadané do priečinkov, no priečinky boli označené čiernym trojuholníkom v pravom hornom rohu kartičky, a teda neboli veľmi odlišené od samostatných kartičiek. Taktiež otvorenie priečinku neobsahovalo žiadnu animáciu, bolo instantné - čo môže pôsobiť mätúco. Skladanie kartičiek do viet je možné iba cez priečinky - priečinok má definíciu "I feel", kartičky v ňom definujú pocity, a teda veta môže vyzerat' napríklad ako "I feel sad". Nie je možné kartičky skladat' do viet samostatne.

Používateľské rozhranie je minimalistické a funkčné, no absencia akéhokoľvek návodu a popisov pri ikonkách je pri prvom použití mätúca. Dizajn aplikácie nie je mierený na deti, je skôr strohý a jednoduchý, pre všetkých užívateľov s poruchou komunikácie.

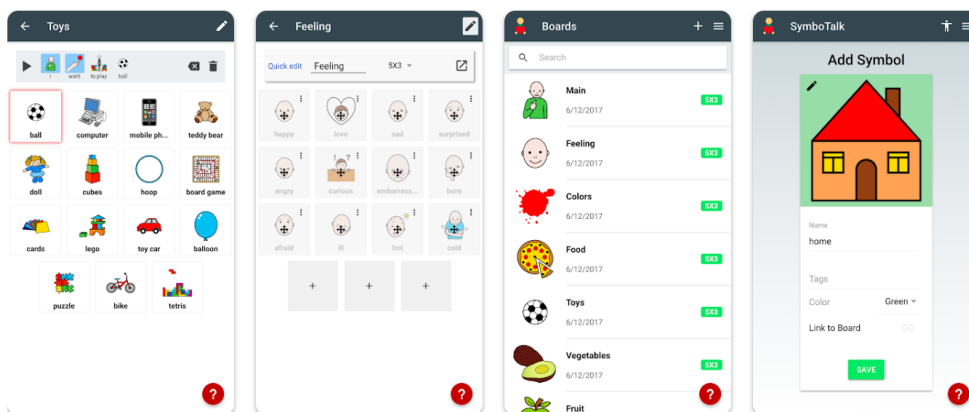
2.0.3 SymboTalk - AAC Talker

Aplikácia má na Google Play Store cez 100 000 stiahnutí a hodnotenie 4 z 5 od 188 užívateľov, a poslednú aktualizáciu získala v januári 2023, čiže je najviac udržiavaná z našich 3 príkladov. [8]

V základe aplikácia ponúka dva módy používania - komunikačný mód a editor.

Pri prvom otvorení aplikácie sa užívateľovi zobrazí jednorázové okno, kde je vysvetlená hlavná funkcionálna pomocou obrázkov a na konci sa nachádza odkaz na systémovú podporu a FAQ.

V komunikačnom móde sa nachádza ochrana voči ukončeniu aplikácie, a hlavnou funkcionalitou je "klávesnica" zložená z obrázkov, cez ktorú užívateľ skladá vety a tak komunikuje. Z našich troch príkladov to funguje najlepšie, podobne ako reálna klávesnica, no užívateľ nie je schopný jeden obrázok použiť vo vete dvakrát, čo môže pôsobiť limitujúco pre niektorých užívateľov.



Obr. 2.3: Ukážka hlavných používateľských rozhraní aplikácie Symbotalk - AAC Talker, zľava: používanie kartičiek vo vetách, upravenie balíčku kartičiek, zobrazenie balíčkov, vytvorenie vlastnej kartičky.

V móde editor je možné vytvárať nové kartičky, kategorizovať ich pomocou tagov, zvoliť im farbu, prípadne upravovať už existujúce, usporadúvať ich do balíčkov a kategorizovať ich podľa seba. Po vytvorení vie užívateľ nahráť zvuk, opisujúci kartičku. Užívateľ si vie vytvárať vlastné balíčky, no jedna kartička nevie byť v dvoch balíčkoch naraz. Prispôbiť sa dá aj veľkosť mriežky, v ktorej sa kartičky zobrazujú.

V každom okne aplikácie sa nachádza ikonka malého otáznika vpravo dole, ktorý slúži na vysvetlenie aktuálneho módu. V nastaveniach sa taktiež nachádza možnosť zvoliť vysokokontrastný režim pre užívateľov so zhoršeným zrakom.

Aplikácia ponúka platenú verziu za 1€ na mesiac, no tento balíček odomkne cloudové úložisko, zálohovanie a zosynchronizovanie viacerých zariadení, a automatické generovanie kartičiek pomocou AI. Samotná tvorba a použitie kartičiek teda obmedzená nie je. Taktiež sa v aplikácii nenachádzajú žiadne reklamy,

Popri našom testovaní sa v aplikácii vyskytlo pár chýb - tlačidlo späť nefungovalo, po vytvorení kartičky a následne stlačení tlačidla späť sme sa dostali k tvorbe tej istej kartičky, a pri vyhľadávaní sa zobrazovala fráza "error, please try again" aj napriek tomu, že vyhľadávanie bolo úspešné.

Táto aplikácia je funkcionalitou a spracovaním na tom najlepšie z našich troch príkladov a je najviac použiteľná, no zaslúžila by si malé prerobenie užívateľského rozhrania.

Kapitola 3

Technológie

3.1 Použité knižnice

3.1.1 CameraX

CameraX je nové API priamo od vývojárov v Google slúžiace na využitie kamery mobilu v aplikáciách. Je spätne kompatibilné až po Android 5.0 [2] a podľa testovaní dosahuje konzistentné výsledky na väčšine zariadeniach. Disponuje taktiež rozsiahlymi návodmi a dokumentáciou. [2] Pre našu aplikáciu je dôležité zobrazenie kamery a odfotenie obrázka priamo v aplikácii bez toho, aby sme museli otvárať samostatnú aplikáciu kamery.

3.1.2 Room

Room je knižnica od vývojárov v Google ktorá je súčasťou Android Jetpack. Slúži na ukladanie netriviálneho množstva dát v aplikácii, a to ako abstraktná vrstva nad SQLite systémom. Ponúka jednoduché nástroje na prácu s databázou bez toho, aby sme museli duplicitne vytvárať dopyty do databázy, a taktiež dokáže v reálnom čase kontrolovať nami vytvorená SQL dopyty. [7] V našej aplikácii využijeme databázu na ukladanie odkazu na fotografiu a jej popis.

3.1.3 OpenCV

OpenCV (*Open Source Computer Vision Library*) je open-source knižnica ¹ s nástrojmi na spracovanie obrazu. Obsahuje mnoho optimalizovaných algoritmov, ktoré slúžia na rôzne potreby - rozpoznávanie tváre, identifikácia objektov a pohybu a iné. Keďže je open-source, a zároveň najväčšia a najrozšírenejšia knižnica zameriavajúca sa na túto problematiku, tak ju tisíce ľudí aktívne upravujú, opravujú a pridávajú novú

¹Knižnica s otvoreným zdrojovým kódom

funkcionalitu. Ponúka natívne interfaces pre jazyky C++, Python, Java a MATLAB. [5]

Knižnica disponuje obsiahlou dokumentáciou všetkých modulov, viacerými vzorovými projektmi a online fórom, kam užívatelia vedia nahrať otázky, prípadne chyby v kóde. Bohužiaľ, implementácia v jazyku Kotlin, a teda kompatibilita s Androidom je veľmi limitujúca, a aktuálne vydanie neobsahuje všetky potrebné nástroje potrebné k našej práci.

3.1.4 Chaquopy

Chaquopy je open-source knižnica ktorá umožňuje plne integrovať Python do jazyka Kotlin, spolu s podporou Python knižníc dostupných cez pip. Ponúka jednoduché API na volanie Python funkcií z jazyka Kotlin a naopak. Disponuje obsiahlou dokumentáciou a taktiež GitHub Issues systémom, kde sa vývojári aktívne zapájajú do diskusií a existuje tak možnosť vyriešiť chybu/opraviť bug do pár dní. [4]

3.2 Použité programy

3.2.1 Android Studio

Android Studio je oficiálne integrované vývojové prostredie od spoločnosti Google vyvíjané v spolupráci s JetBrains. Tvorilo základ pri vývoji našej aplikácie, a to najmä kvôli veľmi dobrým funkcionalitám, ako napríklad natívne prepojenie s jazykom Java, ľahké pridávanie modulov, virtuálne zariadenie pre testovanie priamo v počítači, prehliadanie databázy vytvorenej aplikáciou a obsiahle možnosti pri odchyťovaní chýb pri ladení.

3.2.2 IntelliJ IDEA

IntelliJ IDEA je integrované vývojové prostredie od spoločnosti JetBrains, ktoré ponúka možnosť vývoja vo viacerých jazykoch (Java, Kotlin, Python, HTML...), no taktiež ponúka integrované nástroje na správu databázy, prepojenie s GitHub a správu modulov cez Maven alebo Gradle.

Bolo potrebné pre vývoj Python scriptov na spracovanie obrazu, keďže Kotlin knižnice natívne túto funkcionalitu nepodporujú.

Kapitola 4

4.1 Uživateľské rozhranie

V tejto kapitole opisujeme nami navrhnuté používateľské rozhranie. Popíšeme fungovanie všetkých funkcionalít z pohľadu užívateľa, všetky obrazovky a iné grafické prvky aplikácie.

4.1.1 Úvodná obrazovka

Po zapnutí aplikácie sa užívateľovi na chvíľu zobrazí tzv. "splash screen", ktorá vyobrazuje logo aplikácie. Následne je užívateľ presmerovaný na domovskú stránku aplikácie.

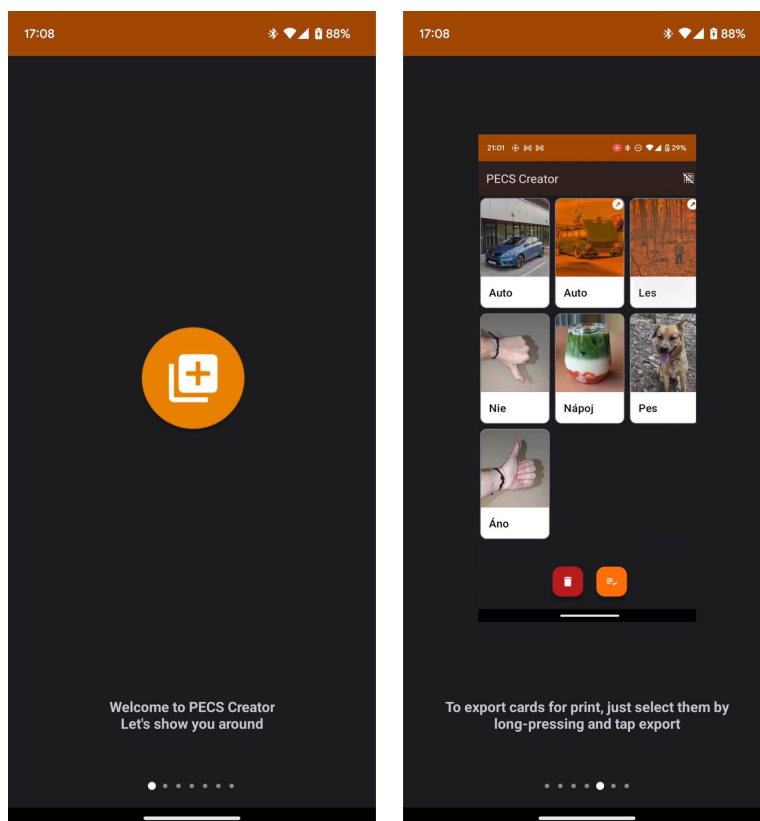


Obr. 4.1: Úvodná stránka aplikácie, ktorá sa zobrazí pri spustení aplikácie

4.1.2 Návod na použitie aplikácie

Pri prvom zapnutí aplikácie sa užívateľovi zobrazí okno, v ktorom je vo forme videí postupne vysvetlená celá funkcionálnosť aplikácie. Užívateľ je "nútený" postupne prejsť celým návodom, aby mu bolo umožnené dostať sa do aplikácie.

Zvolili sme formu videonávodu z jednoduchého dôvodu - aplikácia je mierená na ľudí, ktorí nemusia byť s technológiami zruční, a priame ukázanie funkcionality tak bude viac nápomocné, ako napríklad textový návod.



Obr. 4.2: Videonávod na používanie aplikácie, ktorý sa užívateľovi zobrazí pri prvom otvorení aplikácie

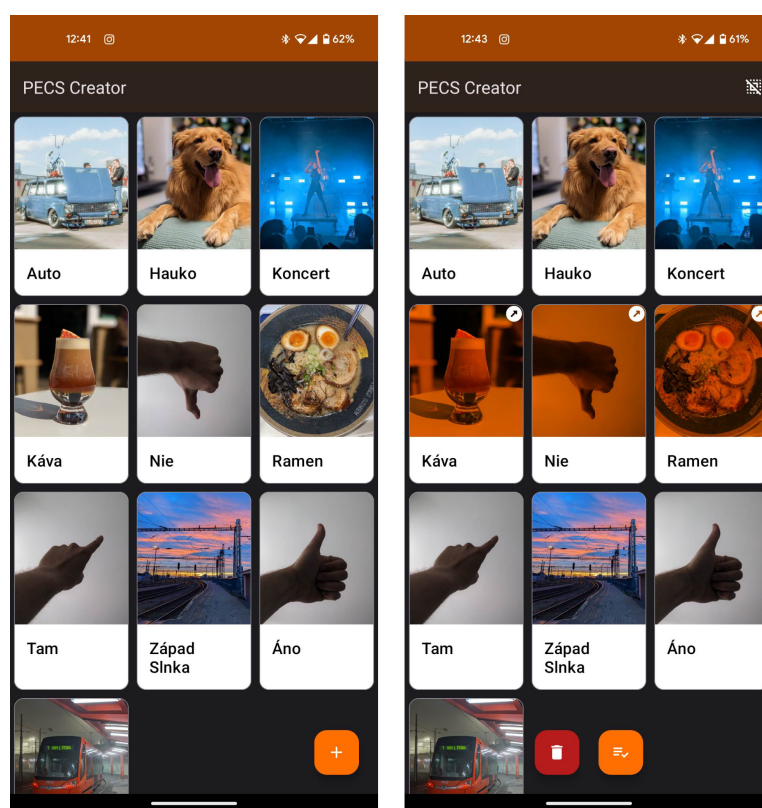
4.1.3 Domovská stránka

Hlavná stránka aplikácie, kde užívateľ vie interagovať s už vytvorenými kartičkami, prípadne zapnúť ďalšie funkcionality aplikácie. Kartičky sú usporiadané podľa abecedy.

Interakcia s vytvorenými kartičkami

Ak už užívateľ má vytvorené nejaké kartičky, vie ich dlhým podržaním označiť. Kartička následne zmení odtieň na oranžový, a v jej ľavom hornom rohu sa zobrazí symbol označenia. Po označení kartičiek sa užívateľovi zobrazia 3 nové tlačidlá - vymazanie,

export v spodnej časti obrazovky, a vymazanie označenia v pravom hornom rohu. Po kliknutí na tlačidlo vymazania sa užívateľovi zobrazí dialógové okno, s potvrdením na vymazanie, a po jeho schválení sa kartičky nenávratne vymažú z databázy. Po kliknutí na tlačidlo export-u, podľa počtu zvolených kartičiek sa stane nasledovné: najprv sa zobrazí dialógové okno kde si užívateľ zvolí, koľko kartičiek chce mať uložených na jednej strane, a následne, ak je počet kartičiek väčší ako 1, do priečinka Dokumenty v zariadení sa uloží vygenerovaný dokument, a užívateľovi sa zobrazí okno na zdieľanie do všetkých možných aplikácií, prípadne poslanie kontaktu. Ak je počet kartičiek väčší ako nastavený najvyšší počet kartičiek na stranu, dokument bude obsahovať viac ako jednu stranu. Po kliknutí na tlačidlo pre vymazanie označenia sa užívateľom označené kartičky anulujú do pôvodného stavu.



Obr. 4.3: Zľava doprava - Domovská stránka s užívateľom vygenerovanými kartičkami, zobrazenie kartičiek zvolených užívateľom

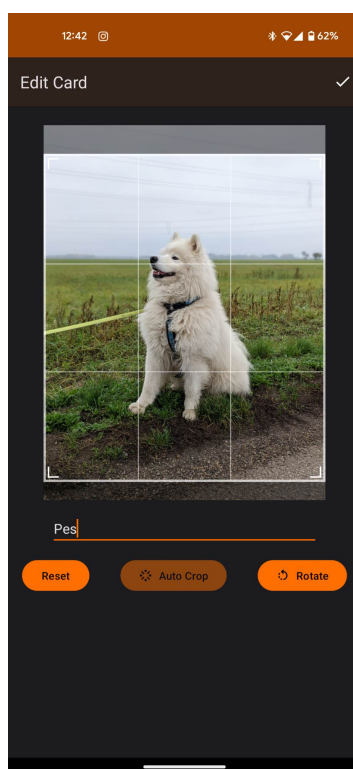
Vytvorenie novej kartičky

Užívateľovi je umožnené vytvárať nové kartičky, a to dvoma spôsobmi - fotografiu vyberie z galérie, alebo ju priamo odfoťí. Obidve možnosti sú spustené rovnakou akciou. Ak užívateľ nemá označené žiadne kartičky, v ľavom dolnom rohu je tlačidlo, na ktoré keď užívateľ klikne, zobrazia sa dve ďalšie tlačidlá s ikonami galérie a fotoaparátu. Pri použití fotoaparátu sa v rámci aplikácie zobrazí náhľad fotografie a tlačidlo spúšte.

Kliknutím na spúšť sa fotografia uloží a užívateľ je presmerovaný na okno pre úpravu kartičky. Pri použití galérie sa zobrazí galéria, a po zvolení obrázka je užívateľ taktiež presmerovaný na okno pre úpravu.

4.1.4 Úprava novej kartičky

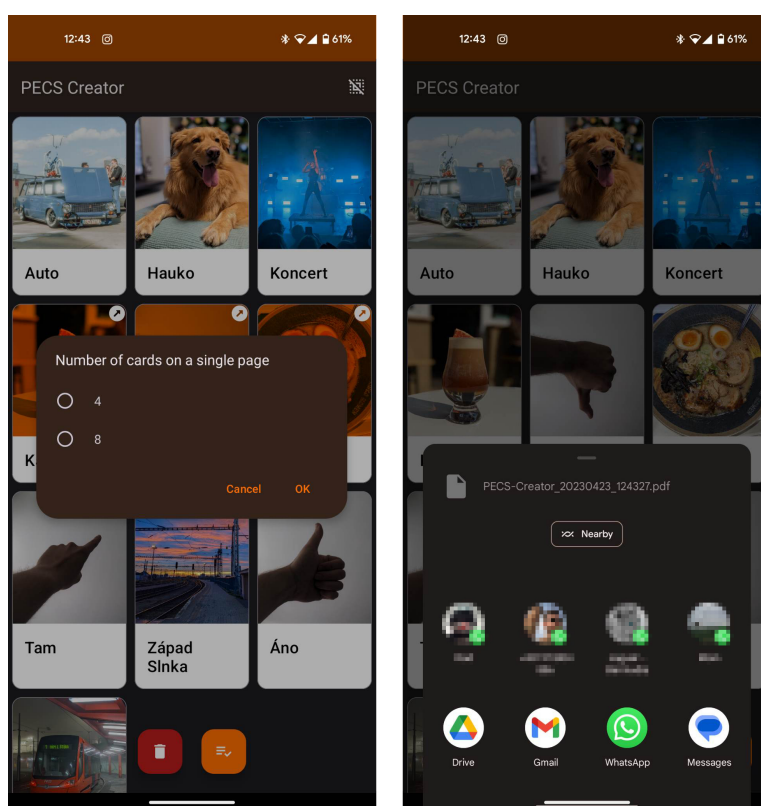
Po získaní obrázku na kartičku sa užívateľovi zobrazí okno pre úpravu fotografie. Pod obrázkom sa nachádza textové pole, kde je potrebné vpísať popis obrázku. Pod textovým polom sa nachádzajú tri tlačidlá. Ľavé slúži na anulovanie vykonaných úprav - fotografia sa vráti do pôvodného stavu. Stredné tlačidlo slúži na automatické orezanie. Spočiatku je zablokované, hneď po získaní fotografie sa fotografia začne spracovávať, a akonáhle je získaný výsledok, tlačidlo sa odblokuje, umožňujúc užívateľovi použiť návrh orezania na báze počítačového videnia a rozpoznávania tvárí. Pravé tlačidlo slúži na otáčanie obrázku v smere hodinových ručičiek. Ak je užívateľ spokojný s orezom fotografie a s jej popisom, potvrdí uloženie kartičky v pravom hornom rohu. Kartička sa uloží do databázy a užívateľ je presmerovaný na domovskú obrazovku, kde sa už nachádza aj práve vytvorená kartička,



Obr. 4.4: Stránka s možnosťami pre úpravu fotografie, pridanie textu a automatického orezania

4.1.5 Export kartičiek do formátu PDF

Ak užívateľ má potrebný počet označených kartičiek pre export (najmenej jedna), je mu umožnené kartičky vyexportovať do PDF formátu pre tlač. Na jednu stranu sa zmestí najviac 8 kusov kartičiek, respektíve toľko, koľko si vyberie pred exportom, no užívateľovi je umožnené vytvoriť dokument aj s menším počtom - či už pre to, lebo viac kartičiek nemá, alebo viac kartičiek nepotrebuje. Po potvrdení počtu kartičiek sa dokument uloží do pamäte zariadenia, a ihneď sa užívateľovi zobrazí okno s viacerými možnosťami zdieľania dokumentu - email, Bluetooth alebo iné komunikačné aplikácie, ktoré má nainštalované v jeho mobilnom zariadení. Dokument je vo formáte A4, a kartičky sú oddelené tenkými čiarami, pre jednoduché nastrihanie.



Obr. 4.5: Zľava doprava - Dialógové okno pre zvolenie počtu kartičiek na jednej strane, dialógové okno pre zdieľanie vygenerovaného dokumentu

4.2 Dátový model

Naša aplikácia umožňuje vytvorené kartičky ukladať a následne používať. Na ukladanie sme využili Room API, ktoré slúži ako komunikátor medzi SQLite databázou a aplikáciou. V podkapitole 2.2.1 popíšeme pôvodne zamýšľaný dátový model a v podkapitole 2.2.2 popíšeme model, ktorý sme zvolili do konečnej verzie aplikácie.

4.2.1 Pôvodne zamýšľaný dátový model

V začiatkových fázach vývoja sme mali v pláne fotografie kartičiek ukladať do pamäte zariadenia, a ich URI uložiť do databázy spolu s popisom a identifikátorom. Tento prístup sa nám neosvedčil, a to z dôvodu, že v galérii užívateľa vznikol priečinok s fotografiami, ktoré použil na kartičky, a ten môže pôsobiť rušivo a mätúco, nehovoriac o tom, že užívateľ mal možnosť tieto fotografie vymazať a tým pádom znefunkčnit' im prislúchajúce kartičky.

4.2.2 Výsledný zamýšľaný dátový model

Nakoniec sme zvolili prístup ukladania fotografií priamo do databázy aplikácie, bez potreby ukladania ich lokálne v súboroch užívateľa. Room API povoľuje ukladanie obrázkov vo formáte *ByteArray*. My tento formát dosiahneme prekonvertovaním bitmapy obrázka tesne pred uložením do databázy. Samozrejme nemôžeme ukladať obrázky v plnom rozlíšení (problém by to bol najmä pri kartičkách s priamo odfotenými fotografiami, ktoré sú v natívnom rozlíšení fotoaparátu zariadenia), a preto fotografie ešte pred konverziou zmenšujeme do veľkosti dostačujúcej pre tlač do medzinárodného formátu A4. V našom testovaní sme sa dostali k záveru, že 600 pixelov na šírku a 700 na výšku je adekvátnych pre naše potreby.

4.3 Porovnanie algoritmov na spracovanie obrazu

Na zistenie, ktorý algoritmus vyhodnocovania významných oblastí je lepší, sme použili časť obrázkovej databázy FSOD (*Few-Shot-Object-Detection-Dataset*). Na každý obrázok sme aplikovali oba algoritmy, výsledok sme upravili rovnakými efektami a výsledné ohraničenie významnej oblasti sme si ručne porovnali, zapisujúc si výsledky. 84% snímok bolo analyzovaných správne, a z toho 55% bolo úspešne orezaných algoritmom na princípe spektrálneho zvyšku, 25% bolo úspešne orezaných Fine-Grained algoritmom, a zvyšných 20% bolo orezaných oboma algoritmami približne narovnať. Zvyšných 16% snímok bola neúspešne analyzovaná a orezaná oboma algoritmami.

Pri testovaní sme dospeli k záveru, že Fine-Grained algoritmus je úspešný iba pri čistých, nerušivých pozadiach na ktorých sa subjekt vyníma, a pri komplikovanejších fotografiách bol úspešnejší práve Spectral Residual algoritmus, ktorý nemal problémy ani s náročnejšími scénami, kde subjekt nebol až tak výrazný.

Tabuľka 4.1: Tabuľka znázorňuje celkový počet obrázkov, a následne zľava doprava úspešnosti algoritmov, počet obrázkov, pri ktorých boli výsledky podobné a počet tých, pri ktorých nebol úspešný ani jeden algoritmus.

Celkový počet obrázkov	Spectral Residual	Fine Grained	Oba podobne	Ani jeden
750	345	157	127	121

4.4 Material Design 3

Material Design je open-source knižnica od Google, ktorá ponúka adaptívny systém smerníc, komponentov a nástrojov. Je založená na pravidlách správneho užívateľského dizajnu, a jej cieľom je ponúknuť vývojárom efektívny a elegantný spôsob vývoja moderne vyzerajúcich a funkčných aplikácií. Momentálne je dostupná verzia 3, ktorá je doplnená o dynamické farby aplikácie a nové nástroje pre správu tém. Vďaka tejto knižnici vyzerajú aplikácie podobnejšie a ich používanie je intuitívnejšie a prehľadnejšie.

V našej aplikácii sa riadime princípmi knižnice Material 3 a využívame jej komponenty, aby sme dosiahli jednoduchý a uniformný výzor. Taktiež dodržiujeme základy správneho užívateľského dizajnu a snažíme sa zapracovať všetky smernice na zachovanie príjemného užívateľského zážitku.

Kapitola 5

5.1 Vytváranie nových kartičiek

V nasledujúcej kapitole popíšeme podrobnosti implementácie systému pridávania nových kartičiek. Pre odfoťenie fotografie používame knižnicu CameraX, pre vybratie fotografie z galérie používame nástroje, ktoré sú obsiahnuté v knižniciach Androidu. Pre orezanie fotografie používame knižnicu Image Cropping Library, a na spracovanie obrazu používame OpenCV.

5.1.1 Použitie kamery mobilného zariadenia

Pri fotení fotografie sme mali viacero možností implementácie. Pôvodne sme mali v pláne použiť knižnicu, ktorá spustí aplikáciu fotoaparátu mobilného zariadenia, no v skorých fázach vývoja sme od tohoto prístupu upustili. Android aktuálne povzbudzuje vývojárov k používaniu knižnice CameraX, ktorá je vyvíjaná pod záštitou Google. Jej najväčšou výhodou je možnosť prispôsobenia všetkých aspektov, od veľkosti zobrazovania náhľadu po vlastné tlačidlá.

Po odfoťení sa fotografia uloží do zariadenia a ďalej pracujeme s URI odkazom súboru. URI, po slovensky Jednotný identifikátor prostriedku je reťazec znakov slúžiaci na pomenovanie zdroja. Tento odkaz následne posielame do upravovacej časti aplikácie. Po upravení a uložení fotografie do databázy sa tento súbor vymaže z pamäte.

5.1.2 Vybratie fotografie z galérie

Vyberanie fotografie z galérie bolo v našom prípade veľmi priamočiare a nenarazili sme na žiadne komplikácie pri implementácii. Nástroje obsiahnuté priamo v základných knižniciach Androidu nám umožnili implementovať možnosť vybratia obrázka v súlade s dizajnovými princípmi Material 3.

Po označení obrázka aplikácia získa URI odkaz na súbor, a ako pri fotografovaní, sa tento odkaz pošle do upravovacej časti aplikácie. Pri tomto spôsobe ale neriešime vymazávanie obrázka, keďže sa nejedná o dočasný súbor slúžiaci len pre naše potreby.

5.2 Automatické orezanie fotografie na báze významných oblastí

Jednou z hlavných funkcionalít aplikácie je navrhnutie automatického orezania fotografie na báze počítačového videnia, významných oblastí obrázka a rozpoznávania tvárí. Knižnica OpenCV disponuje modulmi pre spracovanie obrázka pomocou počítačového videnia pomocou dvoch rôznych algoritmov. V nasledujúcich podkapitolách popíšeme náš postup, problémy, na ktoré sme narazili, výslednú implementáciu a porovnanie algoritmov na spracovanie obrazu.

5.2.1 Natívne implementovanie OpenCV

Prvotný plán spočíval v implementácii knižnice OpenCV natívne. Knižnica disponuje verziou dostupnou pre vývoj na zariadenia s operačným systémom Android, no po jej implementovaní do projektu sme zistili, že jej obsah je zredukovaný oproti plnej verzii a funkcionalita, ktorú sme my vyžadovali, nebola dostupná.

Programovací jazyk Kotlin je založený na jazyku Java, a povoľuje nekonfliktné použitie oboch jazykov v jednom projekte. Knižnica OpenCV je taktiež dostupná aj vo verzii pre jazyk Java, no narazili sme na rovnaký problém ako pri implementácii natívne do jazyka Kotlin - obsah knižnice je zredukovaný a nedisponuje funkcionalitami, ktoré vyžadujeme.

Implementácia knižnice natívne teda neprichádzala do úvahy, a bolo potrebné nájsť či už alternatívu pre knižnicu OpenCV, alebo nájsť alternatívu implementácie

5.2.2 Implementovanie knižnice OpenCV pomocou knižnice JavaCPP

Pri hľadaní alternatívy ku OpenCV sme narazili na knižnicu JavaCPP, vyvíjanú spoločnosťou Bytedeco. Ich cieľom je vytvoriť chýbajúce prepojenie medzi programovacími jazykmi Java a C++, ktoré zaručí možnosť používania náročnejších výpočtových technológií z oblasti multimédií, počítačového videnia z platformy C++ na platforme Java. [1]

Ako prvé sme túto knižnicu vyskúšali implementovať natívne do nášho projektu. Narozdiel od predošlých pokusov, Bytedeco obsiahlo do ich knižnice celú funkcionalitu OpenCV, spolu s modulmi pre spracovanie obrazu. Implementácia však neprebíhala podľa plánov - niekoľkokrát sme narazili na problémy pri pridávaní knižnice do projektu, a to z dôvodov chýbajúcich závislostí, komplikovaného postupu a nezrovnalostiach vo verziách. Napriek týmto komplikáciám sa nám knižnicu implementovať podarilo, no problémy nastali pri jej používaní. Či už sme knižnicu používali v jazyku Java alebo Kotlin, v oboch prípadoch sme narazili na problémy s alokáciou pamäte.

Keďže knižnica pracuje s C++ kódom, a nieje vyvíjaná ako oficiálne prepojenie medzi platformami, správa pamäte nebola dostatočne implementovaná a naše výpočty nie len trvali dlho, ale po čase zahltali celú pamäť zariadenia a aplikácia spadla.

Po komunikácii s vývojárom z Bytedeco nám bolo navrhnutých niekoľko možných riešení, no ani jedno nebolo dostatočne spoľahlivé na to, aby sme ho v našej aplikácii použili. Museli sme teda od tejto implementácie upustiť.

5.2.3 Implementovanie knižnice OpenCV pomocou knižnice chaquopy

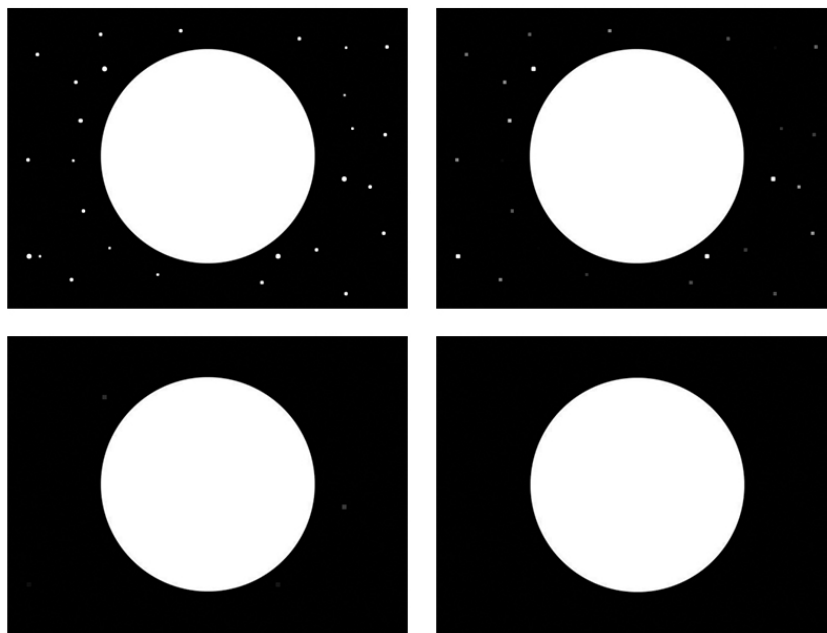
Po neúspechoch v implementácii natívne v jazyku Kotlin, prípadne Java, sme začali hľadať alternatívy v iných oblastiach. Knižnica chaquopy na prvý pohľad pôsobila ideálne vo všetkých smeroch - jednoduché volanie Python kódu v rámci jazyku Kotlin, podpora pre väčšinu knižníc dostupných pre Python, priamočiaru implementáciu, no hlavne, podporu knižnice OpenCV, ktorej vydanie pre Python obsahuje všetky moduly, ktoré vyžadujeme.

Jediný problém, na ktorý sme pri pracovaní v prepojení s jazykom Python natrafili bol pri posielaní obrázka medzi jazykmi. V aplikácii pracujeme s obrázkami vo formáte *Bitmap*, a Python skript potrebuje dostať identický obrázok v rovnakom rozlíšení a farbách. Zvolili sme kompresiu bitmapy do formátu Base64, ktorú posielame ako parameter do volania funkcie z jazyka Python. Tam sa obrázok dekoduje pomocou knižnice *numpy*, a následne sa spracuje knižnicou OpenCV. Pre zrýchlenie procesu spracovania sme taktiež zvolili zmenšiť obrázok, ak je jeho šírka väčšia ako 1500 pixelov - obrázok pred poslaním zmenšíme o polovicu, a časy na spracovanie sa nám tak zmenšili približne o dvojnásobok (bez zmenšenia trvalo spracovanie obrázku z kamery - spravidla 3000 pixelov na šírku - priemerne 10 sekúnd, no po zmenšení je obrázok spracovaný do 5 sekúnd).

Pre získanie automatického orezanie zlučujeme použitie dvoch operácií - vyhodnotenie významných oblastí a detekcia tvárí.

5.2.4 Spracovanie významných oblastí obrázka

Po získaní mapy významných oblastí na ňu aplikujeme rozmazanie typu *Gaussian* a vytvoríme *threshold* mapu, s ktorou budeme pracovať naďalej. Pre úpravu obrazu použijeme dva typy morfologických operácií - Opening (*Otvorenie*) a Dilation (*Rozšírenie*). Opening vyžaduje vytvorenie tzv. kernelu = malého objektu rozličného tvaru, ktorý postupne prejde celým obrázkom a aplikuje nejakú matematickú operáciu postupne na všetky pixely obrázka. Operácia Opening slúži na odstránenie šumu - čím väčší kernel je použitý, tým viac malých artefaktov odstránime (*vid'. obrázok č. 3.1*).



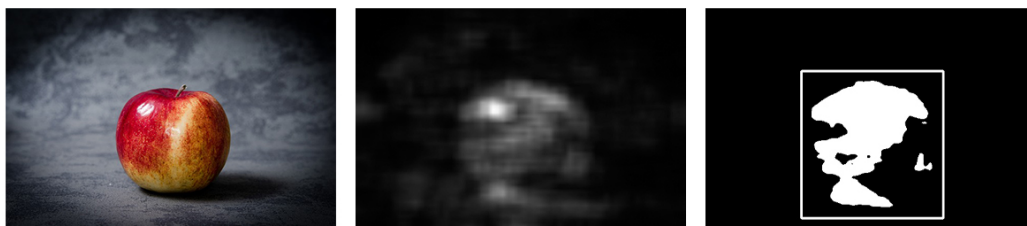
Obr. 5.1: Ilustrácia aplikovania operácie "Opening" na odstránenie šumu s rôznymi veľkosťami kernelu - zľava doprava pôvodný obrázok, 3x3, 5x5 a 7x7

Na druhú stranu, operácia Dilation slúži na rozšírenie objektov v popredí, alebo na zjednotenie rozbitých častí obrazu. Narozdiel od operácie Opening, kde sa používajú kernely, tu efekt násobíme, a väčšinou aplikujeme niekoľko iterácií - čo je priamo úmerné s výsledným efektom - čím je väčší, tým je rozšírenie viditeľnejšie (*vid'. obrázok č. 3.2*)



Obr. 5.2: Ilustrácia aplikovania operácie "Dilation" na rozšírenie objektov. Zľava doprava, od pôvodného obrázku po 3 iterácie operácie.

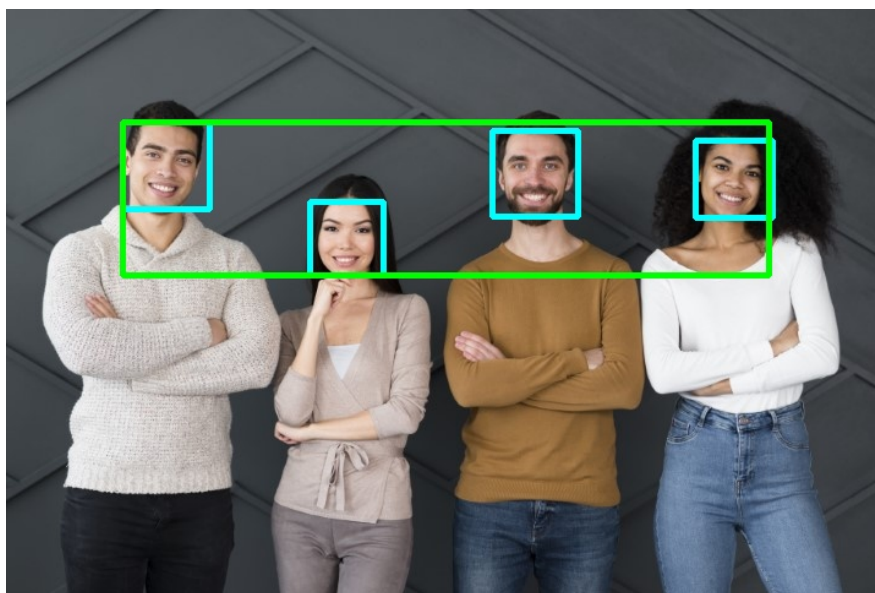
Po aplikovaní všetkých potrebných úprav pre získanie potrebného formátu obrázku spustíme algoritmus, ktorý identifikuje obrysy, uloží krajné súradnice, s ktorými budeme pracovať v ďalšom kroku.



Obr. 5.3: Postup pri spracovaní významných oblastí obrázka. Zľava doprava - pôvodný obrázok, mapa významných oblastí, aplikovanie efektov a nájdenie obrysov

5.2.5 Rozpoznávanie tvárí v obrázku

Na rozpoznávanie tvárí v obrázku používame Viola-Jones algoritmus, ktorý používa Haarove kaskády (popísané v prvej kapitole práce). Narozdiel od spracovania významných oblastí, nepotrebujeme pred spustením algoritmu pripravovať obrázok do potrebnej podoby, stačí nám ho len prekonvertovať do čiernobiela. Po získaní všetkých tvárí na fotografii výsledné súradnice zjednotíme, a teda získame súradnice jedného veľkého obdĺžnika, ktorý obsahuje všetky nájdené tváre.



Obr. 5.4: Zvýraznenie rozpoznaných tvárí v obrázku (tyrkysové obdĺžniky) a zjednotenie výsledkov do jednej selekcie obsahujúcej všetky.

5.2.6 Zjednotenie výsledkov spracovania obrazu

Vo výsledku oboch algoritmov dostaneme dve oblasti, na báze ktorých vieme orezať obrázkov - jedna je na báze počítačového videnia, a druhá je na báze rozpoznávania tvárí. Tieto oblasti treba zjednotiť, kde sme zvolili podobný princíp ako pri zjednocovaní obdĺžnikov pri rozpoznávaní tvárí. Výslednú selekciu teda tvorí zjednotenie dvoch menších, a jej súradnice posielame naspäť do jazyka Kotlin vo formáte jednoduchého listu. Keďže naše kartičky majú jednotný formát - pomer strán 6 ku 7 - musíme súradnice dodatočne prispôbiť. Po získaní listu zistíme, či je orez na šírku alebo na výšku, a následne pridáme pixely do výšky, prípadne do šírky, aby sme zachovali jednotný pomer strán. Upravené súradnice aplikujeme do orezávacieho okna po tom, ako užívateľ stlačí tlačidlo *"Auto crop"*.

5.3 Export kartičiek do formátu PDF

V nasledujúcej podkapitole popíšeme postup pri vytváraní PDF dokumentu pre tlač, ako vytvárame súbor, ukladanie obrázkov a zdieľanie súboru.

5.3.1 Vytvorenie PDF súboru

Pre vytvorenie a správu PDF dokumentu používame natívnu Android knižnicu *Pdf-Document*, ktorá je k dispozícii v základe. Pomocou tejto knižnice vieme vytvárať nové súbory, definovať im strany a pomocou "plátna" (*canvas*) vieme do vytvorených strán aj kresliť. Pri vytvorení dokumentu máme v základe jednu stranu, a postupným prechádzaním obrázkov vieme dodatočne pridávať ďalšie. Po uložení všetkých obrázkov dokument uložíme do pamäte zariadenia, kde využívame základné umiestnenie *Documents*, a hneď po uložení môže užívateľ dokument zdieľať, a to tiež za pomoci natívneho modulu zabudovaného priamo do Androidu. Tento modul pracuje s odkazom URI, ktorý vie použiť a rozposlať dokument do všetkých podporovaných aplikácií.

5.3.2 Uloženie obrázkov do PDF súboru

Keďže sú naše obrázky uložené v jednotnom tvare, vieme si predom definovať presné súradnice umiestnenia v dokumente, bez toho, aby sme tieto výpočty museli vykonávať pri každom jednom exportovaní dokumentu. Pomocou "plátna" (*canvas*) si na každú stranu nakreslíme pomocné čiary, slúžiace na ohraničenie každej kartičky a jednoduchšie strihanie, postupne prechádzame zoznam uložených kartičiek, pričom ich ukladáme do dokumentu. Popis a obrázok nemusíme brať zakaždým z databázy, keďže na hlavnej stránke sú obrázky predom načítané pre užívateľa.

5.4 Testovanie

Z dôvodu cielenia našej aplikácie na ľudí, ktorí nemusia byť s technológiami zruční, bolo potrebné vykonať testovanie užívateľského rozhrania a funkcionality s cieľom zistenia, či sa nám podarilo dodržať požiadavky na aplikáciu. V nasledujúcich podkapitolách popíšeme testovací scenár, postup testovania, jeho výsledky a zmeny, ktoré sme do aplikácie museli na základe výsledkov zapracovať.

5.4.1 Testovací scenár

Pri testovaní aplikácie sme zvolili taký scenár, ktorý otestuje všetky funkcionality aplikácie, a zároveň overí, či je rozhranie dostatočne jednoduché a intuitívne. Testovacím subjektom sme neukázali video návod aplikácie - popísali sme im stručne aplikáciu, prípadne vysvetlili systém PECS, a následne im ukázali scenár úkonov, ktoré musia splniť. Počas testovania sme nijako nezasahovali do interakcie užívateľa s aplikáciou a nijako sme im nepomáhali v splnení scenáru.

Scenár obsahoval následné body:

1. Vytvoriť kartičku pomocou fotoaparátu
2. Pridať popis
3. Uložiť kartičku
4. Vytvoriť kartičku pomocou fotogalérie
5. Použiť automatické orezanie
6. Uložiť kartičku
7. Vytvoriť PDF dokument s použitím 2 vytvorených kartičiek tak, aby sa na jednu stranu zmestilo 8 kartičiek
8. Zazdieľať dokument pomocou mailu
9. Vymazať všetky kartičky

5.4.2 Výsledky testovania

Prvý testovací subjekt objavil hneď niekoľko závažných chýb v programe - medzi stlačením tlačidla na odfotenie fotografie a ukázaním obrazovky na úpravu kartičky je niekoľkomilisekundová odozva, a ak užívateľ "odfotí" fotografiu niekoľkokrát, obrazovka na úpravu kartičky sa ukáže duplicitne a fotografia v galérii sa nevymaže. Zapracovali sme teda vibračnú odozvu po odfotení, a taktiež sme tlačidlo "zatmavili"

a znemožnili sme jeho duplicitné stlačenie. Druhá chyba, ktorú subjekt objavil, sa vyskytla pri vytváraní dokumentu - ak užívateľ stlačí tlačidlo na vytvorenie dokumentu, a v dialógovom okne nechá zvolený počet 8 (ktorý tam je nastavený ako základný), tak aplikácia spadne. Dokument sa totižto začne vytvárať bez toho, aby sme zadefinovali, aký má byť počet kartičiek na jednu stranu - užívateľ nezvolil na možnosť, a tým pádom aplikácia pracovala s číslom 0, čo viedlo k chybe. Všetky nasledujúce testovania sme vykonávali s opravenými chybami a zapracovanými pripomienkami od prvého testovacieho subjektu.

Záver

Literatúra

- [1] Bytedeco - home. <http://bytedeco.org/>. Accessed: 29.4.2023.
- [2] Camerax - overview. <https://developer.android.com/training/camerax>. Accessed: 19.3.2023.
- [3] Card talk - google play store. <https://play.google.com/store/apps/details?id=jp.co.litalico.cardtalk>. Accessed: 6.2.2023.
- [4] Chaquopy - about. <https://chaquo.com/chaquopy/doc/current/>. Accessed: 7.2.2023.
- [5] Opencv - about. <https://opencv.org/about/>. Accessed: 7.2.2023.
- [6] Pecs - about. <https://pecsusa.com/pecs/>. Accessed: 9.2.2023.
- [7] Room - about. <https://developer.android.com/training/data-storage/room>. Accessed: 19.3.2023.
- [8] Symbotalk - aac talker - google play store. <https://play.google.com/store/apps/details?id=com.ellelad.comboard>. Accessed: 6.2.2023.
- [9] Talk in pictures x - aac speak - google play store. <https://play.google.com/store/apps/details?id=com.itoucantalk>. Accessed: 6.2.2023.
- [10] Petr Adamus, Alica Vančová, and Monika Löfflerová. *Poruchy autistického spektra v kontextu aktuálních interdisciplinárních poznatků*. Ostravská univerzita, Pedagogická fakulta, 2017.
- [11] D American Psychiatric Association, American Psychiatric Association, et al. *Diagnostic and statistical manual of mental disorders: DSM-5*, volume 5. American psychiatric association Washington, DC, 2013.
- [12] Miroslava Bartoňová, Barbora Bazalová, and Jarmila Pipeková. *Psychopedie. Texty k distančnímu vzdělávání*. Paido, 2007.
- [13] Andy Bondy and Lori Frost. The picture exchange communication system. *Behavior Modification*, 25(5):725–744, 2001.

- [14] Jill Boucher. Autism spectrum disorder: characteristics, causes and practical issues. *Autism Spectrum Disorder*, pages 1–368, 2017.
- [15] Derek Bradley and Gerhard Roth. Adaptive thresholding using the integral image. *Journal of graphics tools*, 12(2):13–21, 2007.
- [16] Věra Čadilová and Zuzana Žampachová. *Pacient s poruchou autistického spektra v ordinaci lékaře*. APLA, 2012.
- [17] Geraldine Dawson. *Autism: Nature, Diagnosis, and Treatment*. Guilford Publications, 1989.
- [18] Eric Hollander, Randi Hagerman, and Deborah Fein. *Autism Spectrum Disorders*. American Psychiatric Association Publishing, 2018.
- [19] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2007.
- [20] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.
- [21] Theo Peeters and Christopher Gillberg. *Autism: Medical and Educational Aspects*. PORTÁL sro, 2008.
- [22] Ali Sharifara, Mohd Rahim, Mohd Shafry, and Yasaman Anisi. A general review of human face detection including a study of neural networks and haar feature-based cascade classifier in face detection. In *2014 International symposium on biometrics and security technologies (ISBAST)*, pages 73–78. IEEE, 2014.
- [23] Elena Sikudova, Zuzana Cernekova, Wanda Benesova, Zuzana Berger Haladova, and Julia Kucerova. *Počítačové videnie: Detekcia a rozpoznávanie objektov. - 1. vydanie*. 01 2014.
- [24] William C Stokoe. Sign language structure. *Annual review of anthropology*, 9(1):365–390, 1980.
- [25] Kateřina Thorová. *Poruchy autistického spektra: Rozšířené a přepracované vydání*. PORTÁL sro, 2006.
- [26] Yi-Qing Wang. An analysis of the viola-jones face detection algorithm. *Image Processing On Line*, 4:128–148, 2014.

- [27] Lorna Wing and Judith Gould. Severe impairments of social interaction and associated abnormalities in children: Epidemiology and classification. *Journal of autism and developmental disorders*, 9(1):11–29, 1979.
- [28] Jinan Zeidan, Eric Fombonne, Julie Scolah, Alaa Ibrahim, Maureen S Durkin, Shekhar Saxena, Afqah Yusuf, Andy Shih, and Mayada Elsabbagh. Global prevalence of autism: A systematic review update. *Autism Research*, 15(5):778–790, 2022.
- [29] Kateřina Šrahůlková. *Žák s poruchou autistického spektra a jeho fungování v kolektivu běžné třídy. Speciální pedagogika*, 28(1),. Praha : Pedagogická fakulta Univerzity Karlovy, 2018.