

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**VYTVORENIE MOBILNEJ APLIKÁCIE URČENEJ
NA VYHĽADÁVANIE SIVÉHO ZÁKALU U
POUŽÍVATEĽA**

TÍMOVÝ PROJEKT

2023

Nošík D., Ištók P., Mičuda R., Vrbovský A., Kopecký M., Danko T.

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**VYTVORENIE MOBILNEJ APLIKÁCIE URČENEJ
NA VYHĽADÁVANIE SIVÉHO ZÁKALU U
POUŽÍVATEĽA**
TÍMOVÝ PROJEKT

Študijný program: Aplikovaná informatika
Názov studijného odboru: Informatika
Školiace pracovisko: Ústav informatiky a matematiky
Vedúci tímového projektu: MUDr. Veronika Kurilová, PhD.

Bratislava 2023

Nošík D., Ištók P., Mičuda R., Vrbovský A., Kopecký M., Danko T.

Podčakovanie

Radi by sme sa podčakovali našej vedúcej práce, MUDr. Veronike Kurilovej, PhD., za užitočné rady a pripomienky, ktoré nám poskytla počas našej uplynulej spolupráce.

Obsah

Úvod	1
1 Analýza	2
1.1 Anatómia oka	2
1.1.1 Segmenty oka	5
1.2 Sivý zákal	5
1.2.1 Príznaky	6
1.2.2 Príčiny vzniku	7
1.2.3 Diagnostika	8
1.2.4 Liečba	10
1.3 Prehľad existujúcich prác	10
1.4 Existujúce datasety	11
1.4.1 Retina Dataset	11
1.4.2 Cataract Surgery Dataset	12
1.4.3 Ocular Disease Intelligent Recognition (ODIR) dataset	13
1.4.4 Retinal Fundus Multi-Disease Image Dataset (RFMID)	13
1.5 Červený reflex	14
2 Experimenty	16
2.1 Experiment 1 - Fotografia prednou a zadnou kamerou	16
2.2 Experiment 2 - Dávkové snímanie zadnou kamerou	18
2.3 Experiment 3 - Video zadnou kamerou	18
2.4 Testovanie experimentu 3	19
3 Návrh	21
3.1 Diagramy	21
3.1.1 UML diagram prípadov použitia	22
3.1.2 Vývojový diagram mobilnej aplikácie	22
3.1.3 Diagram aktivít servera	23
4 Použité technológie	25
4.1 Klient	25
4.1.1 CameraX	25
4.1.2 Exoplayer	26
4.1.3 Retrofit	26

4.2	Server	27
4.2.1	Flask	27
4.2.2	OpenCV (Open Source Computer Vision Library)	27
4.2.3	Dlib	27
4.2.4	CNN	27
5	Implementácia	29
5.1	Server	29
5.1.1	Rozpoznávanie červenej zrenice	31
5.2	Klient	34
5.2.1	UI klienta	35
5.2.2	Backend klienta	38
6	Zhodnotenie výsledkov	40
	Zoznam použitej literatúry	42
	Prílohy	I
	A Návod na spustenie aplikácie	II
	B Zápisnice zo zimného semestra	III
	C Zápisnice z letného semestra	XIV

Zoznam obrázkov a tabuliek

Obrázok 1	Časti ľudského oka znázornené na priereze oka [2]	2
Obrázok 2	Sietnica [3]	4
Obrázok 3	Predný segment oka [4]	5
Obrázok 4	Prierez oka [5]	5
Obrázok 5	Priebeh katarakty [6]	6
Obrázok 6	Príklad katarakty [7]	7
Obrázok 7	Snellenove tabuľky [8]	8
Obrázok 8	Vyšetrenie štrbinovou lampou (na obr. vľavo sa nachádza katarakta)[9]	9
Obrázok 9	Katarakta viditeľná pomocou retroiluminácie[10]	9
Obrázok 10	Fotografia z triedy očí postihnutých kataraktou z datasetu Retina [15]	12
Obrázok 11	Fotografia zobrazujúca operáciu katarakty z Cataract Surgery datasetu [16]	13
Obrázok 12	Fotografia katarakty z ODIR datasetu [17]	13
Obrázok 13	Ukážková vzorka z RFMID [18]	14
Obrázok 14	Geometria červeného reflexu očí [19]	14
Obrázok 15	Fotografia s červeným reflexom očí [20]	15
Obrázok 16	Experiment 1 - zachytenie červeného reflexu prednou kamerou	17
Obrázok 17	Experiment 1 - zachytenie červeného reflexu zadnou kamerou	17
Obrázok 18	Experiment 2 - zachytenie červeného reflexu zadnou kamerou pomocou dávkového snímania	18
Obrázok 19	Experiment 3 - extrahovaná snímka z videa s červeným reflexom	19
Obrázok 20	a) svetlo za subjektom, b) svetlo z boku subjektu, c) snímanie v šere, d) snímanie v tme	19
Obrázok 21	a) kamera pod subjektom, b) kamera nad subjektom, c) kamera priamo na oči subjektu	20
Obrázok 22	a) vzdialenosť 40 cm, b) vzdialenosť 70 cm, c) vzdialenosť 110 cm	20
Obrázok 23	UML diagram prípadov použitia	22
Obrázok 24	Flowchart diagram mobilnej aplikácie	23
Obrázok 25	Diagram aktivít - spracovanie videa	24
Obrázok 26	Rozloženie tvárových bodov [31]	29

Obrázok 27	Ukážka detekcie na jednej zo snímok videa	30
Obrázok 28	Výrez oblasti pravého oka	30
Obrázok 29	Výrez oblasti ľavého oka	30
Obrázok 30	Ukážka datasetu - 16 obrazov so zrenicou s červeným reflexom (vľavo), 16 obrazov so zrenicou s nečerveným reflexom (vpravo) .	31
Obrázok 31	Ukážka použitia Google Teachable Machine v našom konkrétnom prípade	32
Obrázok 32	Grafy zachytávajúce presnosť (vľavo) a stratu (vpravo) v závislosti od epochy počas trénovalia	32
Obrázok 33	Konfúzna matica klasifikačného modelu	33
Obrázok 34	Obrazy ľavej zrenice z videa nového testovacieho subjektu . . .	34
Obrázok 35	a) obrazovka nahrávania videa, b) inštrukcie pre úspešné nahratie	35
Obrázok 36	a) obrazovka prehrávania videa, b) inštrukcie	36
Obrázok 37	Obrazovka odosielania videa	37
Obrázok 38	Obrazovka so zobrazeným výsledkom	38
Obrázok 39	Červený reflex u diagnostikovaného pacienta	41
Tabuľka 1	Presnosť klasifikácie katarakty vybranými modelmi (upravené z [14])	11
Tabuľka 2	Úspešnosť nášho modelu na našej testovacej množine	34

Zoznam skratiek

API	Application programming interface
CNN	Convolutional Neural Networks
HD	High-definition
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
ODIR	Ocular Disease Intelligent Recognition
OS	Operačný systém
REST	Representational State Transfer
RFMID	Retinal Fundus Multi-Disease Image Dataset
XML	eXtensible Markup Language

Úvod

Oko sa považuje za jeden z najdôležitejších zmyslových orgánov, pretože ním človek získava približne 80 % informácií zo svojho okolia. Oči sú orgán, ktorý reaguje na svetlo a umožňuje ľuďom orientovať sa v priestore, vnímať tvary a farby. Avšak, oči sú komplexným orgánom, ktorý môže byť postihnutý mnohými ochoreniami. Aj menšie poškodenia zraku môžu mať vplyv na kvalitu života človeka. Včasné diagnostika a liečba očných ochorení môžu zmierniť poškodenia a predchádzať úplnej strate zraku.

Ludské oko je nástrojom videnia, ktorý zachytáva obrazy a prenáša ich do mozgu. Šošovka, ktorá sa nachádza v prednej časti oka, je kľúčovou štruktúrou pre videnie, pretože vďaka svojej optickej mohutnosti pomáha koncentrovať lúče svetla na sietnicu. Ochorenia šošovky môžu spôsobiť stratu zraku, preto je dôležité ich včas identifikovať a liečiť.

Sivý zákal je najčastejšie ochorenie šošovky. Prejavuje sa postupným zakalením šošovky, ktoré môže viesť k zhoršeniu videnia, rozmazanému obrazu, zmenšenému vnímaniu farieb, až po úplnú stratu zraku. Môže byť spôsobené starnutím, genetickými faktormi, žiareniom, chorobami a nadmerným užívaním alkoholu. Diagnózu stanovuje oftalmológ pomocou vyšetrenia na štrbinovej lampe. Liečba sa uskutočňuje chirurgicky, pričom sa nahradí zakalená šošovka novou umelou šošovkou.

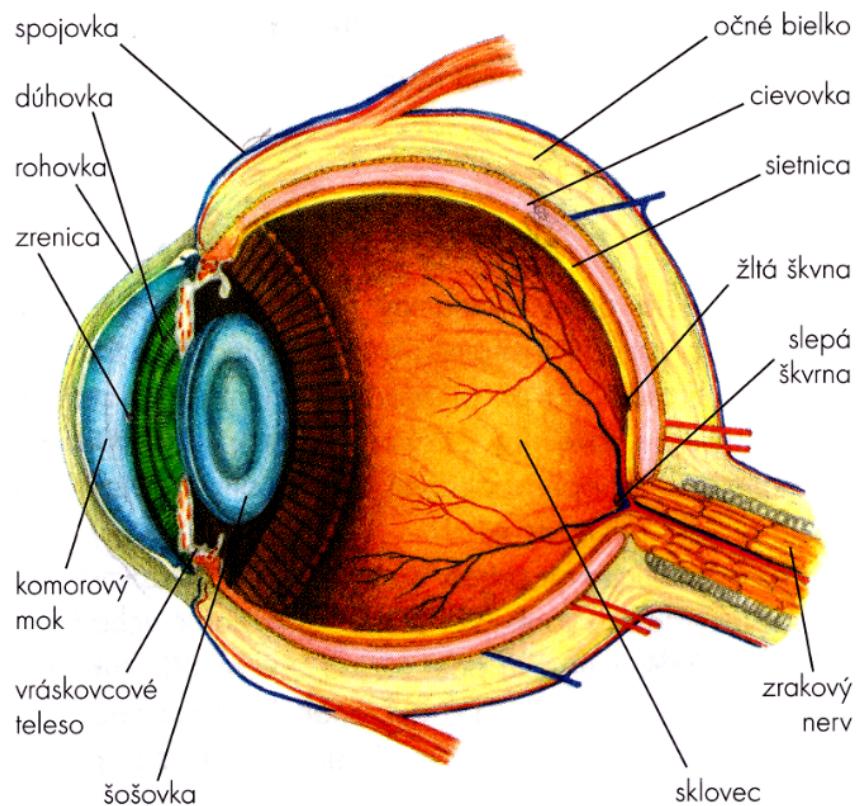
Cieľom tohto tímového projektu je vytvorenie mobilnej aplikácie, ktorá by mohla byť perspektívne použitá na vyhľadávanie sivého zákalu u používateľa. Aplikácia by používala zabudovaný fotoaparát a používateľ by si nasnímal svoje oko za rôznych svetelných podmienok a nastavení, ktoré by sa testovali v spolupráci s oftalmológom. Ďalej by sa nasnímaný obraz aplikáciou spracoval a mohol by byť použitý na ďalšie účely. Screeningové vyhľadávanie sivého zákalu pomocou mobilnej aplikácie by takto mohlo byť rýchle, ľahko dostupné a jednoduché na ovládanie.

1 Analýza

V tejto kapitole sa zaoberáme definíciou a bližším vysvetlením základných pojmov nutných k pochopeniu problematiky. Ďalej popisujeme už existujúce práce, ktoré sa zaoberajú danou problematikou a algoritmus na detekciu zrenice. Optimálny algoritmus bude použitý pri implementácii nášho riešenia.

1.1 Anatómia oka

Oko je orgán reagujúci na svetlo. V priebehu vývoja živočíchov došlo k výraznému rozvoju od svetlocitlivých orgánov reagujúcich iba na to, či svetlo je alebo nie je, až po „jednoduché“ oko u stavovcov (vrátane človeka) a hlavonožcov a zložené oko u článkonožcov. Časti ľudského oka sú zobrazené na obrázku č.1[1]:



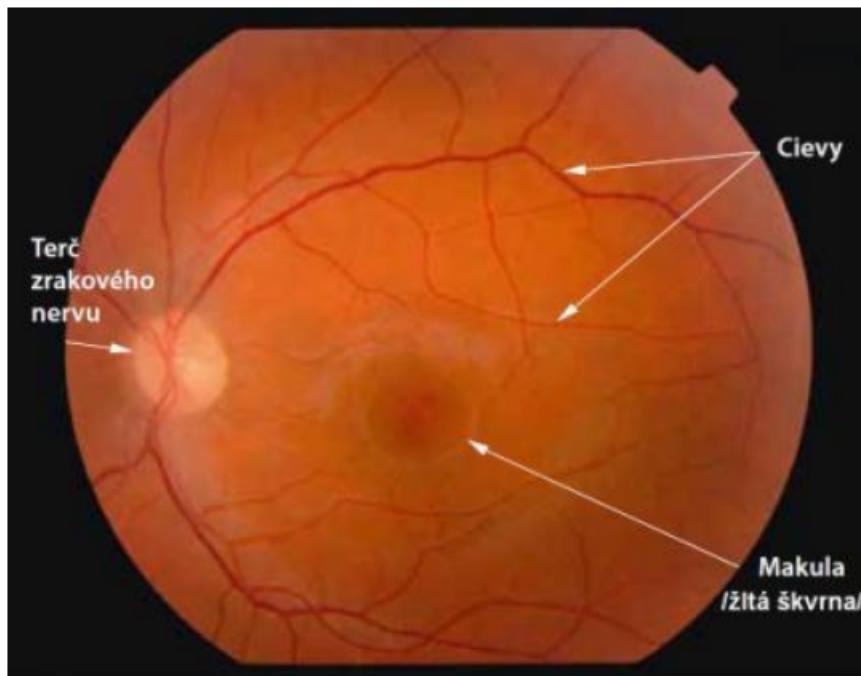
Obr. 1: Časti ľudského oka znázornené na priereze oka [2]

- **Spojovka** je tenká, vlhká sliznica, ktorá pokrýva povrch oka a vnútornú stranu mihalníca. Začína na vonkajšom okraji rohovky, pokrýva viditeľnú časť vonkajšieho

bielka a vystiela vnútorný povrch mihalníc až k ich okraju. Priestor medzi okom a mihalnicami, ktorý pokrýva spojovka, sa nazýva spojovkový vak.

- **Rohovka** je vonkajšia vrstva oka, pokrytá slznou tekutinou, ktorá jej dodáva vlhkosť. Je zapustená do vrstvy známej ako bielko. Má diskovitý tvar a prepúšťa do oka svetlo. Okrem toho chráni oko pred nečistotami, prachom a povrchovými poraneniami. V prirodzenom stave je veľmi pružná a svojím zakrivením zlepšuje optické vlastnosti oka, čo je dôležité pre jasné videnie.
- **Bielko** je hrubšie a pevnejšie ako rohovka čím chráni oko pred poškodením. Pokrýva takmer celé oko až na prednú časť, kde je zapustená rohovka, a zadnú časť, kde sa nachádzajú nervové vlákna.
- **Zrenica** je čierne vyzerajúci otvor uprostred ľudského oka. Je schopná reagovať na intenzitu dopadajúceho svetla a prispôsobiť sa jej. Toto prispôsobenie nie je riadené samotnou zrenicou, ale svalom na vnútornom okraji dúhovky. Veľkosť našich zreníc môže byť ovplyvnená aj našim emocionálnym stavom. Napríklad strach a intenzívna radosť môžu spôsobiť rozšírenie zreníc. Okrem toho, veľkosť zreníc môže byť ovplyvnená aj účinkami alkoholu a drog.
- **Dúhovka** je farebný kruh, obklopuje zrenicu a plní funkciu clony, riadiacej množstvo svetla vstupujúceho do oka. V svetlom prostredí dúhovka zabezpečuje zmenšenie zrenice, čím sa dovnútra prepúšťa menej svetla. Naopak, v tme sa zvierač zrenice otvára a dúhovka sa rozširuje, čím umožní, aby do oka vstúpilo viac svetla. Dúhovka tiež ovplyvňuje farbu našich očí, pričom však táto vlastnosť nemá na zrak žiadny vplyv. Každý človek má jedinečnú štruktúru dúhovky.
- **Šošovka** slúži na zachytávanie svetla, ktoré vstupuje do zrenice, a zabezpečuje tak ostrosť obrazu na sietnici. Je elastická a schopná meniť svoj tvar pomocou ciliárneho svalu. Týmto spôsobom dokáže zaostrovať na objekty v menšej aj väčšej vzdialosti. Keď sa pozérame na blízke objekty, šošovka sa prispôsobuje zakrivením. Pri sledovaní vzdialenejších objektov sa šošovka sploštuje. Šošovka prevracia obraz, ktorý vidíme, a na sietnici ho projektuje obrátene. Správne otočený obraz sa spracúva až neskôr v mozgu.
- **Sklovec** je základnou časťou oka, ktorá vypĺňa jeho vnútorný priestor medzi šošovkou a sietnicou. Je priečladný a 98% jeho štruktúry tvorí voda. Zvyšné 2% obsahujú kyselinu hyalurónovú a kolagénové vlákna.

- **Sietnica** je zodpovedná za spracovanie svetla a farebných podnetov a prenos týchto informácií do mozgu prostredníctvom zrakového nervu. Konvertuje dopadajúce svetlo, ktoré následne spracúva mozog.

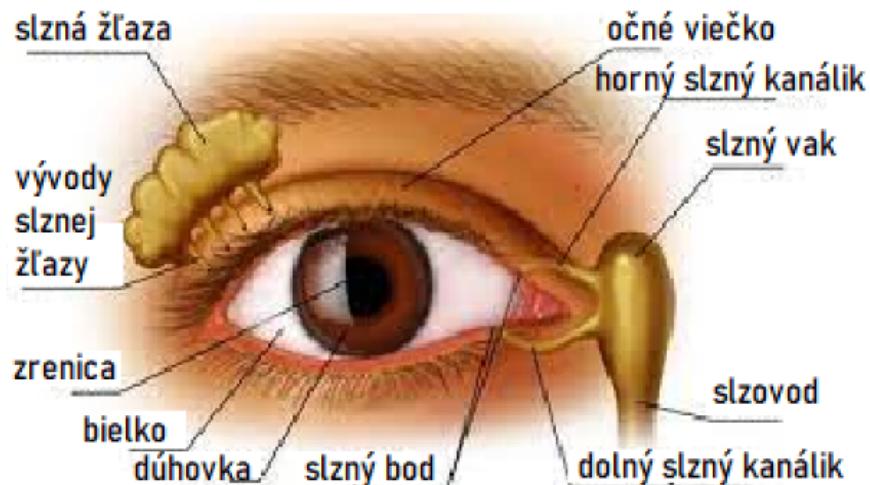


Obr. 2: Sietnica [3]

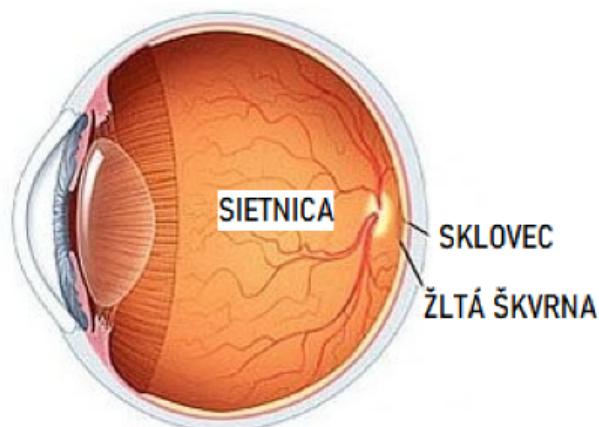
- **Zrakový nerv** je zodpovedný za prenos informácií zo sietnice do mozgu. Skladá sa z približne jedného milióna nervových vlákien, nazývaných axóny. Zrakový nerv vychádza zo sietnice cez bod, nazývaný papila. Tento bod je známy aj ako "slepá škvrna", pretože sietnica na tomto mieste neobsahuje žiadne senzorické bunky.
- **Žltá škvrna** je miesto na sietnici, s najväčšou koncentráciou čapíkov a teda aj najostrejším videním. Obraz, na ktorý človek hľadí kolmo, sa premieta práve na dané miesto.
- **Komorový mok** sa nachádza v prednej a zadnej komore ľudského oka. Obsahuje dôležité živiny pre šošovku a rohovku, a zásobuje ich kyslíkom. Okrem toho má ochrannú funkciu, kedy pomáha oku udržiať si svoj tvar a chrániť ho pred patogénmi.

1.1.1 Segmenty oka

Oko sa delí na predný a zadný segment. Predný segment môžeme vidieť na Obr. č.3 nižšie.



Obr. 3: Predný segment oka [4]

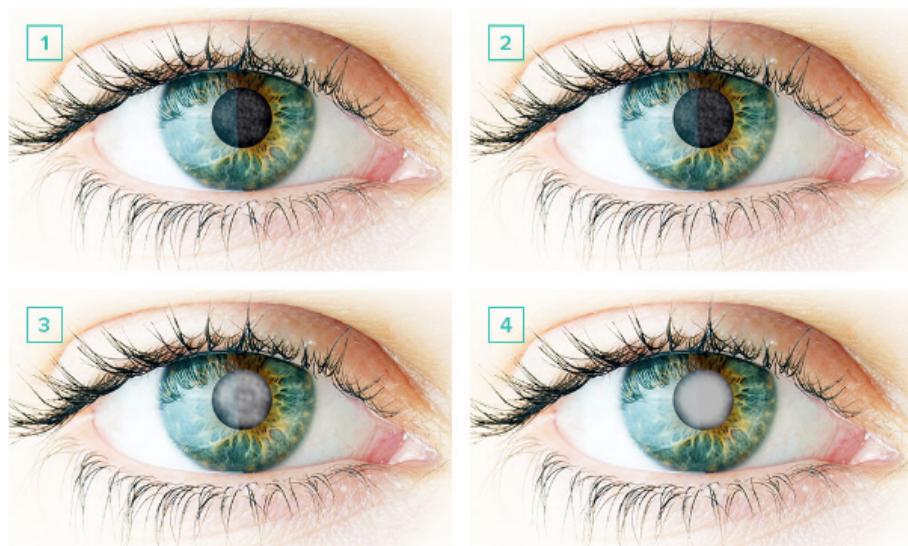


Obr. 4: Prierez oka [5]

1.2 Sivý zákal

Sivý zákal - známy tiež ako Katarakta, je očné ochorenie, pri ktorom dochádza k skaleniu a strate priehľadnosti pôvodne kryštalicky čírej očnej šošovky. Šošovka sa vekom

postupne kalí a farbí dožlta či dosiva. Do oka sa tak nedostáva dostaok svetla potrebného pre tvorbu ostrého a správne farebne interpretovaného obrazu. Sivý zákal sa časom zhoršuje a ak sa neodstráni, môže prerásť až do slepoty. Podľa Svetovej zdravotníckej organizácie (WHO) je katarakta celosvetovo najčastejšou príčinou slepoty s podielom až 51 % zo všetkých prípadov slepoty[6].



Obr. 5: Priebeh katarakty [6]

1.2.1 Príznaky

Najčastejšími príznakmi katarakty sú:

- zhoršenie ostrosti videnia,
- zhoršenie kontrastnej citlivosti,
- zdvojené videnie,
- pokles farebného videnia,
- zhoršené videnie v noci,
- častá zmena dioptrií,
- zvýšená citlosť na svetlo a oslnenie.



Obr. 6: Príklad katarakty [7]

1.2.2 Príčiny vzniku

Vo väčšine prípadov vzniká katarakta prirodzeným starnutím očnej šošovky. Značný stupeň jej zakalenia sa prejavuje u každého druhého človeka vo veku nad 65 rokov a až u približne 70 % ľudí vo veku nad 75 rokov. Rozvinúť sa však môže aj skôr ako v seniorskom veku.

Okrem veku môžu byť príčinou vzniku sivého zákalu tiež:

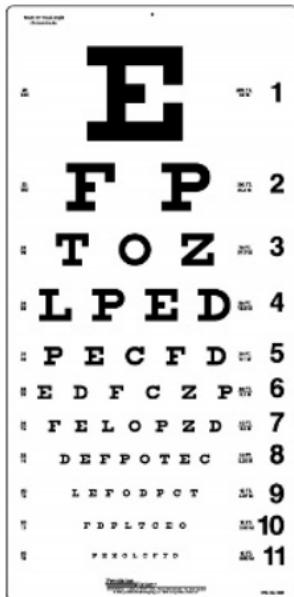
- celkové ochorenia – napr. cukrovka, roztrúsená skleróza či reuma,
- iné očné ochorenia – napr. glaukóm či zápal dúhovky,
- úraz oka,
- nadmerná konzumácia alkoholu,
- slnečné žiarenie,
- fajčenie,
- dlhodobé užívanie niektorých druhov liekov – napr. kortikosteroidov,
- genetika.

1.2.3 Diagnostika

Sivý zákal sa zistuje očným vyšetrením, pri ktorom lekár kontroluje tiež anamnézu a príznaky, ktoré sa prejavujú u pacienta. Pri vyšetrení sa môže použiť niekoľko spôsobov, napríklad:

Vyšetrenie zrakovej ostrosti

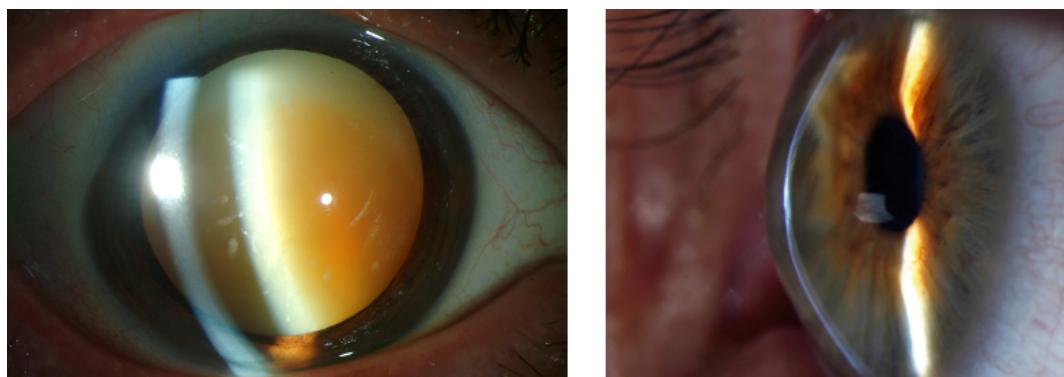
Na test zrakovej ostrosti sa používajú Snellenove tabuľky s písmenami či znakmi, ktoré sú zoradené v riadkoch, pričom každý riadok smerom nadol je s menšími písmenami v porovnaní s predchádzajúcim riadkom.



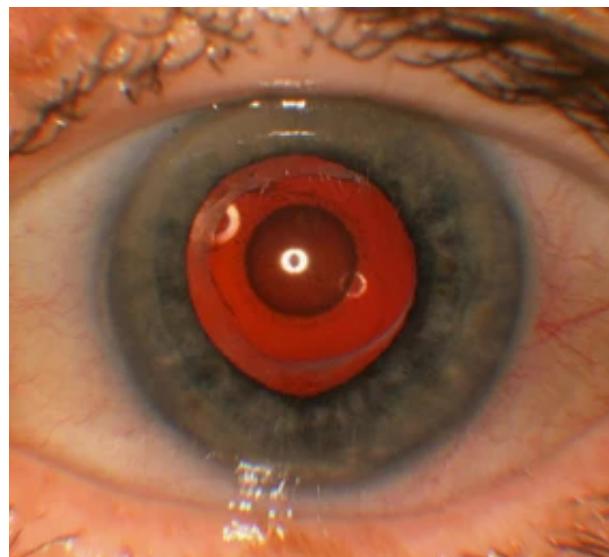
Obr. 7: Snellenove tabuľky [8]

Vyšetrenie štrbinovou lampou

Štrbinová lampa umožňuje očnému lekárovi vidieť zväčšené štruktúry prednej časti oka. Zariadenie využíva intenzívnu líniu svetla - štrbinu, na osvetlenie pacientovej rohovky, dúhovky, šošovky a priestoru medzi dúhovkou a rohovkou. Štrbina umožňuje lekárovi prezerat tieto štruktúry v malých častiach, čo ulahčuje detekciu akýchkoľvek drobných abnormalít.



Obr. 8: Vyšetrenie štrbinovou lampou (na obr. vľavo sa nachádza katarakta)[9]



Obr. 9: Katarakta viditeľná pomocou retroiluminácie[10]

Retroiluminácia

Pri retroiluminácii je na očné pozadie nasmerovaný svetelný lúč, ktorý sa odrazí, a vyvolá tak červený reflex. Pomocou neho je jednoduchšie rozpoznať prítomnosť sivého zákalu v šošovke[11].

Vyšetrenie sietnice

Vyšetrenie zadnej časti oka – sietnice sa vykonáva pomocou štrbinovej lampy alebo oftalmoskopom. Pred samotným vyšetrením je potrebné rozšíriť zreničky, čo sa robí kvapkaním špeciálnych očných kvapiek.

1.2.4 Liečba

Konzervatívna liečba katarakty neexistuje, očné kvapky či cvičenia nemajú na zmierenie ochorenia nijaký účinok. Sivý zákal je možné odstrániť jedine chirurgicky. Vyšetrujúci určí, či je už potrebné sivý zákal operovať, alebo sa bude zatiaľ sledovať. V prípade nevhodne zvoleného odkladania hrozia väznejšie perioperačné i pooperačné komplikácie. Operácia sivého zákalu je pomerne jednoduchým úkonom, ktorý každoročne podstupujú milióny ľudí. Patrí zároveň medzi najbezpečnejšie operácie vôbec. Počas zákroku dôjde k vybratiu zakalenej šošovky a k jej nahradeniu novou, čírou, umelou šošovkou.

1.3 Prehľad existujúcich prác

Mobile Cataract Screening App Using a Smartphone Zu Quan Ik a kolektív [12] skúmali možnosti detektie katarakty z fotografií zachytených smartfónom pomocou červeného reflexu. Pri pokusoch boli využité smartfóny s tromi druhmi blesku (xenónový, LED a dvojitý LED blesk). Zu Quan Ik tvrdí, že pre zachytenie červeného reflexu je dôležité rozšírenie zrenice, trvanie blesku, intenzita svetla, osvetlenie prostredia, uhol kamery a vzdialenosť. Pomocou zadného fotoaparátu bol dosiahnutý najlepší výsledok pri vzdialosti 60 cm v priamom frontálnom uhle (angl. *direct frontal angle*). Predný fotoaparát mal nedostatočné vlastnosti pre vznik červeného reflexu, pričom, ako zdroj svetla bola použitá biela predná obrazovka smartfónu. V tejto práci nie je vypracovaná klasifikácia katarakty, len zachytenie červeného reflexu oka [12].

MTCD: Cataract Detection via Near Infrared Eye Images Pavani Tripathi a kolektív [13] sa zaoberali detekciou katarakty pomocou NIR (angl. *near infrared*) svetelného spektra. Práca navrhuje možnosť využitia NIR kamery pre detekciu katarakty v porovnaní s tradičnou metódou diagnostiky pomocou štrbinovej lampy. NIR je dostupná a minimalizuje potrebnú odbornosť. NIR svetelné spektrum svetla má dlhšie vlnové dĺžky ako viditeľné spektrum [13].

Smartfóny nezachytávajú NIR svetelné spektrum a teda toto riešenie nie je možné realizovať ako mobilnú aplikáciu.

Automated Cataract Grading using Smartphone Images Mona Nasirzonouzi [14] sa zaoberala klasifikáciou katarakty podľa štátia choroby. V práci je použitý dataset *IITD Cataract Mobile Periocular (CMP)*, ktorý obsahuje dvojice fotiek s kataraktou pred operáciou a po operácii bez červeného reflexu. Fotografie boli zachytené mobilným fotoaparátom. V práci bola navrhnutá 4-vrstvová konvolučná neurónová sieť. V práci boli využité CNN (VGG-16, Inception V3, ResNet-101) a SVM klasifikátor nad extrahovanými

príznakmi zo CNN ResNet-101. V tabuľke č.1 sú priemerné presnosti jednotlivých modelov [14].

Tabuľka 1: Presnosť klasifikácie katarakty vybranými modelmi (upravené z [14])

Model	CNN	VGG-16	Inception v3	ResNet-101	SVM s ResNet-101
Priemerná presnosť(%)	84,67	87,64	84,67	89,62	87,14

1.4 Existujúce datasety

Ďalší vývoj aplikácie bude vyžadovať metodiku k rozpoznávaniu sivého zákalu na základe fotografie červeného reflexu nasnímaného pomocou smartfónu. Na realizáciu rozpoznávania by mohla byť využitá technológia strojového učenia. Efektívnymi nástrojmi na rozpoznávanie a klasifikovanie obrázkov sú konvolučné neurónové siete, angl. Convolutional Neural Networks (CNN). Vstupom tejto siete je množina obrázkov (dataset), na ktorých sa navrhnutá sieť natrénuje.

Obrázky vhodné na vytvorenie datasetu pre našu sieť by mali mať tieto vlastnosti:

- záchytenie červeného reflexu očí fotoaparátom s bleskom,
- dostatočné rozlíšenie (aspoň full HD),
- malý odraz blesku v zrenici,
- žiadne prekrytie zrenice.

Obrázky s týmito vlastnosťami by mali byť predspracované, tak aby z nich boli extrahované iba zrenice očí s červeným reflexom. Takto extrahované obrázky vytvoria dataset pre konvolučnú neurónovú sieť. Dataset by mal obsahovať vyvážený počet vzoriek očí postihnutých sivým zákalom a zdravých očí.

V rámci prehľadu problematiky sme sa preto zamerali na naštudovanie dostupných datasetov obrazov, ktoré pomohli iným autorom s diagnostikou sivého zákalu. Zároveň aj popíšeme ich vhodnosť pre riešenie našej problematiky.

1.4.1 Retina Dataset

Retina dataset obsahuje obrázky oka kategorizované do 4 tried:

- zdravé oko,
- oko postihnuté sivým zákalom,

- oko postihnuté zeleným zákalom,
- oko postihnuté ochoreniami sietnice.

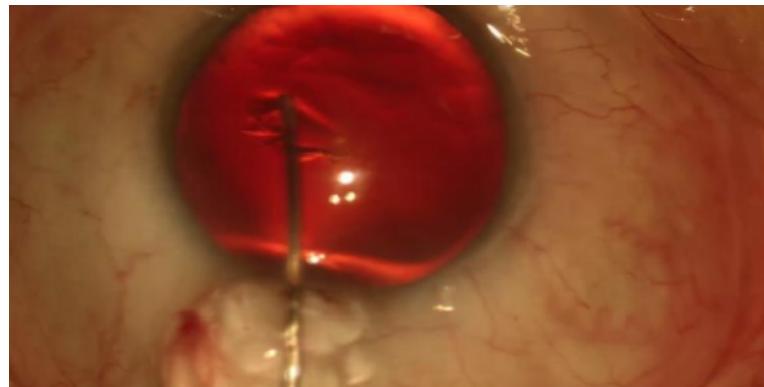
Tento dasaset je pre nás nepoužiteľný, pretože obrázky nespĺňajú nami definované vlastnosti. Konkrétnie tu nie je snímaný červený reflex pomocou fotoaparátu s bleskom, ale pomocou fundus kamery. Prítomnosť sivého zákalu je tak určená vďaka zhoršenej ostrosti obrazu očného pozadia spôsobenej zakalenou šošovkou. Obraz očného pozadia môžeme s mobilným telefónom získať len za pomoci prídavného zariadenia so šošovkou, čo nebolo v oblasti nášho záujmu. Na Obr. 10 je fotografia očí postihnutých jemným sivým zákalom z datasetu Retina.



Obr. 10: Fotografia z triedy očí postihnutých kataraktou z datasetu Retina [15]

1.4.2 Cataract Surgery Dataset

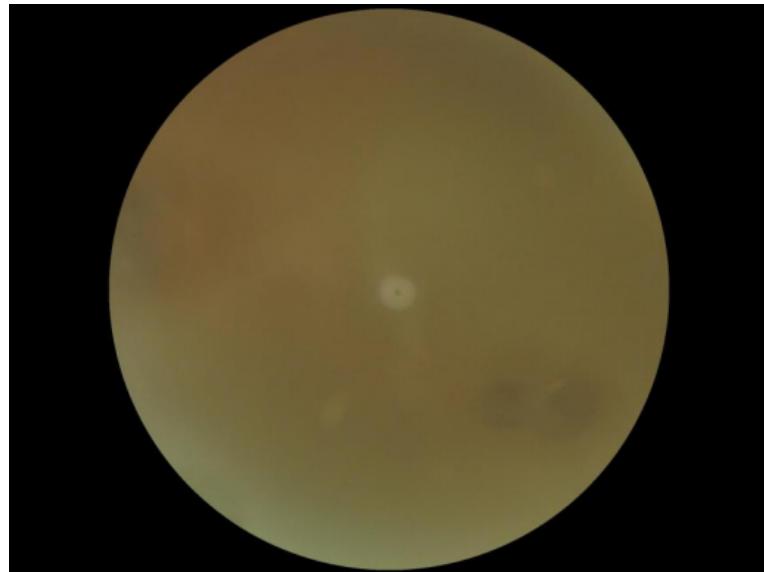
Je to dataset obrázkov a videí, ktoré vznikli počas operácie sivého zákalu. Pôvodne bol tento dataset použitý na vytvorenie neurónovej siete na detekciu nástrojov použitých lekármi pri operácii. Na jednotlivých obrázkoch vidieť červený reflex oka, lenže tieto obrázky nie sú zachytené fotoaparátom smartfónu. Taktiež pri tejto operácii sú zrenice pacienta rozšírené kvapkami, čo nezodpovedá nasnímaniu červeného reflexu za "domácich podmienok" bez použitia rozširujúcich kvapiek. Vo väčšine obrázkov sú zrenice pacienta prekryté nástrojmi lekára. Kvôli týmto faktorom je pre nás táto množina obrázkov nepoužiteľná.



Obr. 11: Fotografia zobrazujúca operáciu katarakty z Cataract Surgery datasetu [16]

1.4.3 Ocular Disease Intelligent Recognition (ODIR) dataset

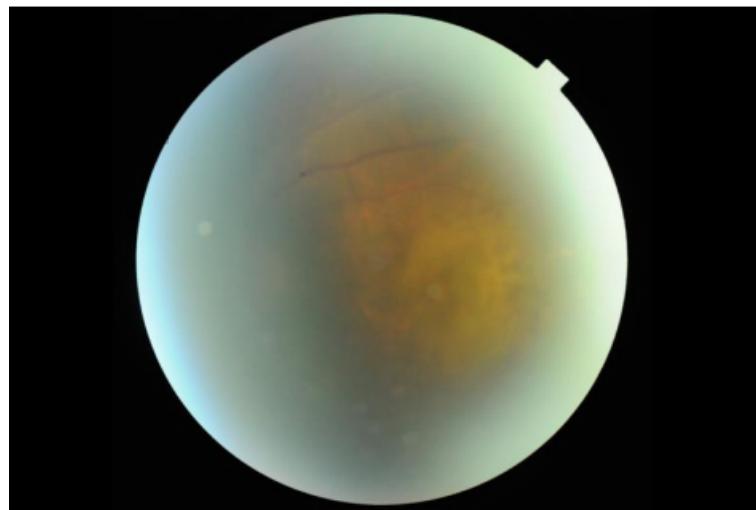
V tomto datasete sú obrázky rozličných ochorení oka a bol zhotovený za účelom klasifikácie týchto chorôb do tried pomocou neurónovej siete. Fotografie zachytávajú očné pozadie a sú zhotovené prostredníctvom fundus kamery, preto tento dataset, rovnako ako pri Retina datasete, nemôžeme použiť v našej neurónovej sieti.



Obr. 12: Fotografia katarakty z ODIR datasetu [17]

1.4.4 Retinal Fundus Multi-Disease Image Dataset (RFMID)

RFMID obsahuje obrázky 46 rôznych ochorení oka. Aj v tomto datasete boli fotografie vyhotovené fundus kamerou, preto ich nevieme využiť.

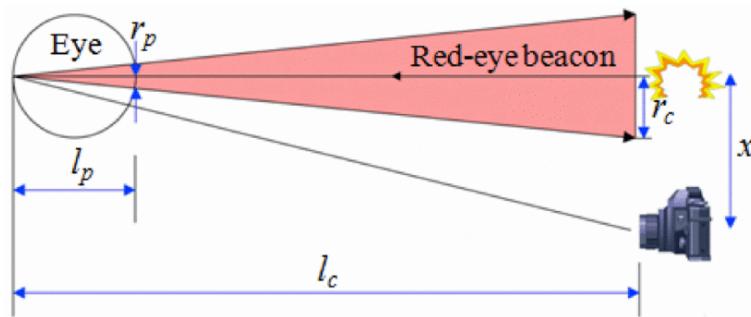


Obr. 13: Ukážková vzorka z RFMID [18]

Z existujúcich datasetov sme nenašli žiadnen, v ktorom by boli obrázky červeného reflexu oka so sivým zákalom a preto by sme ich v ďalšom vývoji aplikácie nemohli použiť.

1.5 Červený reflex

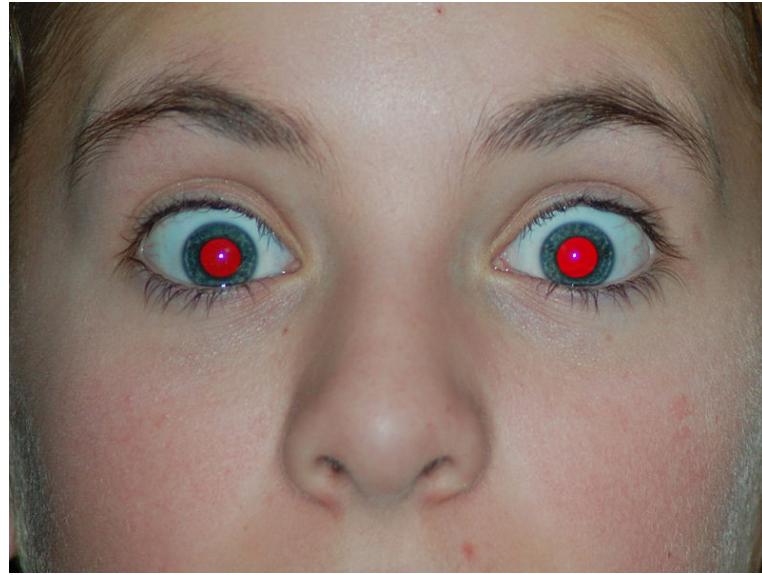
Červený reflex oka je jav, pri ktorom sa svetlo vstupujúce do oka odráža od sietnice v zadnej časti oka a vytvára červenú žiaru. Je to spôsobené krvnými cievami v sietnici, ktoré dodávajú odrazenému svetlu červenú farbu. Tento jav je možné pozorovať aj pri fotografovaní predovšetkým počas tmy, kedy sú zrenice zväčšené. Príklad takejto fotografie je možno vidieť na Obr. 15.



Obr. 14: Geometria červeného reflexu očí [19]

Väčšina fotoaparátov a moderných smartfónov sa snaží efekt červených očí potláčať špecifickým nastavením blesku, využitie preflash, alebo prípadným aplikovaním funkcie na redukciu červených očí (angl. red eyes), využitie postprocessingu. Efekt červených očí

na fotografiách vzniká v dôsledku náhleho rozsvietenia blesku fotoaparátu, ktorému sa zrenice nestihnú včas prispôsobiť.



Obr. 15: Fotografia s červeným reflexom očí [20]

Vzťah 1 opisuje geometrickú situáciu na Obr. 14, v ktorej sa vyskytuje jav červených očí. Svetlo odrazené od oka tvorí svetelný lúč červených očí (angl. red-eye beacon). V prípade, že snímač fotoaparátu sa nachádza vo vnútri svetelného lúča, teda platí vzťah

$$x < r_c/2 \quad (1)$$

kde x je vzdialenosť medzi snímačom fotoaparátu a bleskom, a r_c je polomer svetelného lúča v pozícii fotoaparátu.

Polomer svetelného lúča v pozícii fotoaparátu, r_c je vyjadrený ako

$$r_c = r_p * l_c / l_p \quad (2)$$

kde l_p , l_c a r_p označujú priemer očnej gule, vzdialenosť medzi okom a fotoaparátom, a polomer zreničky. V prípade, že l_c berieme ako konštantu, tak r_c je úmerné l_p a l_c . Zo vzťahov 1 a 2, je možné poznámenať, že efekt červených očí sa vyskytuje najčastejšie, keď x je malé a hodnoty l_p a l_c sú veľké [19].

2 Experimenty

Z prieskumu existujúcich prác na zachytávanie červeného reflexu vieme, že pre jeho nasnímanie je potrebné mať vhodné podmienky. Je potrebné snímať v šerej až tmavej miestnosti, snímať oko z vhodnej vzdialenosťi a mať vhodne nastavenú dĺžku trvania blesku rovnako ako aj intenzitu. Nakoľko nastavenie trvania dĺžky a intenzity blesku v smartfóne je veľmi obtiažne, vykonali sme sériu experimentov, aby sme prišli na vhodnú alternatívnu zachytávania červeného reflexu oka.

2.1 Experiment 1 - Fotografia prednou a zadnou kamerou

Princípom prvého experimentu bolo snímanie červeného reflexu pomocou aplikácie Fotoaparát prednou, aj zadnou kamerou. Experimenty boli vykonané v tmavom prostredí, pričom zdroj svetla bol za snímaním subjektom a kamera smartfónu bola vzdialenosť približne 30 cm od oka.

Pri prvom experimente sme vykonávali snímanie prednou kamerou. Subjekt nasnímal svoje oči smartfónom sám. Predná kamera nezachytila červený reflex, pričom experiment bol realizovaný 20-krát a zdroj svetla bola predná, biela obrazovka. Pri tomto experimente sa vyskytol problém s kvalitou snímok vzhľadom na to, že predná kamera nemala dostatočné rozlíšenie na kvalitné nasnímanie oka. Ďalší problém bol s odrazom svetla od zrenice oka. Ako zdroj svetla slúžila obrazovka smartfónu a na zrenici vznikol veľký odraz svetla. Výsledná snímka je na Obr. 16.



Obr. 16: Experiment 1 - záchytenie červeného reflexu prednou kamerou

Snímanie zadnou kamerou bolo úspešnejšie, avšak inštrukcie na záchytenie červeného reflexu boli pre používateľa zložité a ľahko pochopiteľné. Najskôr bolo potrebné zaznamenať čas, ako dlho svieti blesk v danom prostredí, predtým než sa záhytí snímka. Potom bolo potrebné, aby mal subjekt zatvorené oči počas tejto doby a tesne pred záchytením snímky ich otvoril. Týmto spôsobom sa nám sice červený reflex podarilo záhytiť, ale úspešnosť záchytenia červeného reflexu bola len približne 20%.



Obr. 17: Experiment 1 - záchytenie červeného reflexu zadnou kamerou

2.2 Experiment 2 - Dávkové snímanie zadnou kamerou

Na základe predchádzajúceho experimentu sme zistili, že možnosť nastavovania dĺžky trvania a intenzity blesku nie je nevyhnutná, keďže sme úspešne nahradili túto možnosť inštruovaním subjektu, aby si počas experimentu držal oči zatvorené, pred zachytením červeného reflexu. Preto sme pri druhom experimente použili dávkové snímanie pomocou vstavanej aplikácie smartfónu. Bol využitý zadný fotoaparát, pričom experimentu sa zúčastnil subjekt a asistent, ktorý zachytil subjektov červený reflex. Experiment bol realizovaný v tmavom prostredí. Pred samotným experimentom mal subjekt zatvorené oči počas určitej doby s cielom rozšírenia zreníc. Vzdialenosť smartfónu bola približne 30cm. Asistent zapol blesk. Na pokyn otvoril subjekt oči a bol zachytený červený reflex s lepšou úspešnosťou v porovnaní s prvým experimentom. Rozlíšenie fotografií bolo 3000x4000. Experiment sa opakoval desaťkrát. Bolo úspešných približne 50% týchto experimentov, pričom pri najúspešnejšom experimente sa podarilo zachytiť 7 fotografií s červeným reflexom. Hardvér a softvér smartfónu má vplyv na rýchlosť dávkového snímania. Na Obr. 18 je zachytený červený reflex pomocou dávkového snímania.



Obr. 18: Experiment 2 - zachytenie červeného reflexu zadnou kamerou pomocou dávkového snímania

2.3 Experiment 3 - Video zadnou kamerou

Princípom tretieho experimentu bolo natočenie videa pomocou vstavanej aplikácie smartfónu. Bola využitá zadná kamera, pričom počas experimentu sa zúčastnil subjekt a asistent, ktorého úlohou bolo zachytiť červený reflex. Experiment bol vykonaný v tmavom prostredí. Subjekt a asistent vykonal rovnaký postup ako pri druhom experimente. Rozlíšenie videa bolo 1080x1920 pri 30 snímkach za sekundu. Z výsledného videa boli extrahované fotografie, pričom 9 z nich obsahovali červený reflex. Na obrázku 19 je extrahovaná snímka z videa s červeným reflexom.



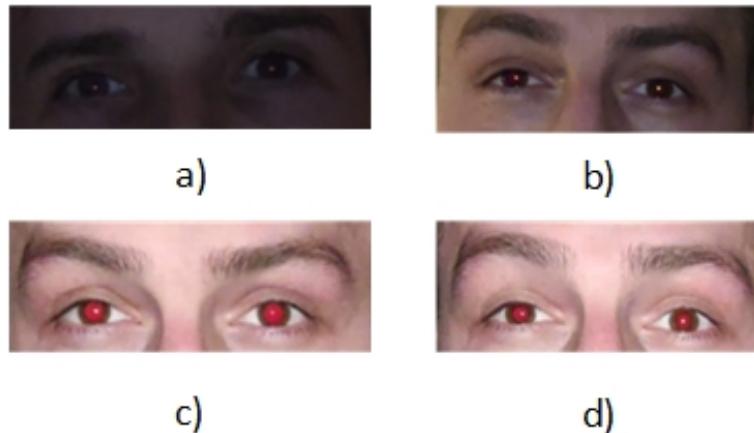
Obr. 19: Experiment 3 - extrahovaná snímka z videa s červeným reflexom

Zo všetkých 3 popísaných experimentov považujeme ako najúspešnejší, experiment so zachytením videa pomocou zadnej kamery. Pri tomto experimente bolo možné získať najväčší počet snímok s červeným reflexom s dostatočnou úspešnosťou a kvalitou. Preto sme sa rozhodli použiť v našej implementácii práve tento spôsob zachytenia červeného reflexu.

2.4 Testovanie experimentu 3

Po zistení najúspešnejšej metódy na zachytenie červeného reflexu sme vykonali sériu experimentov na zistenie najlepších podmienok pre získanie najlepších výsledkov.

Najskôr sme porovnávali vplyv svetla v miestnosti na kvalitu červeného reflexu. Experimenty sme robili so zdrojom svetla za subjektom, na boku od subjektu, v šere a v tme, pričom výsledky je možné vidieť na obrázku 20. Z tohto experimentu sme zistili, že najkvalitnejší červený reflex získame, keď snímame subjekt v šere, prípadne v úplnej tme.



Obr. 20: a) svetlo za subjektom, b) svetlo z boku subjektu, c) snímanie v šere, d) snímanie v tme

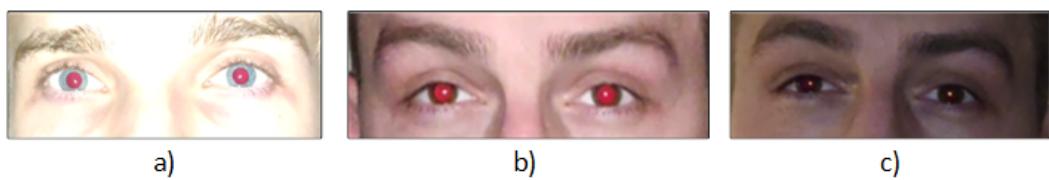
Pri ďalšom experimente sme porovnávali vertikálnu vzdialenosť kamery a blesku od snímanej osoby. Výsledky je možné vidieť na obrázku 21, pričom najlepšia metóda na

zachytenie červeného reflexu je držať smartfón zvislo a v úrovni očí snímanej osoby.



Obr. 21: a) kamera pod subjektom, b) kamera nad subjektom, c) kamera priamo na oči subjektu

Nakoniec sme vykonali experimenty na získanie optimálnej vzdialenosť kamery od snímanej osoby (Obr. 22). Zistili sme, že pri kratšej vzdialenosť (40 cm) získame lepší červený reflex ale zároveň je pri týchto snímkach väčší odraz svetla v oku. Naopak pri väčšej vzdialenosť ($>100\text{ cm}$) je červený reflex horšej kvality, a preto ako najlepšiu vzdialenosť považujeme približne 70 cm .



Obr. 22: a) vzdialenosť 40 cm , b) vzdialenosť 70 cm , c) vzdialenosť 110 cm

3 Návrh

Na základe predchádzajúcej analýzy, experimentov a zadania úlohy sme pristúpili k návrhu riešenia. Z experimentov sme zistili, že načasovanie rozsvietenia blesku vzhľadom na otvorenie očí subjektu je jedným z dôležitých aspektov úspešného zachytenia červeného reflexu očí. Pri najúspešnejšom experimente popísanom v podkapitole 2.4 bola využitá vstavaná aplikácia smartfónu. Preto sme pre našu aplikáciu zadefinovali nasledovné nefunkcionálne požiadavky:

- implementovaná ako natívna Android aplikácia,
- aplikácia nebude zobrazovať reklamy,
- podporovaný OS Android verzie 12.0 a novší,
- vyvinutá v programovacom jazyku Kotlin,
- v aplikácii budú dostupné textové inštrukcie ako postupovať pri snímaní.

Medzi definované funkcionálne požiadavky patria:

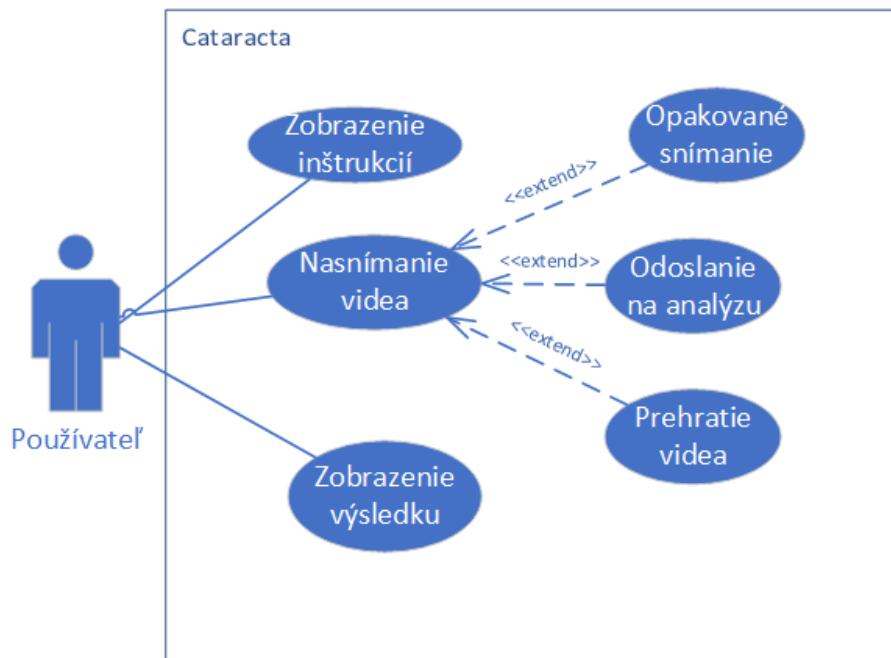
- aplikácia musí poskytovať možnosť vyhotoviť videonahrávku,
- aplikácia musí poskytovať možnosť prehrávania vyhotovenej videonahrávky,
- musí byť dostupné pozastavenie a obnovenie prehrávania pomocou vizuálnych elementov,
- používateľ musí mať možnosť opakovať nahrávanie,
- aplikácia musí poskytnúť používateľovi výsledok pre každé oko samostatne, pokiaľ prebehla úspešne detekcia očí.

3.1 Diagramy

V tejto podkapitole sú zobrazené a vysvetlené rôzne diagramy popisujúce toky udalostí vo vyvinutej aplikácii. Konkrétnie sú to UML diagram prípadov použitia, vývojový diagram popisujúci stranu klienta a diagram aktivít popisujúci stranu servera.

3.1.1 UML diagram prípadov použitia

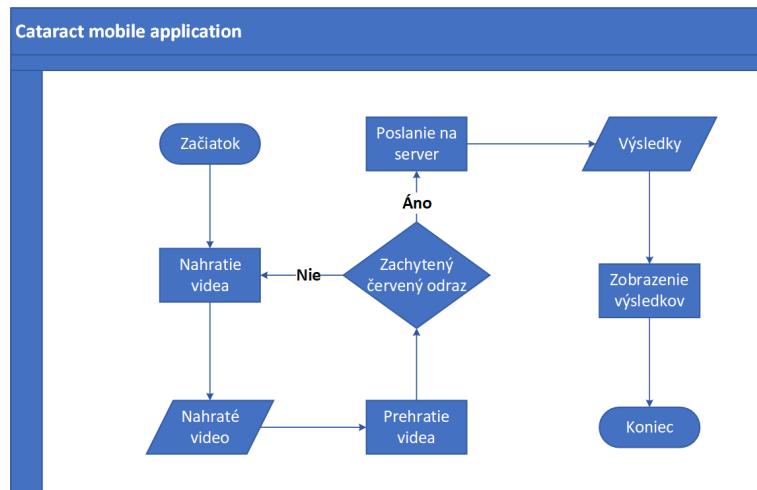
Na Obr. 23 je zobrazený UML diagram, zobrazujúci funkciaľitu z pohľadu používateľa. Používateľ má možnosť nasnímať video, prehrať a odoslať na analýzu. V aplikácii sú dostupné inštrukcie, ktorými možno docieliť zachytenie červeného reflexu. V prípade, že používateľ nie je s nahratým videom spokojný, môže nahrávanie zopakovať. Po odoslaní videa sa v prípade úspešnej detekcie zobrazia výsledky.



Obr. 23: UML diagram prípadov použitia

3.1.2 Vývojový diagram mobilnej aplikácie

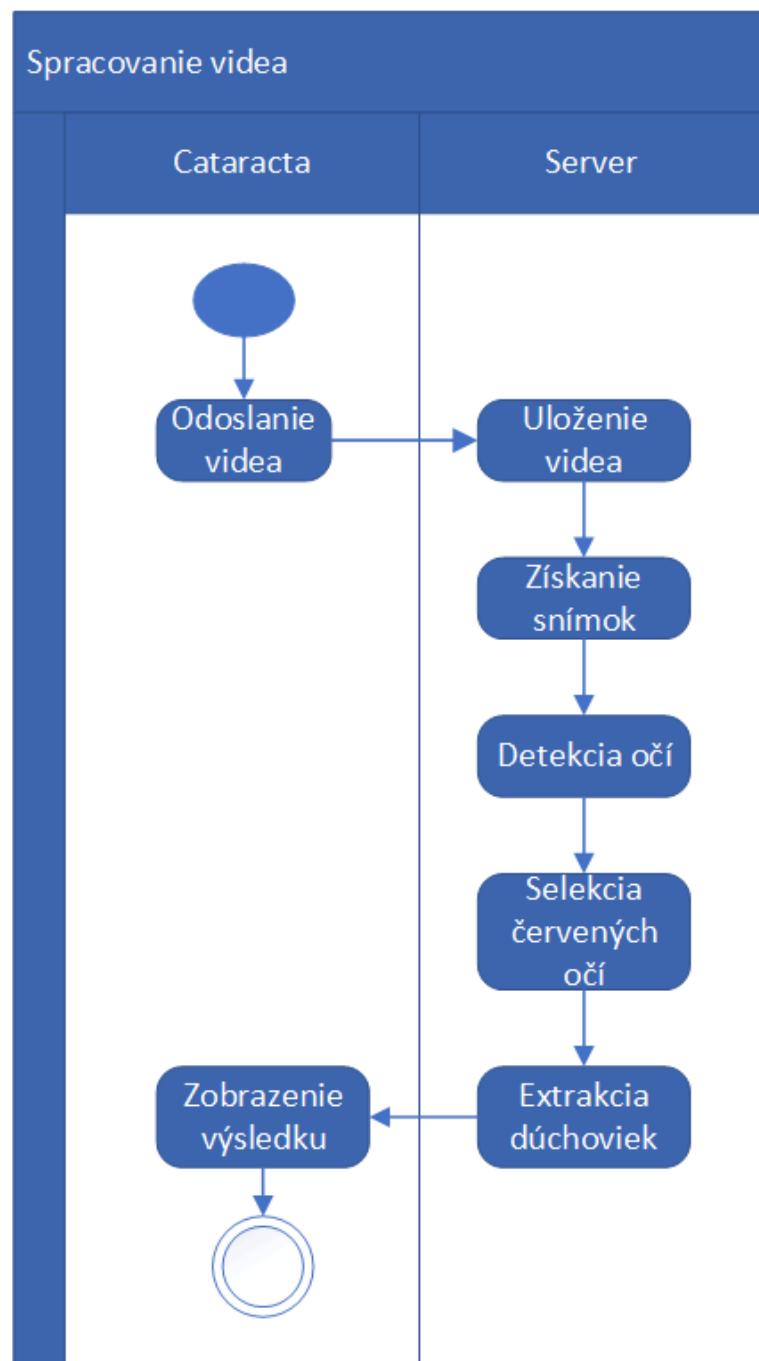
Na Obr. 24 je zobrazený vývojový diagram popisujúci postup a tok udalostí, ktoré sa vyskytujú v rámci mobilnej aplikácie. Konkrétnie popisuje rôzne kroky, rozhodnutia, vstupy, výstupy a akcie aplikácie. Prvou vykonanou akciou je nahratie videa, ktoré si následne vie používateľ prehrať a určiť, či je na ňom zachytený červený reflex očí. Ak sa tam nenachádza, tak sa opakuje akcia nahrávania videa a pokial je zachytený, video sa pošle na server. Server spracuje toto video a vráti výsledky, ktoré sa zobrazia v kroku toku udalostí v mobilnej aplikácii.



Obr. 24: Flowchart diagram mobilnej aplikácie

3.1.3 Diagram aktivít servera

Diagram aktivít na Obr. 25 bližšie popisuje postupnosť jednotlivých aktivít vedúcich k získaniu výsledku na strane servera. Po odoslaní videa používateľom na server je toto video uložené. Následne sú z neho získané jednotlivé snímky, v ktorých sa detegujú oči s červeným reflexom. Posledným krokom je extrakcia dúhoviek a odoslanie výsledku klientskej aplikácií.



Obr. 25: Diagram aktivít - spracovanie videa

4 Použité technológie

Celková architektúra tejto aplikácie pozostáva z dvoch hlavných častí - klientskej a serverovej. V klientskej časti je implementovaná mobilná aplikácia, ktorá umožňuje používateľom nahrávať videá a zobrazovať detegované červené reflexy. Serverová časť zabezpečuje spracovanie nahratých videí a detekciu červeného reflexu. Medzi klientskou a serverovou časťou je využitý REST API, ktorý umožňuje komunikáciu medzi mobilnou aplikáciou a serverom. Týmto spôsobom je dosiahnutá interakcia medzi klientom a serverom v rámci celej aplikácie.

4.1 Klient

Táto podkapitola sa zaoberá mobilnou aplikáciou pre Android vo verzii 12 (a vyššie), vyvíjanou v jazyku Kotlin v prostredí Android Studio. Aplikácia umožňuje nahrávanie videa s červeným reflexom, prehrávanie a odosielanie na server a zobrazenie detegovaného červeného reflexu.

Pre implementáciu tejto aplikácie sa využívajú knižnice CameraX, ExoPlayer a Retrofit. CameraX poskytuje jednoduché rozhranie pre fotoaparát. ExoPlayer je knižnica pre prehrávanie médií s rôznymi formátm a prispôsobiteľnosťou. Retrofit zjednoduší sietovú komunikáciu a prácu s API.

4.1.1 CameraX

CameraX je Android knižnica [21], ktorá poskytuje jednoduché a konzistentné rozhranie pre prácu s fotoaparátom na rôznych zariadeniach. Spoločnosť Google ju vyvinula s cieľom zjednodušiť prácu s fotoaparátom.

Poskytuje niekolko výhod pre vývojárov. Jednou z hlavných výhod je jej jednoduché použitie. Poskytuje API, ktoré zjednoduší prístup k základným funkciám fotoaparátu, ako je snímanie fotografií a nahrávanie videí, ovládanie zaostrenia a ďalších funkciám.

Výhodou knižnice CameraX je podpora rôznych zariadení a verzií Androidu. Knižnica poskytuje abstrakciu nad rozdielnymi implementáciami fotoaparátu v rôznych zariadeniach. Vývojári nemusia riešiť komplikácie spojené s rôznymi hardvérovými a softvérovými rozdielmi medzi jednotlivými zariadeniami. CameraX zabezpečuje, aby aplikácia fungovala konzistentne naprieč rôznymi zariadeniami.

CameraX podporuje nasledujúce prípady použitia: Náhľad (Preview), Analyzátor obrázkov (Image analysis), snímanie obrázkov (Image capture) a snímanie videa (Video capture).

Architektúra CameraX sa skladá z nasledujúcich častí:

- Náhľad (angl. Preview): Táto časť umožňuje zobrazenie náhľadu z fotoaparátu na obrazovke,
- Analyzátor obrázkov (angl. Image analysis): Táto časť poskytuje prístup a spracovanie obrazových dát pomocou strojového učenia. Poskytuje vývojárom jednoduché rozhranie pre analýzu a spracovanie obrazových dát priamo v aplikácii,
- Snímanie obrázkov (angl. Image capture): Táto časť umožňuje zachytenie a uloženie statických fotografií,
- Snímanie videa (angl. Video capture): Táto časť slúži na zachytávanie videa a zvuku pomocou zariadenia.

4.1.2 Exoplayer

ExoPlayer je Android knižnica implementujúca prehrávač médií na aplikačnej úrovni[22]. Ide o flexibilnú knižnicu, ktorá je navrhnutá s cieľom poskytovať pokročilé funkcie a podporu pre rôzne video formáty. Na rozdiel od knižnice MediaPlayer, ktorá má obmedzenú podporu video formátov. Aplikácia využívajúca ExoPlayer nepotrebuje implementovať nízko-úrovňové ovládanie prehrávania a ide o predvolenú implementáciu rozhrania pre prehrávanie videa v jetpack knižnici media3 [22]. Exoplayer vyžaduje verziu Androidu 7.1 a API level 25 [22].

4.1.3 Retrofit

Retrofit je Android knižnica, ktorá implementuje typovo bezpečný HTTP klient na spracovanie a odosielanie HTTP požiadaviek[23].

Hlavnou výhodou Retrofitu je jeho schopnosť automaticky generovať implementáciu rozhrania API na základe popisu rozhrania pomocou anotácií (@GET, @POST, a.i.) a formát dát (JSON, XML). Retrofit potom vygeneruje všetky potrebné triedy a metódy, ktoré zabezpečia odosielanie a spracovanie požiadaviek.

Retrofit podporuje automatické prevodové nástroje na spracovanie odpovedí v rôznych formátoch, ako napríklad JSON, XML, alebo aj vlastných formátov. Tieto prevodové nástroje umožňujú previesť dátá z požiadaviek (angl. request) na objekty v pamäti.

Retrofit umožňuje spracovať HTTP požiadavky asynchronne, teda mimo hlavného vlákna aplikácie, čo umožní neprerušený chod aplikácie. Zatiaľčo pri synchrónnom spracovaní, by bolo hlavné vlákno aplikácie blokované až do spracovania requestu, čo by pri dlhšom spracovaní spôsobilo tzv. mrznutie aplikácie. Retrofit vo verzii 2.9.0 vyžaduje OS Android 5 a API level 21[24].

4.2 Server

V tejto podkapitole sú popísané technológie použité na vytvorenie serverovej časti aplikácie. Backend našej aplikácie bol napísaný v jazyku Python verzie 3.10 s využitím Flask frameworku. Ďalej boli použité rôzne knižnice a neurónové siete na detekciu očí, červeného reflexu očí a dúhovky.

4.2.1 Flask

Flask je malý a nenáročný webový framework, ktorý poskytuje užitočné nástroje a funkcie zjednodušujúce vytváranie webových aplikácií v jazyku Python. Poskytujúc vývojom flexibilitu a je prístupnejším frameworkom pre začiatočníkov, pretože umožňuje rýchlo vytvoriť webovú aplikáciu pomocou jediného Python súboru. Flask je tiež rozšíriteľný a nevyžaduje špecifickú adresárovú štruktúru ani komplexné šablóny kódu [25].

Framework Flask sme použili na vytvorenie jednoduchého REST API pre komunikáciu s klientskou časťou našej aplikácie. Tento minimalistický framework nám umožnil jednoducho definovať koncové body API a logiku, ktorá sa vykonáva pri spracovaní jednotlivých požiadaviek.

4.2.2 OpenCV (Open Source Computer Vision Library)

OpenCV je rozsiahla open-source knižnica určená na počítačové videnie, strojové učenie a spracovanie obrazu. OpenCV podporuje širokú škálu programovacích jazykov, ako sú Python, C++, Java atď. Dokáže spracovať obrázky a videá na identifikáciu objektov, tvári alebo dokonca ľudského rukopisu [26].

Integráciou s knižnicami, ako napríklad Numpy, vysoko optimalizovanou knižnicou pre numerické operácie, dosiahneme najoptimálnejšie použitie funkcií poskytovaných v OpenCV.

4.2.3 Dlib

Všeobecne rozšírená open-source knižnica využívaná na úlohy strojového videnia a učenia. Bola vyvinutá Davisom Kingom v jazyku C++. Poskytuje rôzne možnosti ako napríklad detekcia tváre, tvárových orientačných bodov a sledovanie objektov. Je často využívaná v akademickom a podnikateľskom prostredí. V našej implementácii využívame detektor na detekciu tvárových bodov trénovaný na iBUG 300-W datasete v kombinácii s ResNet modelom na detekciu tváre [27][28].

4.2.4 CNN

Konvolučné neurónové siete (CNN) sú špeciálnym typom neurónových sietí, ktoré sú široko používané v oblasti spracovania obrazových a vizuálnych dát.

Využívajú konvolučné vrstvy na efektívne extrahovanie vzorov z obrazových dát. Konvolučné vrstvy obsahujú filtre, ktoré prechádzajú cez obraz a zachytávajú rôzne vizuálne znaky, ako sú hrany, textúry a obrysy objektov. Tieto filtre sa učia počas trénovania siete a automaticky sa prispôsobujú dôležitým vzorom v dátach[29].

V našom prípade sme CNN použili na trénovanie modelu na rozpoznanie červenej zrenice v obraze. Trénovanie sme vykonali pomocou *Google Teachable Machine* [30].

Google Teachable Machine

Jedná sa o webový nástroj, pomocou ktorého sa dajú jednoducho natrénovať modely neurónových sietí. Dokáže vytvárať modely, ktoré vedia klasifikovať obraz, video alebo zvuk. Jednou z jeho pozitívnych vlastností je to, že na vytvorenie modelu, používateľ nepotrebuje žiadnu znalosť neurónových sietí ani programovania[30].

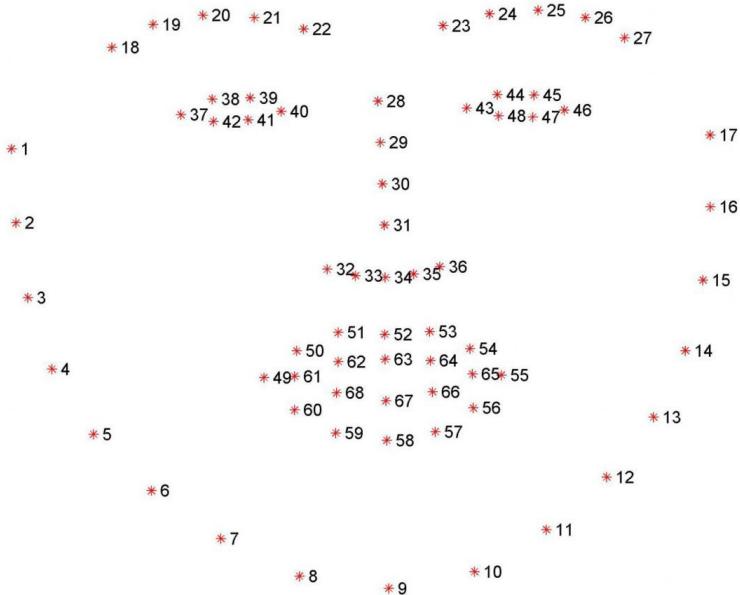
5 Implementácia

Naše riešenie je postavené na klient-server architektúre. Táto kapitola popisuje implementáciu klientskej a serverovej časti riešenia.

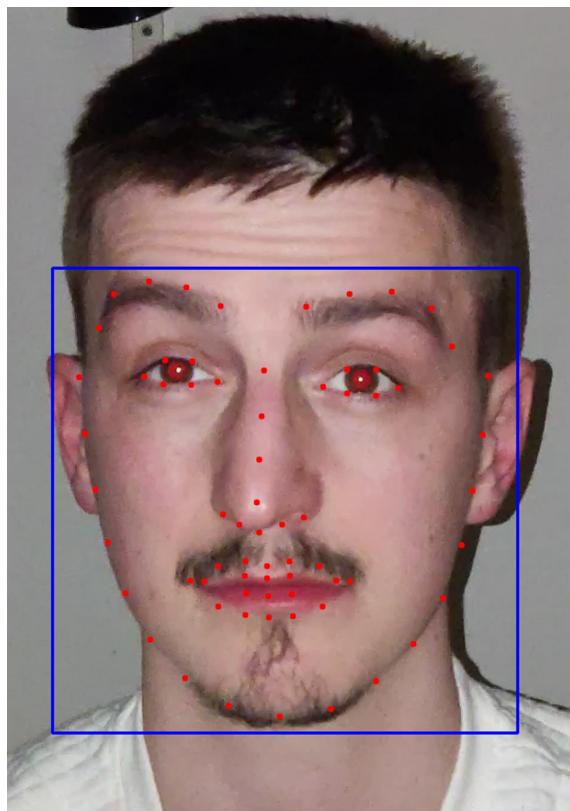
5.1 Server

Serverová strana pozostáva z jednoduchej Flask REST aplikácie obsahujúcej jeden POST endpoint. Tento endpoint je využívaný klientskou aplikáciou na zaslanie videa na server. Ďalšou súčasťou je skript, pomocou ktorého sú z videa získané najskôr jednotlivé snímky videa. V týchto snímkach je následne detegovaný box popisujúci tvár a 68 tvárových bodov pomocou knižnice Dlib. Box popisujúci tvár je reprezentovaný ako obdĺžnik definovaný dvomi bodmi, ľavým horným a pravým dolným bodom.

Tvárové body sú reprezentované ako pole 68 bodov. Ich vizuálne rozloženie možno vidieť na Obr. 26. Jeden bod predstavuje x a y koordinát na snímke. Výsledok detekcie je vykreslený do jednej zo snímkov videa, Obr. 27.



Obr. 26: Rozloženie tvárových bodov [31]



Obr. 27: Ukážka detekcie na jednej zo snímok videa

Pre vyrazenie očí zo snímok sme teda mohli použiť body popisujúce pravé oko (body 37-42) a ľavé oko (body 43-48). Z týchto bodov sú approximované body definujúce obdĺžnik, teda box, v ktorom sa nachádzajú oči.



Obr. 28: Výrez oblasti pravého oka



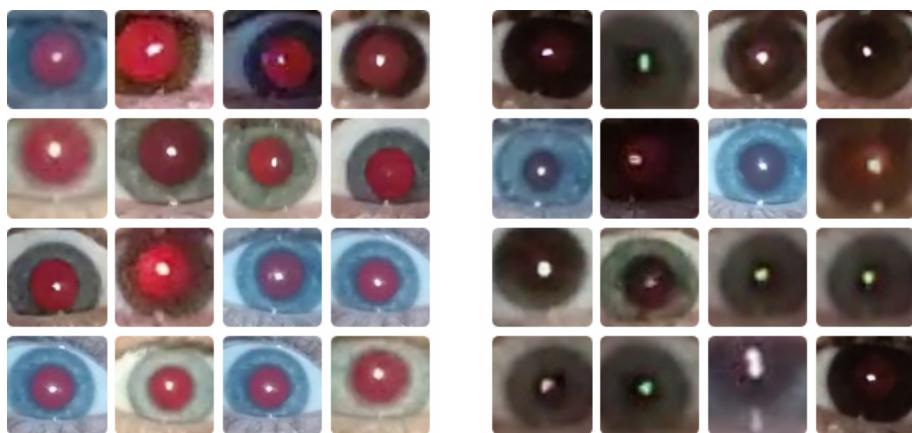
Obr. 29: Výrez oblasti ľavého oka

Oblast týchto boxov je zo snímky následne vyrezaná, Obr. 29 a Obr. 28. Detekcia je realizovaná na všetkých snímkach videa. Z tohto dôvodu je nevyhnutné zabezpečiť, aby výrezy očí obsahujúce zatvorené alebo nie úplne otvorené oči boli odfiltrované. Odfiltrovanie je realizované prostredníctvom CNN modelu popísaným v podkapitole 5.1.1 nižšie. Nakoniec sa z výrezov oblastí očí vyreže stred snímky oka (zoberie sa najkratší rozmer snímky oka

a vydáva sa štvorec v strede obrázka), teda miesto, kde sa nachádza dúhovka, a výsledok sa pošle zo servera klientovi v tvare base64.

5.1.1 Rozpoznávanie červenej zrenice

Ako už bolo spomenuté v podkapitole 4.2.4, rozpoznávanie červeného reflexu očí z obrazu sme implementovali pomocou konvolučných neurónových (CNN) sietí cez webový nástroj *Google Teachable Machine*. Pomocou nami vytvoreného Python skriptu sme dokázali vydávať všetky oči z jednotlivých snímok vstupného videa. Z týchto snímok sme vytvorili vyvážený dataset obsahujúci 100 obrazov očí s červeným reflexom a 100 obrazov očí bez červeného reflexu. Dataset bol následne použitý na trénovanie modelu.



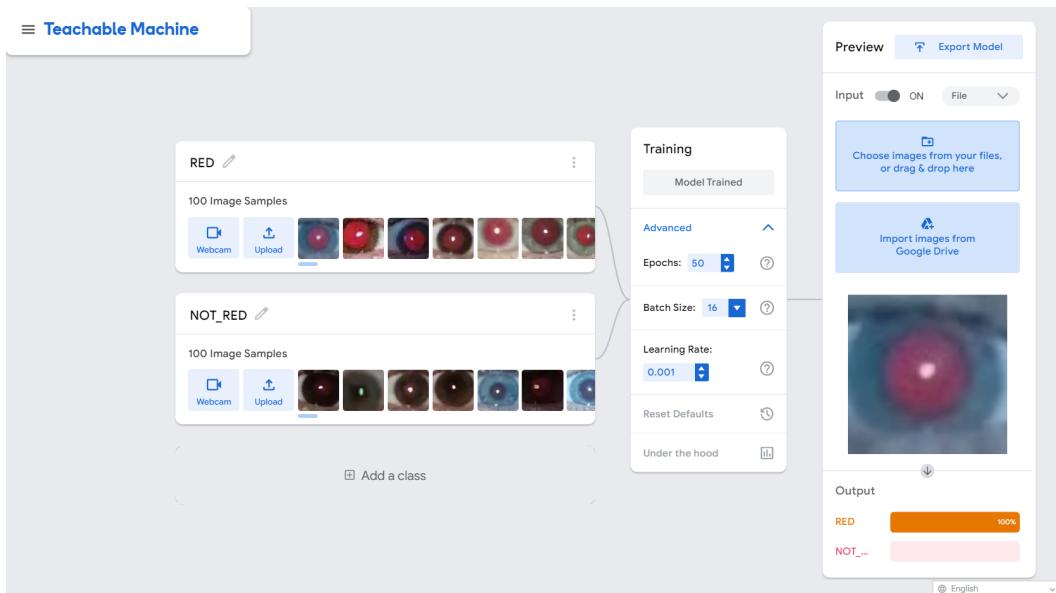
Obr. 30: Ukážka datasetu - 16 obrazov so zrenicou s červeným reflexom (vľavo), 16 obrazov so zrenicou s nečerveným reflexom (vpravo)

Jednou z výhod *Google Teachable Machine* je aj jednoduchosť jeho použitia. Vstupný dataset dokáže automaticky rozdeliť na trénovaciu a testovaciu množinu v pomere 85:15. Pri CNN je potrebné, aby všetky obrazy, ktoré do nej vstupujú, mali rovnaké rozlíšenie. V našom prípade náš dataset obsahoval obrazy s rôznymi velkosťami strán. Preto *Google Teachable Machine* automaticky oreza vstupné obrazy na štvorce tak, aby sa tieto štvorce nachádzali v strede šírky každého obrazu. Taktiež tu vieme ľahko nastavovať parametre trénovania, pričom v našom prípade sme použili konkrétné tieto:

- 50 epoch - počet prechádzaní cez kompletnú trénovaciu sadu dát počas procesu trénovania,
- 16 batch size - počet vzoriek, ktoré sú spracované súčasne pred aktualizáciou váh siete počas jednej iterácie trénovania,
- 0,001 learning rate - určuje, ako rýchlo sa aktualizujú váhy siete počas trénovania na

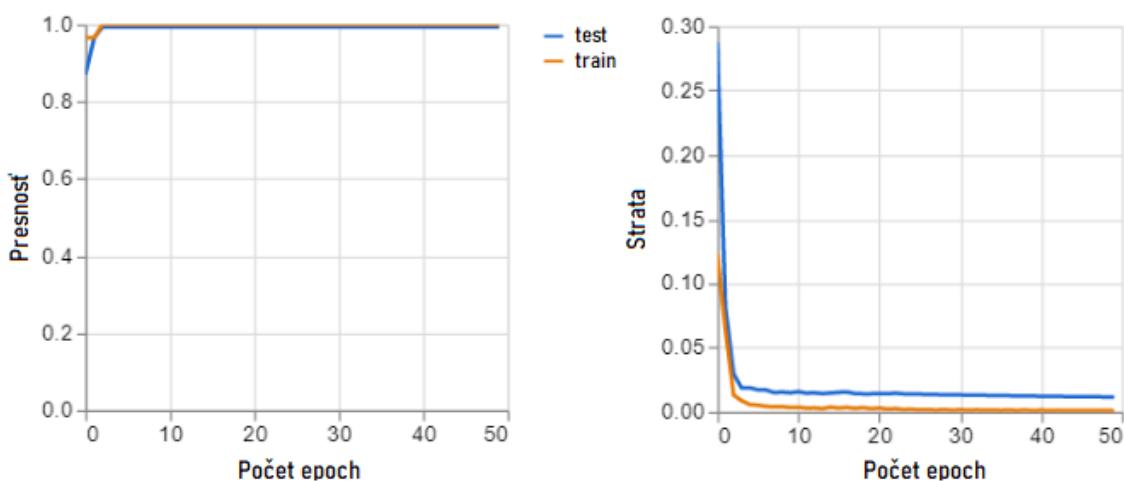
základe chyby a gradientu.

Výsledný model sme exportovali ako Keras model formátu .h5.



Obr. 31: Ukážka použitia Google Teachable Machine v našom konkrétnom prípade

V nasledujúcich 2 grafoch môžeme pozorovať, že trénovanie prebiehalo veľmi úspešne a teda nenastalo pretrénovanie ani podtrénovanie.



Obr. 32: Grafy zachytávajúce presnosť (vlavo) a stratu (vpravo) v závislosti od epochy počas trénovania

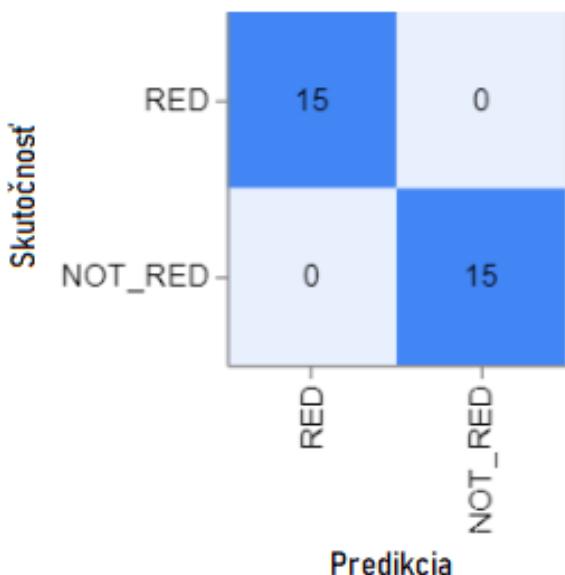
Graf vľavo zobrazuje zmeny presnosti modelu počas trénovania v jednotlivých epochách.

Presnosť (accuracy) sa meria ako pomer počtu správne klasifikovaných obrazov k celkovému počtu obrazov v trénovacej sade.

Graf vpravo zobrazuje zmenu straty modelu počas trénovania v jednotlivých epochách. Strata (loss) je hodnota, ktorá vyjadruje rozdiel medzi predikciou modelu a skutočnosťou v trénovacej sade.

Oba grafy ukazujú, ako sa presnosť modelu vyvíja počas trénovania a poskytujú informácie o tom či dochádza k pretrénovaniu alebo podtrénovaniu.

Úspešnosť nášho natrénovaného modelu dosahovala 100% klasifikáciu, čo nám potvrzuje aj konfúzna matica na obrázku 33 nižšie.



Obr. 33: Konfúzna matica klasifikačného modelu

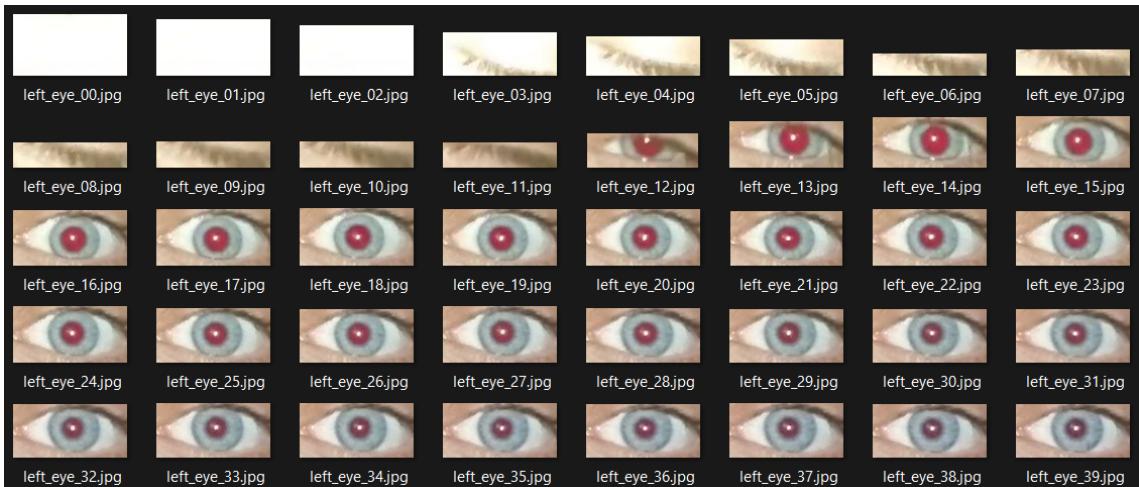
Po úspešnom natrénovaní a exportovaní modelu, sme daný model otestovali na vlastnej testovacej množine, ktorá obsahovala 15 obrazov očí s červeným reflexom a 15 obrazov očí s nečerveným reflexom. Testovanie sme vykonávali 9-krát a pri každom teste sme zvyšovali hranicu presnosti klasifikácie oka s červeným reflexom nášho modelu. To znamená, že aj keď model klasifikoval konkrétny obraz ako **RED**, ale jeho presnosť klasifikácie bola nižšia ako aktuálne určená hranica, daný obraz sme klasifikovali ako **NOT RED**. Výsledky tohto testovania sú zobrazené v tabuľke 2 nižšie.

Hranica	0,5	0,6	0,7	0,8	0,9	0,95	0,97	0,98	0,99
Presnosť	1	1	1	1	0,967	0,967	0,967	0,967	0,967
Špecificka	1	1	1	1	1	1	1	1	1
Senzitivita	1	1	1	1	0,938	0,938	0,938	0,938	0,938

Tabuľka 2: Úspešnosť nášho modelu na našej testovacej množine

Kedže náš model dosahoval veľmi vysokú úspešnosť aj pri hranici presnosti vyššej ako 99%, rozhodli sme sa práve túto hodnotu použiť ako hranicu presnosti aj vo výslednej implementácii.

Následne sme vykonali aj testovanie na ľavých zreniciach z videa doteraz nezainteresovanej osoby.



Obr. 34: Obrazy ľavej zrenice z videa nového testovacieho subjektu

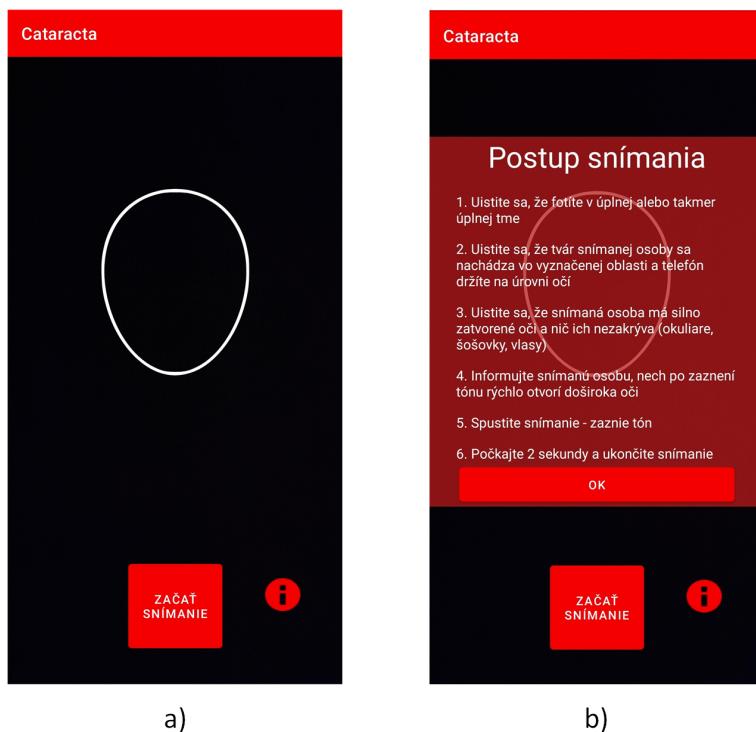
Zo vstupných obrazov z obrázku 34 vyššie, model klasifikoval ako RED tie, s názvom od **left_eye_13** po **left_eye_29** vrátane, čo považujeme za úspešné. S najvyššou presnosťou klasifikoval náš model ako RED **left_eye_14**. Je veľmi ľahké aj pre človeka určiť hranicu, ktorá z uvedených obrazov ešte obsahuje červený reflex a ktorá už nie. Pre ďalšie použitie sú vhodné čo najčervenejšie reflexy s čo najväčšou zrenicou.

5.2 Klient

Táto podkapitola detailnejšie popisuje mobilnú aplikáciu a to jej používateľské rozhranie, fungovanie a architektúru.

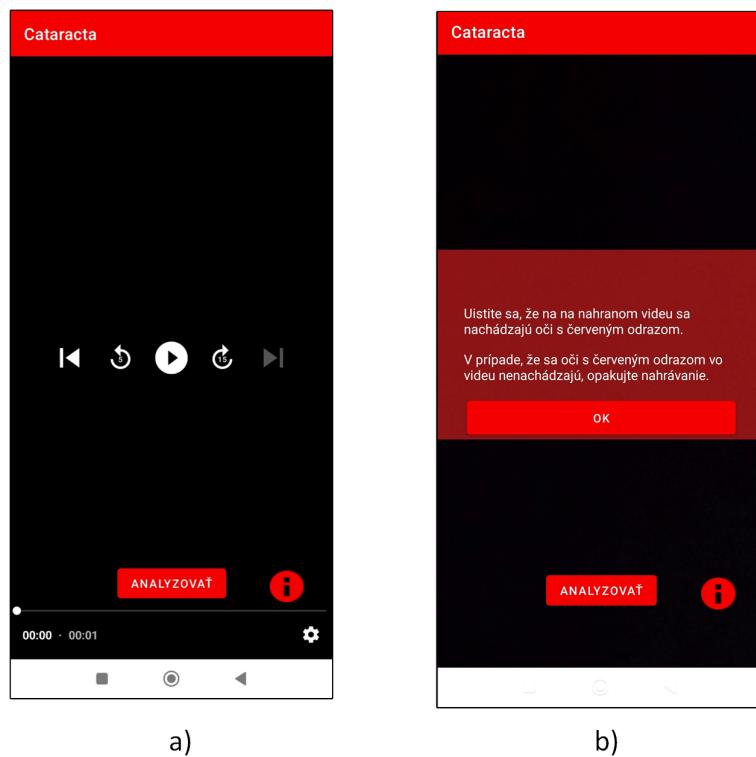
5.2.1 UI klienta

Klientská strana je tvorená aplikáciou, ktorá pozostáva zo 6 obrazoviek. Po zapnutí aplikácie sa používateľ dostane na prvú obrazovku, kde má možnosť nahrať video (Obr. 35 a)). Na tejto obrazovke sú používateľovi prístupné informácie (Obr. 35 b)), ktoré mu objasnia metódu akou má nahráť červený reflex očí snímanej osoby a obrys hlavy, ktorý mu uľahčí zarovnanie pre správnu vzdialenosť. Tieto informácie hovoria o tom, že používateľ má byt snímaný v tme, nemá mať ničím prekryté, či blokované oči a má mať zavreté oči, až pokým nebude počuť zvukový signál.



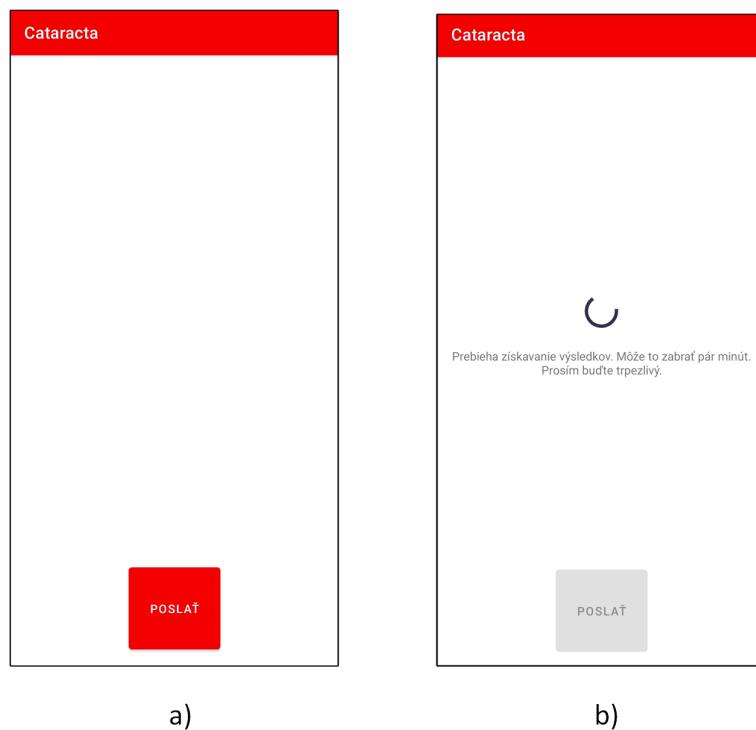
Obr. 35: a) obrazovka nahrávania videa, b) inštrukcie pre úspešné nahratie

Po skončení nahrávania sa používateľ dostane na ďalšiu obrazovku, kde má možnosť vidieť nahrané video (Obr. 36). Na tejto obrazovke je inštruuovaný aby pozrel nahrané video a zistil, či sa mu podarilo úspešne zachytiť oči s červeným reflexom snímanej osoby a v prípade, že sa mu to nepodarilo má nahrávanie opakovat. V prípade úspešného nahrávania, má používateľ možnosť poslať video na analýzu na server.



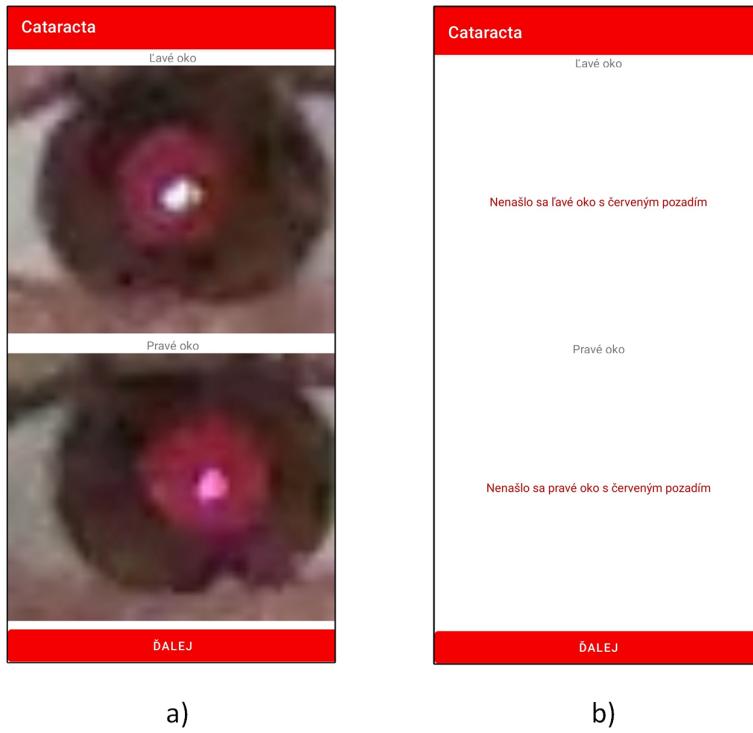
Obr. 36: a) obrazovka prehrávania videa, b) inštrukcie

Ďalej je používateľ presmerovaný na obrazovku odosielania videa (Obr. 37), kde môže poslať video na server. Po odoslaní videa sa mu zobrazí informácia, že spracovanie videa môže trvať niekoľko minút.



Obr. 37: Obrazovka odosielania videa

Po tom, čo server úspešne spracuje video, je používateľovi zobrazený výsledok (Obr. 38). Na tejto obrazovke môže používateľ vidieť vyrezaný červený reflex pre obe oči, prípadne ak server nenájde oko s červeným reflexom, používateľ o tom bude informovaný (Obr. 38 b)).



Obr. 38: Obrazovka so zobrazeným výsledkom

5.2.2 Backend klienta

Mobilnú aplikáciu sme sa rozhodli vyvinúť pre OS Android, pretože ho všetci členovia tímu mali k dispozícii, čo bolo dôležité pre plynulý vývoj aplikácie v rámci tímu.

Vývoj mobilných aplikácií pre OS Android je možný v programovacom jazyku Java a Kotlin, pričom Kotlin je moderný programovací jazyk, ktorý používa viac ako 60% profesionálnych vývojárov Android aplikácií. Kotlin pomáha zvyšovať produktivitu, spokojnosť vývojárov a bezpečnosť kódu. Aj spoločnosť Google odporúča pri vývoji natívnych aplikácií využiť programovací jazyk Kotlin pred jazykom Java, ide o tzv. Kotlin-first prístup[32]. Pri implementácii mobilnej aplikácie sme sa rozhodli vychádzať z tohto odporúčania, a preto sme zvolili jazyk Kotlin.

Pri vývoji sme zvolili vývojový vzor single-activity. Tento vzor je aj v odporúčaniach android dokumentácie [33]. Princípom tohto vzoru je členiť jednotlivé obrazovky aplikácie pomocou java triedy *Fragment* namesto triedy *Activity*. V našej aplikácii sme naimplementovali viaceré fragmenty, pričom každý slúži na zobrazenie jednej obrazovky podľa nášho návrhu. Prvý fragment slúži na nahrávanie videa, ktoré sme naimplementovali pomocou knižnice CameraX pre OS Android.

Aplikácia vyžaduje niekoľko povolení, aby mohla správne fungovať pre Android

zariadenia. Tieto povolenia sú:

- Prístup k internetu: Aplikácia potrebuje povolenie na prístup a komunikáciu cez internet. Jej funkcia môže zahŕňať stahovanie dát, odosielanie žiadostí a interakciu so vzdialenými servermi,
- Povolenie prebudenia zámku (angl. wake lock): Aplikácia vyžaduje povolenie na udržanie zariadenia v zapnutom stave,
- Povolenie zápisu do externého úložiska: Aplikácia má povolenie na zápis dát do externého úložiska zariadenia. Týmto povolením môže ukladať súbory, ako napríklad obrázky, do pamäte zariadenia,
- Prístup k fotoaparátu: Toto povolenie umožňuje aplikácii používať fotoaparát.

Okrem toho je táto aplikácia optimalizovaná pre cieľovú verziu Android API 31. To znamená, že je navrhnutá tak, aby plne využívala vlastnosti a funkcie dostupné v tejto verzii Android API.

Android [34] poskytuje tutoriál, ktorý sme použili pre náhľad nahrávaného videa a nahratie videa v mobilnej aplikácii. Pre zachytenie červeného reflexu je aplikácia optimalizovaná pre zobrazenie iba vo vertikálnej orientácii obrazovky.

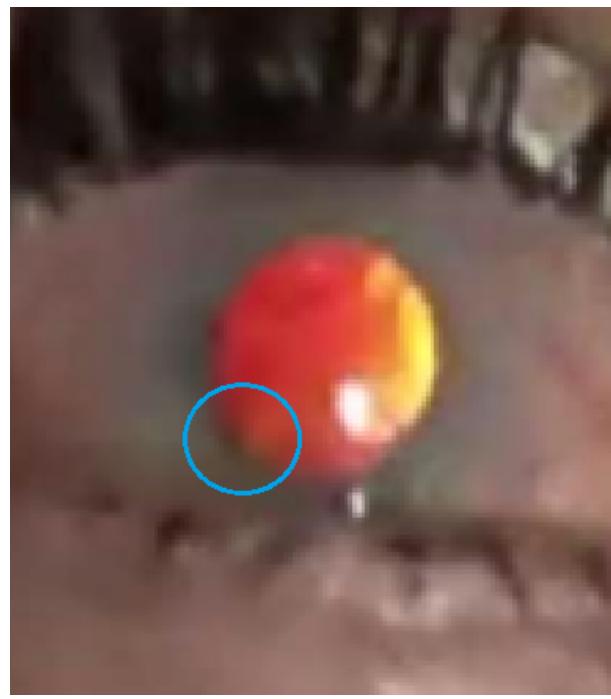
Prehrávaniu videa sme nastavili rýchlosť 0.25, aby používateľ dokázal pomocou prehrania videa skontrolovať, či sa na snímke zachytili červené oči alebo nie. Ak áno, tak používateľ má možnosť odoslať video na analýzu. Prednastavený časový limit (angl. timeout) HTTP požiadavky implementovanej pomocou knižnice Retrofit je 30 sekúnd. Podľa našich testovanií trvá spracovanie videa na serveri približne 60 sekúnd. Dĺžka spracovanie však závisí aj od dĺžky videa a teda dlhšie videá sa môžu spracovať dlhšie. Časový limit sa nastavil na 500 sekúnd, po uplynutí tohto času bez prijatia odpovede zo serveru bude požiadavka vyznačená, ako neúspešná.

6 Zhodnotenie výsledkov

Úlohou tímového projektu bolo vytvoriť mobilnú aplikáciu, ktorá by bola schopná nasnímať červený reflex očí a následne z toho zanalyzovať, či sa na danom oku nachádza alebo nenachádza sivý zákal. K tomu aby sme našli vhodnú metódu na zachytenie červeného reflexu sme vykonali radu experimentov, na základe ktorých sme implementovali našu aplikáciu.

V našej práci sa nám podarilo naimplementovať aplikáciu, ktorá je schopná úspešne a spoľahlivo zachytiť červený reflex oka snímanej osoby vo forme videa. Z tohto videa vie následne server detegovať snímku, kde je červený reflex očí najlepšie viditeľný a vymazať centrum snímky s týmto reflexom, ktorá má byť následne analyzovaná pomocou neurónovej siete, s cieľom zistiť výskyt sivého zákalu.

Aplikáciu sme tiež otestovali na pacientovi s potvrdeným sivým zákalom. Sivý zákal sa nachádza vo vyznačenej oblasti na fotografiu oka s červeným reflexom na Obr. 39. Môžeme vidieť, že kvalita tejto snímky nie je dostatočná pre natrénovanie neurónovej siete určenej na identifikáciu katarakty. Z obrázku je totiž zložité lokalizovať kataraktu aj v prípade, že vieme miesto kde sa nachádza, ktorú sme získali z predchádzajúceho vyšetrenia pomocou štrbinovej lampy. Navyše sa objavuje žltý reflex ako artefakt (pacient ho reálne nemá, je zachytený len na snímke). Z uvedeného vyplýva, že bude potrebná ďalšia úprava a skúmanie predpripravenej mobilnej aplikácie s cieľom nasnímania snímkov vyššej kvality a implementácie aplikácie do klinického prostredia za účelom vyhľadávania sivého zákalu u reálnych pacientov.



Obr. 39: Červený reflex u diagnostikovaného pacienta

Zoznam použitej literatúry

1. Ludské oko. *zeiss* [online]. 2017 [cit. 2023-01-09]. Dostupné z : <https://www.zeiss.sk/vision-care/better-vision/ako-funguje-zrak/ludske-oko.html>.
2. Zrak, oko, prierez oka. *encyklopediapoznania* [online]. 2016 [cit. 2022-12-19]. Dostupné z : <https://encyklopediapoznania.sk/clanok/6438/zrak-oko-prierez-oka>.
3. FARKAŠKOVÁ, J. Diabetická retinopatia. *diasport* [online]. [B.r.] [cit. 2022-12-19]. Dostupné z : <https://diasport.webnode.sk/diabetes-mellitus/chronicke-komplikacie/diabeticka-retinopatia/>.
4. Dry Eye Syndrome. *englishrowseyecare* [online]. [B.r.] [cit. 2022-12-19]. Dostupné z : <https://englishrowseyecare.com/dry-eye-treatment-center/>.
5. Sietnica, makula, sklovec. *neovizia* [online]. [B.r.] [cit. 2022-12-21]. Dostupné z : <https://www.neovizia.sk/liecba-sietnice-a-sklovca/sietnica-makula-sklovec/>.
6. ŠEDÝ ZÁKAL - KATARAKTA. *Iclinic* [online]. [B.r.] [cit. 2022-12-15]. Dostupné z : <https://www.iclinic.sk/sedyzakal-katarakta>.
7. Šedý zákal (katarakta). *nzip* [online]. [B.r.] [cit. 2022-12-21]. Dostupné z : <https://www.nzip.cz/clanek/394-sedy-zakal-katarakta>.
8. Snellenove tabuľky. *mixxer* [online]. [B.r.] [cit. 2022-12-29]. Dostupné z : https://www.mixxer.sk/fotky4756/fotos/_vyr_162031308.jpg.
9. Cataract Grading. *eyeguru* [online]. [B.r.] [cit. 2022-12-29]. Dostupné z : <https://eyeguru.org/learn/practice-cataract-grading/>.
10. PATEL, C., HAMADA, S., ATHANASIADIS, Y., BATES, R. a SALEH, G. M. Intraocular lens opacification mimicking the appearance of a congenital lamellar cataract. *nature* [online]. 2012 [cit. 2023-01-23]. Dostupné z DOI: <https://doi.org/10.1038/eye.2012.182>.
11. CHIU, S. Retroillumination. *optometrystudents* [online]. 2018 [cit. 2023-05-17]. Dostupné z : <https://www.optometrystudents.com/going-retro-a-guide-to-retroillumination/>.
12. IK, Zu Quan, LAU, Sian Lun a CHAN, Jan Bond. Mobile cataract screening app using a smartphone. *2015 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*. 2015. Dostupné z DOI: [10.1109/ic3e.2015.7403496](https://doi.org/10.1109/ic3e.2015.7403496).

13. TRIPATHI, Pavani, AKHTER, Yasmeena, KHURSHID, Mahapara, LAKRA, Aditya, KESHARI, Rohit, VATSA, Mayank a SINGH, Richa. MTCD: Cataract Detection via Near Infrared Eye Images. *arXiv: Computer Vision and Pattern Recognition*. 2021. Dostupné tiež z: <http://arxiv.org/pdf/2110.02564.pdf>.
14. NASIRZONOUZI, Mona. *Automated Cataract Grading using Smartphone Images*. University of Waterloo, [b.r.].
15. cataract dataset. 2019. Dostupné tiež z: <https://www.kaggle.com/datasets/jr2nbg/cataractdataset>.
16. CATARACTS. 2021. Dostupné tiež z: <https://ieee-dataport.org/open-access/cataracts>.
17. ODIR-2019 - Grand Challenge. [B.r.]. Dostupné tiež z: <https://odir2019.grand-challenge.org/dataset/>.
18. Retinal Fundus Multi-disease Image Dataset (RFMiD). 2021. Dostupné tiež z: <https://ieee-dataport.org/open-access/retinal-fundus-multi-disease-image-dataset-rfmid>.
19. YOO, Seunghwan a PARK, Rae-Hong. Red-eye detection and correction using inpainting in digital photographs. *IEEE Transactions on Consumer Electronics*. 2009, roč. 55, č. 3, s. 1006–1014. Dostupné z DOI: 10.1109/TCE.2009.5277948.
20. PAN. *Intense red-eye effect in blue eyes with dilated pupil*. 2007. Dostupné tiež z: <https://commons.wikimedia.org/wiki/File:BoldRedEye.JPG>.
21. DEVELOPERS, Android. *Android Developers* [online]. 2023. [cit. 2023-05-08]. Dostupné z : <https://developer.android.com/>.
22. DEVELOPERS, Android. google, 2023. Dostupné tiež z: <https://developer.android.com/guide/topics/media/exoplayer/supported-devices>.
23. WHARTON, Jake. *Square/Retrofit: A Type-safe HTTP client for Android and the JVM*. 2020. Dostupné tiež z: <https://github.com/square/retrofit>.
24. WHARTON, Jake. *Square/Retrofit: A Type-safe HTTP client for Android and the JVM*. 2019. Dostupné tiež z: <https://github.com/square/retrofit>.
25. *What is Flask Python - Python Tutorial*. online, 2021. Dostupné tiež z: <https://pythonbasics.org/what-is-flask-python/>.
26. *OpenCV: Introduction to OpenCV-Python*. 2023. Dostupné tiež z: https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html.

27. KING, Davis. *Dlib* [<https://dlib.net/>]. accessed 2023. Accessed on 28th April 2023.
28. KING, Davis. *Dlib GitHub Repository* [<https://github.com/davisking/dlib>]. accessed 2023. Accessed on 28th April 2023.
29. GOODFELLOW, Ian, BENGIO, Yoshua a COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
30. SHARMA, S. Build a Machine Learning model in Minutes. *medium* [online]. 2020 [cit. 2023-05-02]. Dostupné z : <https://medium.com/tek-society/what-is-teachable-machine-a4550bfd3af1>.
31. IBUG. *The 68 points mark-up used for our annotations*. accessed 2023. Dostupné tiež z: <https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/>. Accessed on 28th April 2023.
32. DEVELOPERS, Android [online]. google, 2023 [cit. 2023-04-13]. Dostupné z : <https://developer.android.com/kotlin/first>.
33. DEVELOPERS, Android [online]. 2023. [cit. 2023-03-31]. Dostupné z : <https://developer.android.com/topic/architecture/recommendations>.
34. DEVELOPERS, Android. *Android Developers* [online]. 2023. [cit. 2023-05-08]. Dostupné z : <https://developer.android.com/codelabs/camerax-getting-started>.

Prílohy

A	Návod na spustenie aplikácie	II
B	Zápisnice zo zimného semestra	III
C	Zápisnice z letného semestra	XIV

A Návod na spustenie aplikácie

V rámci príloh práce sa nachádzajú súbory:

- **Cataracta-main.zip**: komprimované zdrojové súbory Android projektu,
- **cataractApp.zip** komprimovaný inštalačný .apk súbor aplikácie je spustiteľný na minimálnej verzii OS Android 12 (API level 31), pri inštalácii bude možno nutné povoliť v nastaveniach inštaláciu z neznámych zdrojov.

K tomu, aby Vám aplikácia správne fungovala, musíte byť pripojený na školskej sieti, prípadne využiť pripojenie cez VPN (vid ľ **návod**).

B Zápisnice zo zimného semestra

Na nasledujúcich stránkach sú uvedené zápisnice zo všetkých stretnutí zimného semestra 2022/2023.

Zápisnica 1

Dátum porady:

6.10.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Počas porady sme sa zoznámili a predstavili. Boli nám predstavené ciele projektu, zámer projektu a spôsob, ktorým by mohla byť detekcia katarakty realizovaná, teda pomocou červeného odrazu. Pani MUDr. Veronika Kurilová, PhD. navrhla Trello ako platformu na stanovenie úloh a zdieľanie prezentácií a súborov.

Stanovené úlohy:

- vytvorenie skupiny na platforme WhatsApp a Trello
- zoznámenie sa s online databázami knižníc
- zvolenie vedúceho tímu

Zápisnica 2

Dátum porady:

14.10.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Bola nám predstavená, pani MUDr. Veronikou Kurilovou, PhD. problematika katarakty, jej vyšetrenie pomocou štrbinovej lampy a metódy diagnostiky. Nošík bol zvolený ako vedúci tímu. Nošík vytvoril skupinu na komunikáciu v tíme. Každý z členov tímu sa zoznámil s dostupnými databázami knižníc. Danko poznamenal, že červený odraz je rozličný pre ľudí rôznej rasy a geografickej polohy (napr. Afričania, Aziati a Europánia). Pani MUDr. Veronika Kurilová, PhD. stanovila, že náš zámer je sa orientovať na ľudí s červeným odrazom.

Stanovené úlohy:

- analýza prístupov a nastavení pri implementácii aplikácie

Zápisnica 3

Dátum porady:

20.10.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Bola nám predstavená, pani MUDr. Veronikou Kurilovou, PhD. anatómia oka.

Stanovené úlohy:

- analýza nastavenia mobilného fotoaparátu
- prieskum existujúcich mobilných aplikácií
- analýza databáz katarakty
- preskúmanie existujúcich prác pre detekciu katarakty

Zápisnica 4

Dátum porady:

27.10.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Mičuda prezentoval existujúce práce, databázy a existujúce mobilné aplikácie pre detekciu katarakty. Tiež našiel zdroj, ktorý uvádza, akým spôsobom je najpravdepodobnejšie zachytiť červený odraz. Danko prezentoval nastavenia fotoaparátu pre odfotenie červeného odrazu. Ďalej spomenul, že určité druhy látok (alkohol a nikotín) môžu spôsobiť rozšírenie zrenice. Experimentom zistil, že fotografie formátu DNG sú o 15% až 20% menšie ako fotografie formátu RAW, pričom rozdiel v kvalite je minimálny. Zistil, že pre odfotenie červeného odrazu je optimálne najkratšie trvanie blesku.

Stanovené úlohy:

- prieskum ďalších databáz katarakty
- prieskum knižníc pre React Native a CameraX
- dôkaz konceptu (ang. Proof of concept)

Zápisnica 5

Dátum porady:

11.11.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Ištók a Vrbovský prezentovali dostupné React Native knižnice, vhodné pre zachytenie červeného odrazu. Nošík ukázal aplikáciu a prezentoval spôsoby, ktorým by bola diagnostika katarakty realizovateľná. Kopecký ukázal fotografie svojho červeného oka, pričom jeho zachytenie nebolo vo väčšej miere úspešné (menej ako 25%). Bol navrhnutý mód dávkového snímania, ako alternatíva k zachyteniu červeného odrazu.

Stanovené úlohy:

- analýza červenej zložky červeného odrazu
- experiment, zachytenie a natočenie videa červeného odrazu

Zápisnica 6

Dátum porady:

18.11.2022

Zúčastnení:

Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Išták

Nezúčastnený:

Daniel Nošík (návšteva lekára)

Popis priebehu porady:

Mičuda, Vrbovský a Išták prezentovali video, ktoré zachytil červený odraz a fotografie vyhotovené pomocou dávkového snímania. Kopecký ukázal červenú zložku fotografie červeného odrazu svojho oka.

Stanovené úlohy:

- prieskum riešení na orezávanie očí
- prieskum riešení na detekciu červeného odrazu na fotografií

Zápisnica 7

Dátum porady:

24.11.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Mičuda ukázal spôsoby, ktorým orezanie a extrakcie červeného odrazu je realizovateľná. Prezentoval extrahované fotografie z videa, ktoré zachytilo červený odraz. Kopecký prezentoval algoritmy, ktoré robia detekciu dúhovky a zrenice.

Stanovené úlohy:

- analýza RGB zložiek červeného odrazu

Zápisnica 8

Dátum porady:

1.12.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Danko prezentoval prezentáciu o RGB zložkách červeného odrazu. Mičuda prezentoval spôsob, ktorým je možné roztriediť pravé a ľavé oko.

Stanovené úlohy:

- UML diagram použitia
- zoznámenia sa s vývojovým prostredím AndroidStudio

Zápisnica 9

Dátum porady:

8.12.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Všetci členovia tímu sa zoznámili s vývojovým prostredím AndroidStudio, ktoré slúži na vývoj mobilných aplikácií. Nošík vypracoval UML diagram použitia.

Stanovené úlohy:

- vypracovanie písomnej časti za zimný semester
- analýza doterajších prác v oblasti

Zápisnica 10

Dátum porady:

15.12.2022

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Členovia tímu sa jednoznačne zhodli na vypracovaní písomnej časti v Latex. Nošík mal otázky, ktoré súviseli zo štruktúrou dokumentu a vo výsledku sme sa zhodli, že je potrebné zmeniť štruktúru dokumentu. Danko a Mičuda prezentovali ďalšie práce v oblasti.

Stanovené úlohy:

- dokončenie písomnej časti za zimný semester

C Zápisnice z letného semestra

Na nasledujúcich stránkach sú uvedené zápisnice zo všetkých stretnutí letného semestra 2022/2023.

Zápisnica 11

Dátum porady:

17.2.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Všetci členovia tímu sa zhodli, na štúdiu programovacieho jazyka Kotlin pre vývoj mobilnej aplikácie pre operačný systém Android.

Stanovené úlohy:

- štúdium vývoja mobilnej aplikácie v prostredí Android Studio
- implementácia zachytenia červeného odrazu v mobilnej aplikácii

Zápisnica 12

Dátum porady:

24.2.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadané úlohy neboli splnené, pretože nastavenie trvania blesku je nerealizovateľné v spojení s knižnicou CameraX.

Stanovené úlohy:

- navrhnutie UI aplikácie
- implementácia zachytenia červeného odrazu v mobilnej aplikácii

Zápisnica 13

Dátum porady:

3.3.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadaná úloha, navrhnutie UI aplikácie, bola splnená. Nepodarilo sa nám splniť úlohu implementácie červeného odrazu, preto Nošík rozhadol, že implementáciu záchytenia červeného odrazu budú členovia tímu realizovať samostatne.

Stanovené úlohy:

- implementácia záchytenia červeného odrazu v mobilnej aplikácii

Zápisnica 14

Dátum porady:

10.3.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Tomáš Danko a Patrik Ištók

Nezúčastnený:

Adrián Vrbovský (návšteva lekára)

Popis priebehu porady:

Zadané úlohy boli splnené. Mobilná aplikácia úspešne zachytila červený odraz. Pri vykonaní experimentov na porade sme zistili, že zachytenie nie je vždy úspešné. Vedúca tímového projektu navrhla vykonávanie experimentov za cieľom zistenia optimálnych podmienok.

Stanovené úlohy:

- vykonanie experimentov pre zistenie optimálnych svetelných podmienok

Zápisnica 15

Dátum porady:

17.3.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadané úlohy boli splnené. Získané výsledky z experimentov neboli dostatočné. Vedúca tímového projektu navrhla vykonanie ďalších experimentov pri rôznych vzdialenosťach subjektu.

Stanovené úlohy:

- vykonanie experimentov pre rôzne vzdialosti
- pridanie oválu pre tvár v mobilnej aplikácii

Zápisnica 16

Dátum porady:

24.3.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadané úlohy boli splnené. Z vykonaných experimentov sme zistili, že optimálna vzdialenosť pre zachytenie červeného odrazu je 70 cm

Stanovené úlohy:

- pridanie inštrukcií pre používateľa do mobilnej aplikácie
- pridanie luxmetra do aplikácie
- prispôsobenie oválu na vzdialenosť 70 cm
- pridanie Fragmentu na odosielanie videa na server v mobilnej aplikácii

Zápisnica 17

Dátum porady:

31.3.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadané úlohy boli splnené. Kopecký pridal inštrukcie, luxmeter a prispôsobil ovál v mobilnej aplikácii.

Stanovené úlohy:

- vykonanie experimentov pre luxmeter
- úprava inštrukcií v mobilnej aplikácii prispôsobenie oválu na vzdialenosť 70 cm

Zápisnica 18

Dátum porady:

14.4.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Členovia tímu zistili, že luxmeter je zbytočný v mobilnej aplikácii, pretože získané údaje o svetelných podmienkach sú zo senzora z prednej strany smartfónu. Rozhodli sme sa, pristúpiť k architektúre klient, server, nakoľko skripty implementované Mičudom sú v jazyku Python a sú výpočtovo náročné. Vedúca tímového projektu sa pokúsi zabezpečiť školský server.

Stanovené úlohy:

- implementácia serverovej časti
- odstránenie luxmetra

Zápisnica 19

Dátum porady:

21.4.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadané úlohy boli splnené. Extrahovanie snímok s červeným odrazom očí pomocou Viola Jones detektora nebolo dostatočné, preto sme sa rozhodli natrénovať neurónovú sieť alebo SVM klasifikátor na identifikáciu snímok s červeným odrazom očí.

Stanovené úlohy:

- natrénovanie neurónovej siete
- implementácia SVM klasifikátora
- otestovanie endpointu lokálneho servera na odosielanie videa

Zápisnica 20

Dátum porady:

28.4.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadané úlohy boli splnené. Kopecký prezentoval porovnanie výsledkov natrénovanej neurónovej sieti a SVM klasifikátora na detekciu červeného odrazu oka zo snímky. Z výsledkov sme sa zhodli, že použitie neurónovej sieti je optimálnejšie. Vedúca tímového projektu poskytla prihlásovacie údaje na pripravený školský server.

Stanovené úlohy:

- implementácia odoslania videa v mobilnej aplikácii
- optimalizácia modelu neurónovej sieti (úprava datasetu)
- dokončenie implementácie backendu

Zápisnica 21

Dátum porady:

5.5.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadané úlohy boli splnené. Kopecký ukázal CNN model, upravený dataset a úspešnosť klasifikácie oka s červeným odrazom. Danko ukázal prehrávanie videa v aplikácii. Nošík a Ištók informovali o dokončení backendu.

Stanovené úlohy:

- implementácia prehrávania videa v mobilnej aplikácii
- implementácia navigácie v knižnici Android Jetpack
- nasadenie backendu na školský server

Zápisnica 22

Dátum porady:

18.5.2023

Zúčastnení:

Daniel Nošík, Miroslav Kopecký, Roman Mičuda, Adrián Vrbovský, Tomáš Danko a Patrik Ištók

Popis priebehu porady:

Zadané úlohy boli splnené. Študenti odprezentovali dokončenú aplikáciu a jej komunikáciu so serverom.

Stanovené úlohy:

- dokončenie písomnej časti práce