

Univerzita Komenského v Bratislave Fakulta
matematiky, fyziky a informatiky

**System pre programovanie robotického drona
s počítačovým videním**

Bakalárska práca

2023

Peter Verčimák

Univerzita Komenského v Bratislave Fakulta
matematiky, fyziky a informatiky

**System pre programovanie robotického drona
s počítačovým videním**

Bakalárska práca

Študijný program: Aplikovaná informatika

Študijný odbor: Informatika

Školiace pracovisko: Katedra aplikovanej informatiky

Školiteľ: Mgr. Pavel Petrovič, PhD.

Bratislava, 2023

Peter Verčimák



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

- Meno a priezvisko študenta:** Peter Verčimák
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:** Systém pre programovanie robotického drona s počítačovým videním
System for programming robot drone with computer vision
- Anotácia:** Dron Tello je malý dron do interiéru, ktorý je možné riadiť na diaľku z programu cez wifi. Okrem toho sa cez wifi dá získať obraz z jeho kamery. Na slovenskom kole súťaže Robocup Junior sa už dva roky konala kategória dron, kde sa s týmto typom drona zapojili viacerí súťažiaci zo základných a stredných škôl. Doteraz však programy fungovali "na slepo" - bez toho, aby využívali a spracovávali obraz z kamery.
- Cieľ:** Cieľom bakalárskej práce je pripraviť jednoducho použiteľnú knižnicu funkcií využívajúcich počítačové videnie, tutoriál s jednoduchými príkladmi jej využitia a navrhnúť nové výzvy na rozšírenie tejto kategórie tak, aby bolo nutné využiť spracovanie obrazu. Študent tiež predvedie využitie knižnice na iných netriviálnych dobre zdokumentovaných príkladoch, čím demonštruje spôsob využitia knižnice. Očakáva sa, že sa v tejto sezóne zúčastní na súťaži a bude sa podieľať nielen na príprave výzvy ale aj jej hodnotenia, čo v texte práce zdokumentuje.
- Literatúra:** Ján Špirka: Tello dron s OCR ovládaním, dostupné on-line: https://wiki.robotika.sk/robowiki/index.php?title=Tello_dron_s_OCR_ovladanim_-_Jan_Spirka
- Kľúčové slová:** spracovanie obrazu, dron, robotická súťaž
- Vedúci:** Mgr. Pavel Petrovič, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.
- Dátum zadania:** 01.09.2022
- Dátum schválenia:** 01.06.2023 doc. RNDr. Damas Gruska, PhD.
garant študijného programu

študent

vedúci práce

Čestné vyhlásenie:

Čestne vyhlasujem, že som bakalársku prácu vypracoval samostatne s použitím uvedenej literatúry a zdrojov dostupných na internete.

V Bratislave dňa 02.06.2023

Peter Verčimák

Pod'akovanie : Týmto by som sa chcel poďakovať vedúcemu bakalárskej práce Mgr. Pavlovi Petrovičovi, PhD. za rady, pomoc, trpezlivosť a čas strávený na našich stretnutiach. Taktiež by som chcel poďakovať RNDr. Petrovi Borovanskému PhD., za jeho rady pri otázkach týkajúcich sa vývoja Android aplikácii.

Abstrakt

Táto práca sa zaoberá návrhom a rozšírením kategórie dron súťaže Robocup Junior Slovensko, na ktorej doposiaľ nebol využívaný plný potenciál dronov s kamerou. V práci sme vytvorili knižnicu funkcií, ktorej zámer je komunikácia s dronom s využitím počítačového videnia. Súťaže sme sa aktívne zúčastnili, kde sme predviedli využitie knižnice vyriešením novo navrhutej disciplíny Dron. Okrem toho, pre hodnotenie výsledkov súťažiacich a prácu s nimi sme vyvinuli aplikáciu pre zariadenia s operačným systémom Android.

Kľúčové slová: spracovanie obrazu, dron, robotická súťaž

Abstract

This thesis investigates the possibilities of expanding an ongoing category Drone of a competition called Robocup Junior Slovensko, where drones equipped with cameras have not yet been used to the fullest extent of their potential. We created a programming library, which is intended for communicating with the drone using computer vision. We actively took part presenting the use of spoken library by solving the new category Drone. Furthermore, for the purpose of rating the competing teams and data processing we developed an application for devices using the Android operating system.

Keywords: image processing, drone, robot competition

Obsah

Úvod.....	1
1. Východiská.....	2
1.1 Robocup Junior	2
1.2 Robocup Junior Slovensko.....	4
1.2.1 Kategória Dron dodnes	4
1.2.3 Existujúce riešenia sledovania čiary dronom	6
1.3 Drony vo svete	7
1.3.1 Využitie dronov pri záchranných operáciách.....	7
1.4 Robotický dron Tello	9
1.4.1 Procesor.....	9
1.4.2 Pripojenie na drona	9
1.4.3 Kamera	10
1.4.4 Vision Positioning System	10
1.4.5 Collision detection system	11
1.4.6 Programovateľnosť	11
1.5 Ovládanie drona	11
1.6 Počítačové videnie	11
1.7 HSV model.....	12
1.8 Predspracovanie obrazu	13
1.8.1 Segmentácia	13
1.8.2 Filtrovanie obrazu	13
1.8.3 Morfológické operácie – Erózia a Dilatácia.....	15
1.8.4 Detekcia hrán	16
1.8.5 Minimálny ohraničujúci obdĺžnik	18
1.9 Modul PyZbar	18
1.10 Haar Cascade.....	18
1.11 Vývoj Android aplikácií.....	18
1.11.1 Jazyk a prostredie	18
1.11.2 Perzistencia dát.....	19
2. Návrh.....	20
2.1 Nová kategória Dron	20
2.2 Knižnica pre programovanie drona	23
2.2.2 Návrh tried a skriptov.....	23
2.3 Android aplikácia pre rozhodcov súťaže Robocup Junior Slovensko.....	25

2.3.1 Šablóna hodnotenia	26
2.3.2 Požiadavky funkcionalít aplikácie	26
2.3.3 Zjednodušený Entitno-Relačný model databázy aplikácie	27
3. Implementácia	28
3.1 Aplikovanie knižnice djitellopy	28
3.1.1 Pripojenie sa na drona	28
3.1.2 Získavanie obrazu z drona	28
3.2 Problematika nasmerovania kamery	29
3.3 Problematika Vision Positioning System-u drona Tello	29
3.4 Rozoznávanie farebných geometrických objektov	30
3.5 Pohyb drona po čiare.....	31
3.6 Súbor funkcií.....	31
3.7 Aplikácia pre rozhodcov – implementácia.....	32
3.7.1 Perzistencia dát aplikácie	32
3.7.2 Spracovanie CSV súboru s informáciami o disciplínach a tímo	32
3.7.3 Spracovanie konfiguračného súboru so šablónami hodnotení	34
3.7.5 Use case : Vytvorenie nového hodnotenia	35
3.7.6 Export hodnotení disciplíny do súboru CSV.....	36
3.7.7 Používateľské rozhranie aplikácia.....	36
3.7.8 Komplikácie s aplikáciou.....	37
4. Priebeh kategórie Dron ročníka 2023 Robocup Junior Slovensko.....	38
5. Záver	39
6. Literatúra.....	40
7. Prílohy.....	42

Úvod

Technológie napredujú exponenciálnou rýchlosťou. Príkladom je fakt, že pred pár desaťročiami sa ľudia posmievali myšlienke, že by sme v blízkom čase mali doma robotického pomocníka. Ten by vykonával rôzne jednoduché úkony v domácnosti, pomáhal pri upratovaní či kontroloval zásoby. Dnes však nie je vôbec zvláštne, že takého automatického pomocníka nájdeme v značnom zastúpení domácností, aby pomohol a ušetril čas.

Významnou oblasťou dnešných technológií je počítačové videnie. Vďaka pokročilým algoritmom a technikám dokáže počítačové videnie identifikovať a rozpoznávať rôzne objekty, tváre a pohyby na obrázkoch a videách. Jednoducho povedané, počítač má vidieť a rozumieť tomu, čo vidí, vzdialene pripomínajúc ľudský zrak. Využitie počítačového videnia je široké a ponúka veľa možností v rôznych odvetviach, napríklad v medicíne môže pomôcť pri detekcii chorôb na medicínskych snímkach. Taktiež v automobilovom priemysle môže prispieť k rozvoju samo-riaditeľných vozidiel schopných rozpoznať prekážky a reagovať na ne.

Táto bakalárska práca predstavuje návrh a rozšírenie kategórie dron súťaže Robocup Junior Slovensko. Výstupom bude knižnica, pomocou ktorej budú súťažiaci tohto, ale aj nasledujúcich ročníkov riešiť novú kategóriu Dron a mobilná Android aplikácia, ktorá rozhodcom uľahčí hodnotenie. Samotná práca sa člení do viacerých častí, ktoré spolu súvisia.

V prvej kapitole predstavujeme teoretické východiská práce, ktorých súčasťou je popis existujúcich súťaží robotov, starej kategórie Dron súťaže RJS, drona Tello a prehľad algoritmov a princípov počítačového videnia.

V druhej kapitole opisujeme návrh novej súťažnej kategórie, prvkov vyvíjanej knižnice a aplikácie pre rozhodcov.

V tretej kapitole znázorňujeme implementované funkcionality knižnice a Android aplikácie

V štvrtej kapitole opisujeme celkový priebeh tohtoročnej súťaže RJS

1. Východiská

V tejto časti práce uvedieme informácie o podobných existujúcich súťažiach, o aktuálnom stave robotických dronov vo svete, konkrétne aj o dronovi Tello a vysvetlíme relevantné poznatky o počítačovom videní.

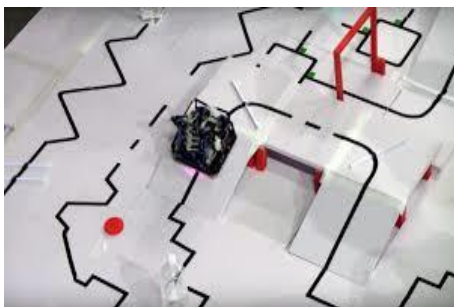
1.1 Robocup Junior

Robocup Junior je medzinárodná súťaž v robotike pre deti a mládež vekovej kategórie do 19 rokov. Je to jedna z kategórií v súťaži Robocup, ktorá sa koná každoročne v rôznych častiach sveta. Cieľom tejto súťaže je podporovať záujem detí a dospelých o vedu, technológie a inžinierstvo a to prostredníctvom vývoja či programovania robotov. [1]

Súťaž má tri základné disciplíny, a to :

I. Robocup Junior Rescue

Úlohou tímu je navrhnuť a naprogramovať robota, ktorých dokáže prekonať rôzne prekážky, zodvihnúť, prípadne prenášať nejaké objekty a tým plniť úlohu záchranára. Efektivita návrhu a naprogramovania robota sa hodnotí v podkategóriách tejto disciplíny, a to v „Rescue Line“ a „Rescue Maze“. Hlavná pointa Rescue line je pohyb robota po čiernej čiare, prekonávanie pripravených nástrah na ceste ako napríklad prekážka v ceste, prerušenie čiary a podobne. Pri Rescue Maze má robot za úlohu prehľadať čo najväčšiu časť bludiska, hľadať obeť a premiestňovať ich do určenej bezpečnej zóny.



Obrázok 1: Herná situácia v disciplínach Rescue Line (vľavo) a Rescue Maze (vpravo) [18]

II. Robocup Junior Soccer

Táto disciplína je zameraná na autonómne hranie futbalu robotov. Tímy navrhnu, sami si postaví a naprogramujú robotov, ktorí dokážu hrať proti iným robotom na ihrisku. Úlohou robotov je pohybovať sa po ihrisku, presne zameriavať loptu, strieľať na bránu ale taktiež aj brániť pred útokmi súpera.



Obrázok 2: Herná situácia v disciplíne Soccer [19]

III. Robocup Junior OnStage

Cieľom tejto disciplíny je vytvoriť nejaké zábavné vystúpenie pre publikum pri ktorom budú využité rôznorodé technológie navrhnuté na interakciu s publikom. Tímy si navrhnu, postaví a naprogramujú napríklad robota – tanečníka, alebo robota – herca, alebo celú skupinu robotov, pričom okrem celkového dojmu z vystúpenia sa hodnotí aj technické prevedenie, softvér a hardvér robotov. Z hľadiska potenciálu rozvoja kreativity je táto kategória najhodnotnejšia. Kreativite sa pri tejto kategórii medze nekladú.



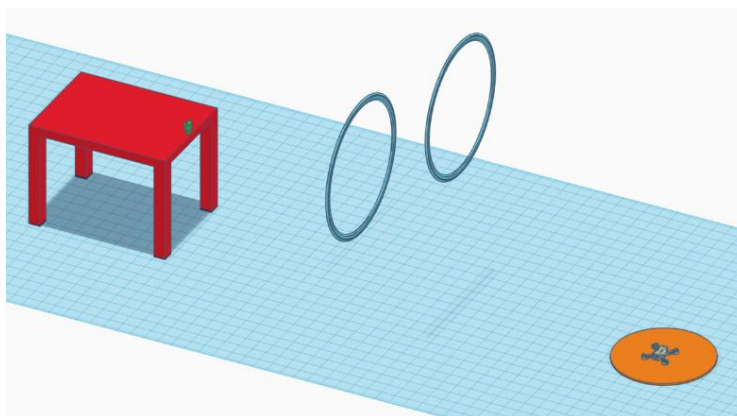
Obrázok 3: Herná situácia v disciplíne OnStage [20]

1.2 Robocup Junior Slovensko

RoboCup Junior sa na Slovensku koná každoročne od roku 2002. Na slovenskom kole RCJ sú rovnaké disciplíny ako na tom medzinárodnom. Okrem hlavných disciplín sa však konajú aj rôzne vedľajšie, menšie disciplíny, ako napríklad programovanie robotického manipulátora podľa zadaných úloh – kategória Robotická ruka, kategória Konštrukcia, alebo kategória Záchranársky dron. [2]

1.2.1 Kategória Dron dodnes

Na slovenskom kole RCJ v ročníkoch 2020-2022 fungovala kategória s názvom „Záchranársky dron“. Úlohou súťažiacich bolo si pripraviť vlastného drona, ktorého sami naprogramovali tak, aby zvládol preletieť prekážkovú dráhu, vyzdvihnúť obeť a úspešne ju doviesť späť na miesto vzletu. Za každé úspešné zvládnutie prekážky boli súťažiacim pripočítané body. Dôležité je poznamenať, že pre vyriešenie úlohy nebolo požiadavkou žiadne využívanie kamery drona. Spôsob riešenia bol teda pomerne jednoduchý a nebol ani tak logickou výzvou, ako skôr správnym zmeraním trasy pre následné zadanie sekvencie príkazov dronu. [3]



Obrázok 4: Ilustrácia mapy pre kategóriu dron ročníkov 2020-2022 [21]

1.2.2 Súťaž MathWorks Minidrone

Súťaž MathWorks Minidrone je súťaž organizovaná autormi programovacích prostredí Matlab-u a Simulink-u. Predmetom súťaže je predstaviť súťažiacim takzvaný Model-Based Design za využitia Simulink-u, pomocou ktorého majú docieľiť navrhnutie algoritmu na naprogramovanie drona – line-followera. Súťaž sa delí na dve hlavné časti, a to :

- Kolo simulácií
- Kolo využitia hardware-u na odprezentovanie funkčnosti algoritmov navrhnutých súťažiacimi

V kole simulácií tímy pracujú na virtuálnom návrhu algoritmu pre drona ktorý má sledovať čiaru. Toto kolo je hodnotené podľa viacerých bodov, ktoré musia uchádzači splniť. Medzi body patria presnosť letu drona po čiare, čas za ktorý sa dronu úspešne podarí zvládnuť mapu a úspešné pristátie na pristávacej plošine. Algoritmy súťažiacich sú hodnotené na viacerých mapách vopred navrhnutých MathWorks inžiniermi. Tímy ktoré prejdú kvalifikáciou tohto kola sú pozvané do nasledujúceho kola, ktoré môže prebehnúť dvoma spôsobmi.

Prvý spôsob je fyzická forma odprezentovania funkčnosti algoritmu za využitia hardware-u, vrátane drona Parrot Mambo mini, ktorý je každému tímu poskytnutý v deň druhého kola. V deň tohoto kola prebieha viacero testovacích spúšťaní drona. Počas tejto doby majú tímy čas na testovanie algoritmov a kalibráciu hardware-u, pričom ešte nie sú bodovo hodnotené. Po testovacích letoch dronov nasleduje hodnotená časť súťaže. Každý tím má sedem pokusov na úspešné zvládnutie trasy. Zisk bodov každému tímu pripisuje rozhodca a závisí od vyššie uvedených podmienok a taktiež podľa počtu úspešne splnených trás.

Druhý spôsob druhého kola súťaže je takzvané virtuálne kolo. Pri tejto forme udalosti tím pošle krátke video ktoré demonštruje riešenie výzvy. Následne kapitáni tímov podstupujú krátke interview o ich riešení. [4]

1.2.3 Existujúce riešenia sledovania čiary dronom

Tématika robota line-followera nie je vo svete novým konceptom. Na internete sa nachádza viacero riešení tejto problematiky. V tejto časti sa pozrieme na riešenie autora Murtaza Hassana. [5]

V tomto existujúcom riešení drona line-followera sa autor algoritmu opiera o teóriu fungovania bežného line-follower robota.

```
def getSensorOutput(imgThres, sensors):  
    imgs = np.hsplit(imgThres, sensors)  
    totalPixels = (img.shape[1] // sensors) * img.shape[0]  
    senOut = []  
    for x, im in enumerate(imgs):  
        pixelCount = cv2.countNonZero(im)  
        if pixelCount > threshold * totalPixels:  
            senOut.append(1)  
        else:  
            senOut.append(0)  
        # cv2.imshow(str(x), im)  
    # print(senOut)  
    return senOut
```

Obrázok 5: Funkcia autorovho algoritmu na určovanie smeru trasy [22]

```
## Rotation  
if senOut == [1, 0, 0]: curve = weights[0]  
elif senOut == [1, 1, 0]: curve = weights[1]  
elif senOut == [0, 1, 0]: curve = weights[2]  
elif senOut == [0, 1, 1]: curve = weights[3]  
elif senOut == [0, 0, 1]: curve = weights[4]  
elif senOut == [0, 0, 0]: curve = weights[2]  
elif senOut == [1, 1, 1]: curve = weights[2]  
elif senOut == [1, 0, 1]: curve = weights[2]
```

Obrázok 6: Spôsob určovania rotácie drona podľa výsledku funkcie v Obrázku 5 [22]

Autor v tomto algoritme zvolil prístup výpočtu potrebnej rotácie znázornený v úryvku kódu na obrázku X. Funkcia `getSensorOutput()` je postavená na logike tradičného line-followera. Funkcia si obraz na ktorý bola prv aplikovaná maska na rozoznávanie farby cesty rozdelí na n -častí (argument funkcie `sensors` predstavuje počet rozdelení obrazu) - sektory. Následne funkcia počíta pre každú rozdelenú časť obrazu, koľko nenulových pixelov sa v nej nachádza. Aproximovaním percentuality prítomnosti nenulových pixelov potom funkcia vracia trojprvkové pole obsahujúce nuly a jednotky. Na obrázku Y je znázornený spôsob vyhodnocovania týchto polí. Do pomocnej premennej `curve` sa zapisuje prvok z pomocného poľa `weights`, ktoré obsahuje konštanty reprezentujúce potrebnú rotáciu drona v daný vyhodnotený obraz.

V teórii tento algoritmus funguje, no z viacerých dôvodov je veľmi nepraktický. Znázornený spôsob funguje za optimálnych podmienok osvetlenia a za nízkej výšky letu drona. Pri vlastných testoch tohto spôsobu bol let veľmi nekoordinovaný. Vo väčšej výške, kedy je detekcia tenšej cesty náročnejšia aj za optimálnych svetelných podmienok, je pri znázornenom výpočte percento nenulových pixelov v ľubovoľnom sektore veľmi variabilné. Toto zapríčiňuje nestabilnú aproximáciu potrebnej rotácie drona.

1.3 Drony vo svete

Drony, či bezpilotné lietajúce zariadenia, sa stávajú výraznou a dôležitou súčasťou mnohých oblastí. Medzi tie patria napríklad geodetické a environmentálne merania, kinematografia, vojenské a bezpečnostné technológie, ale aj rôzne záchranné operácie. Podľa špecifických potrieb sa drony vyrábajú v rôznych veľkostiach, tvaroch, s rôznymi druhmi vybavení ako napríklad špeciálne termokamery a ďalšími inými senzormi.

1.3.1 Využitie dronov pri záchranných operáciách

Prírodné katastrofy sú po celom svete čím ďalej tým frekventovanejšie. Či už sa jedná o zemetrasenia, záplavy, či rôzne iné pohromy, následky bývajú často drvivé. Vo svete sa preto rodia rôzne iniciatívy na zmiernenie dopadov katastrof a následnú výpomoc so záchrannými prácami.

V dnešnej dobe sa už využívajú drony so špeciálnymi termo a infra-red kamerami, vďaka ktorým je možné odhaľovať polohu zranených. Nasadenie záchranárskych dronov trvá pár minút, čo je zlomok oproti času potrebnom pri operácii s helikoptérou. Drony dokážu vďaka pohľadu zhora skenovať rozľahlé, často ťažko prístupné oblasti. [6]

Niekedy však ani takéto drony nie sú využiteľné, kvôli nepriaznivým vizuálnym podmienkam. Nový spôsob priniesla iniciatíva ImPACT Tough Robotics Challenge Program1[7] . Namiesto spoliehania sa na vizuálnu detekciu sa vznikol nový prístup za využívania akustických senzorov s pomenovaním „robot ears“. Tri kľúčové princípy tohto prístupu sú :

1. Sound source localization technology – technológia na lokalizáciu zdroja zvuku pre aproximáciu pôvodu zvuku
2. Sound source separation technology – osamostatnenie individuálnych zvukov pre možnosť vykonávania ďalších vyhodnotení
3. Automatic speech recognition technology – softvér na automatické rozoznávanie ľudskej reči

Tento senzor je schopný detekcie ľudského hlasu s presnosťou na 3 metre. Takáto technológia nájde uplatnenie pri situáciách, kedy je okolie napríklad pri požiaroch husto zadymené, alebo sa zranená obeť katastrofy nachádza pod sutinami budov.

Okrem záchranárskych dronov sú taktiež fascinujúce drony s reproduktorom „Zoe“ od firmy Acecore, ktoré umožňujú šírenie oznamov, prípadne riadenie davu počas nejakej krízovej situácie. Drony takéhoto typu sa využívali už v nedávnej minulosti, a to počas pandémie koronavírusu. Pri lete v nadmorskej výške 100m dosahuje reproduktor hlasitosti 80db, čo sa dá porovnať s hlukom hustej premávky. [8]

1.4 Robotický dron Tello

Tello je malý a ľahký dron od spoločnosti DJI v spolupráci s firmou Ryze Tech. Dron bol navrhnutý aby slúžil ako edukačný prostriedok pre všetkých záujemcov dronov. [9]



Obrázok 7: Dron DJI Ryze Tello [23]

1.4.1 Procesor

Dron Tello je vybavený procesorom Intel Movidius Myriad 2 VPU, ktorý slúži ako jednotka pre rýchle spracovanie obrazu z kamery. Tento procesor je využívaný v miliónoch zariadení v celosvetovom trhu, práve vďaka jeho nízkej spotrebe energie a vysokému výkonu.

1.4.2 Pripojenie na drona

Pripojenie sa na drona je veľmi jednoduché. Prvá voľba je stiahnutie si oficiálnej Tello aplikácie na mobilné zariadenie (Android alebo iOS). Pomocou aplikácie vie používateľ drona ovládať, má prístup k živému obrazu získavanému z kamery, ale taktiež aplikácia umožňuje kalibráciu senzorov a mnoho ďalších funkcionalít.

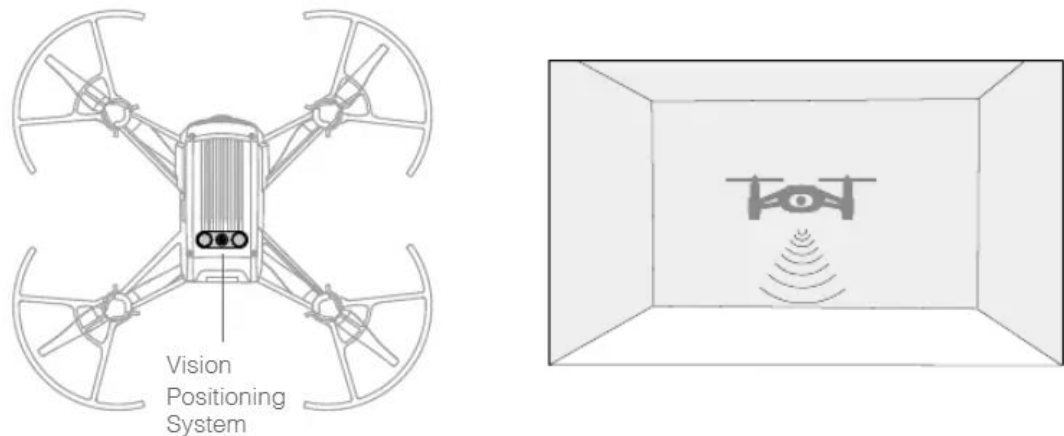
Pripojenie cez počítač funguje na rovnakom princípe, teda pripája sa cez WI-FI. Pre počítač neexistuje dedikovaná aplikácia na ovládanie drona Tello. Pre vývojárov bolo však vyvinuté firmou DJI tzv. Tello SDK (Software Development Kit), čo je istý súbor nástrojov, za ktorého pomoci je možné vytvárať vlastné aplikácie či softvér pre používanie drona Tello.

1.4.3 Kamera

Tello disponuje 5-megapixelovou kamerou, vďaka ktorej možno vytvárať statické snímky či nahrávanie videí v rozlíšení 720p. Kamera je schopná zachytávať plynulé a stabilné videá vďaka vstavanému stabilizátoru obrazu. Podľa voľby pripojenia na drona je taktiež možné ukladať zachytávané snímky buď do Android/iOS zariadenia, či počítača.

1.4.4 Vision Positioning System

Vizuálny polohovací systém, skrátene VPS, pomáha dronu udržiavať sa stabilne na jednom mieste. Skladá sa z dvoch hlavných častí umiestnených na spodnej časti drona, a to z kamery a 3D infračerveného modulu.



Obrázok 8: Lokácia VPS prevzaté z [9]

Najväčšou nevýhodou tohto systému je práve jeho citlivosť. Ak dron vyhodnotí, že sa aktuálne nachádza nad nevhodným povrchom, alebo podmienky letu sú nevhodné, začne sa samovoľne presúvať a hľadať vhodnejší povrch. Systém vyhodnotí napríklad nasledujúce podmienky za nevyhovujúce :

- Lesklý alebo svietiaci povrch
- Priesvitný povrch
- Monochromatický povrch
- Pohybujúci sa povrch
- Dron je príliš nízko alebo naopak príliš vysoko

1.4.5 Collision detection system

Dron Tello taktiež disponuje systémom na detekciu kolízií. V motorčekoch sa nachádzajú senzory, ktoré reagujú na silnejšie vibrácie - nárazy, prípadne na sťaženie či znemožnenie pohybu vrtuliek. Ak systém zaznamená niektorý z uvedených problémov, automaticky vypne všetky motory.

1.4.6 Programovateľnosť

Veľkou prednosťou drona Tello je práve možnosť jeho programovania vo viacerých známych jazykoch ako napríklad Scratch, Python, či Swift. Spojením SDK drona s počítačovým videním sa dajú vytvoriť mnohé zaujímavé projekty.

1.5 Ovládanie drona

Pre komunikáciu s dronom som sa rozhodol používať interface `DjiTelloPy` [13], ktorý využíva oficiálne Tello SDK. Je to open-source knižnica zverejnená pre vývojárov a uľahčuje prácu s dronom. Hlavné funkcionality tejto knižnice pre nás dôležité sú :

1. Ovládanie drona - Knižnica umožňuje posielat' dronu príkazy pomocou Python kódu.
2. Prístup k dátam zo senzorov drona - Dôležitá časť knižnice, vďaka ktorej vieme získať aktuálnu výšku letu či napríklad stav batérie.
3. Prijímanie snímok kamery – Vďaka umožneniu prístupu ku snímkom kamery vieme snímky v našej python aplikácii nie len zobrazovať, ale vykonávať na nich rôzne zložité operácie spracovania obrazu.

1.6 Počítačové videnie

Počítačové videnie je odvetvie počítačovej vedy a umelej inteligencie, ktorého cieľom je vytvárať technológie ktoré umožňujú strojom získavať dôležité informácie z obrázkov, prípadne reagovať želaným výstupom. Zložitosť počítačového videnia podľa vlastností vieme rozdeliť do dvoch kategórií, a to [10]:

Low-level : Sústreďovanie sa na základné aspekty obrazu, ktoré je možné získať jednoduchými technikami. Medzi nízkoúrovňové charakteristiky patria kontúry, hrany, uhly, ale aj napríklad farby.

High-Level : Pri určovaní vysokoúrovňových charakteristík sa využívajú neurónové siete. Táto sekcia sa týka komplexnejšieho strojového učenia na veľkých datasetoch, podľa ktorých vieme následne rozoznávať v obrazoch konkrétne objekty.

V tejto bakalárskej práci sa budeme venovať práve low-level charakteristikám.

Medzi najznámejšie a najpoužívanéjšie knižnice pre počítačové videnie patrí OpenCV (Open Source Computer Vision Library) [14] . Obsahuje širokú škálu funkcií a algoritmov pre spracovanie obrazu, detekciu hrán, porovnávanie charakteristík a mnoho ďalších. Tieto nástroje sú široko využívané v rôznych aplikáciách mnohých oblastí, ako napríklad zdravotníctve, robotike, hernom priemysle, bezpečnosti a mnoho ďalších. OpenCV je dostupná pre viacero programovacích jazykov, ako C++, Python ale aj Java.

1.7 HSV model

HSV model je model reprezentácie farieb, ktorý sa často používa v počítačovom videní, napríklad v oblasti detegovania objektov podľa farby. Farby sú v tomto modeli charakterizované podľa troch kľúčových parametrov :

1. Hue, alebo farebný odtieň, reprezentuje dominantnú vlnovú dĺžku farby.
2. Saturation, alebo nasýtenie, udáva množstvo sivej v pomere k aktuálnemu odtieňu.
3. Value, alebo jas, definuje celkový jas danej farby.

Oproti RGB modelu umožňuje rýchlejšie vyhľadávanie farieb v obrazoch. Oddeluje konkrétnu farbu, teda hue, od nasýtenia a jas, čím nám umožňuje jednoduchšie chápanie a manipuláciu s farbami ako takými.

1.8 Predspracovanie obrazu

Úlohou predspracovania obrazu je zbavenie sa neželaných javov v obraze, ako napríklad šumu, alebo naopak vylepšovať obraz zvýrazňovaním požadovaných častí obrazu, napríklad kontúr objektov. [11]

1.8.1 Segmentácia

Hlavným cieľom je rozdeliť obraz na časti, ktoré sú odlišné od ostatných vzhľadom na farebné vlastnosti, textúru, jas, tvar alebo iné charakteristiky, a ktoré čo najviac zodpovedajú objektom zobrazených na obrázku. Segmentáciu možno rozdeliť na tri skupiny:

- segmentácia založená na hranách
- segmentácia založená na oblastiach
- prahovanie

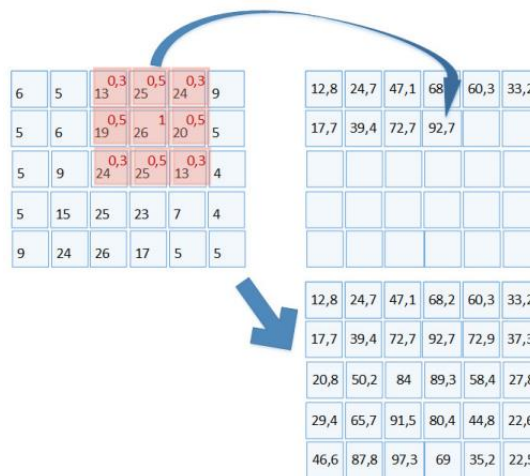
Pri tejto bakalárskej práci sa budeme zaoberať ale hlavne jedným z nich, a to prahovaním. Prahovanie je najjednoduchšia technika segmentovania obrazu, je výpočtovo nenáročná a rýchla. Na segmentovanie sa používa jasová konštanta – prah, aplikovaný buď globálne na celý obraz alebo na jeho časť. Bod, ktorý má hodnotu jasu väčšiu ako daná jasová konštanta sa priradí hodnota 1, iným hodnota 0. Výstupom prahovania je teda binárny obraz.

1.8.2 Filtrovanie obrazu

Pri spracovávaní obrazu rozlišujeme medzi dvomi hlavnými druhmi filtrovania, a to tzv. High-Pass a Low-Pass filtering, alebo horná a dolná priepusť. Vo všeobecnosti, rozdiel medzi nimi je taký, že High-Pass sa zaoberá zaostrovaním, naopak Low-Pass rozmazávaním obrazu.

Základný matematický nástroj pre lineárnu filtráciu je operácia nazývaná konvolúcia. Proces konvolúcie pri počítačovom videní si vieme predstaviť ako

aplikovanie filtračného jadra (matica nepárnej veľkosti, napr. 3x3, 5x5, 7x7 ...) globálne na všetky pixely obrazu.



Obrázok 9: Ilustrácia aplikovania filtračného jadra prevzatá z [11]

Pre nás bude dôležité najmä Low-Pass filtering, teda aplikovanie konvolučnej masky na obraz za účelom získania rozostreného obrazu – takzvané vyhladzovacie filtre. Príklady najpoužívanejších vyhladzovacích filtrov v počítačovom videní :

- Priemerovací filter : členy konvolučnej masky sú definované tak, aby sa dosiahol zakaždým aritmetický priemer okolia bodu daného veľkosťou použitej konvolučnej masky.
- Gaussov filter : Gaussov filter je jeden z najpoužívanejších spôsobov filtrácie obrazu pre jeho zjemnenie.



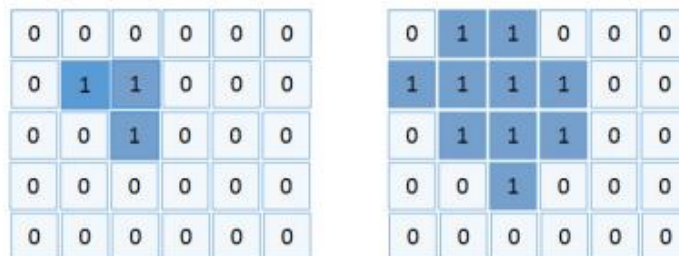
Obrázok 10: Ukážka obrazu pred a po použití Gaussovho filtra

1.8.3 Morfológické operácie – Erózia a Dilatácia

Ďalšou dôležitou súčasťou počítačového videnia sú základné morfológické operácie. Morfológické operácie využívajú mechanizmus podobný filtrácii, kde každý pixel obrazu je nahradený novou hodnotou vypočítanou algoritmom, pričom sú zohľadnené hodnoty jeho topologických susedov. Pre nás najzaujímavejšie budú tzv. :

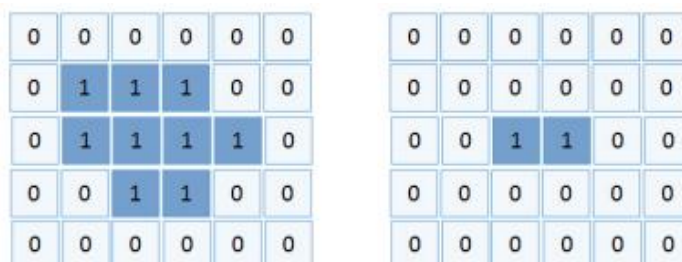
- Binárna Dilatácia
- Binárna Erózia

Binárna dilatácia zväčšuje rozlohu bieleho objektu (alebo jednoducho povedané nenulovému pixelu obrazu) na čiernom pozadí.



Obrázok 11: Porovnanie obrazu pred a po aplikovaní dilatácie prevzaté z [11]

Presný opak binárnej dilatácie, teda erózia, znižuje rozlohu bieleho objektu na čiernom pozadí.



Obrázok 12: Porovnanie obrazu pred a po aplikovaní erózie prevzaté z [11]

Štruktúrálny element

Štruktúrálny element v počítačovom videní je malá šablóna, istá bodová množina, ktorá sa používa pri vykonávaní morfológických operácií na digitálnych obrazoch. Slúži na interakciu medzi pixelmi a ich okolím, a to prostredníctvom aplikácie na každý pixel v obraze. Štruktúrálny element ovplyvňuje výsledok morfológických operácií, ako vyššie spomenuté dilatácia a erózia. Jeho veľkosť a tvar majú vplyv na zachovanie detailov, presnosť, celkový tvar a veľkosť objektov v obraze.

Za postupného použitia erózie a dilatácie s rovnakým štruktúrnym elementom sa môže stať, že výsledný obraz sa bude rovnať vstupnému. Toto však nie je všeobecne platné. Predstavme si prípad, kedy eróziou z obrazu kompletne odstránime nejaký malý objekt. Následnou dilatáciou ho už nevrátíme späť. Táto vlastnosť je často pri počítačovom videní využívaná, pričom nám vznikajú dva nové princípy :

- Morfológické otvorenie : erózia a následná dilatácia

- Morfológické uzavretie : dilatácia a následná erózia

Morfológické operácie, ako je morfológické otvorenie a morfológické uzavretie, vo všeobecnosti zachovávajú veľkosť objektov. Avšak, rozdiely medzi týmito operáciami sú zreteľné najmä v detailoch, ktorých veľkosť sa približuje veľkosti použitého štruktúrného elementu. Princíp otvorenia má vplyv na obraz hlavne taký, že odstraňuje objekty menšie ako štruktúrálny element. Pri uzavretí ide viac o vyplňanie malých „neživých“ oblastí pixelov v častiach nášho obrazu.

1.8.4 Detekcia hrán

Detekcia hrán je dôležitým procesom v oblasti počítačového videnia, ktorý slúži na identifikáciu miest v digitálnych obrázkoch, na ktorých sa vyskytujú zmeny intenzity. Cieľom detekcie hrán je zvýrazniť hranice objektov, kontúry a významné obrysy v obraze. Existuje niekoľko metód detekcie hrán [12] :

Metóda gradientu

- Keď hovoríme o výpočte gradientu intenzity, skutočne sa zaujímame o zmenu intenzity medzi jednotlivými pixelmi. Ak je zmena intenzity medzi pixelmi v blízkosti veľká, hovoríme o silnom gradiente. Naopak, ak je zmena intenzity malá alebo žiadna, hovoríme o slabom alebo nulovom gradiente. Hrany sa nachádzajú na miestach s vysokou hodnotou gradientu. Často sa používa Sobelov operátor, ktorý vypočíta horizontálny a vertikálny gradient pre každý pixel.

Cannyho detekcia hrán

- Využíva kombináciu redukcie šumu, výpočtu gradientu, potlačenia nepotrebných maximálnych hodnôt a prahovania pre získanie presných hrán. Cannyho detekcia hrán poskytuje robustné výsledky a dokáže zachovať jasné hrany pri minimálnom množstve šumu. Práve tento algoritmus budeme využívať pri riešení úloh v tejto práci.

Konvolučné neurónové siete

- S nástupom „deep-learning-u“ sa konvolučné neurónové siete stali účinným nástrojom pre detekciu hrán. Tieto siete využívajú viacvrstvové konvolučné vrstvy na automatické učenie a extrakciu príznakov z obrázkových dát. Príklady takýchto sietí zahŕňajú U-Net, ResNet alebo VGG, ktoré sú často používané pri detekcii hrán v komplexných obrázkoch.



Obrázok 13: Porovnanie obrázku pred a po aplikovaní Canny edge detektora

1.8.5 Minimálny ohraničujúci obdĺžnik

Minimálny ohraničujúci obdĺžnik, často označovaný aj ako „bounding box“, je obdĺžnik, ktorý obklopuje celú kontúru obrazu bez prebytočného prázdneho priestoru. Je definovaný pomocou minimálnych a maximálnych hodnôt X-ovej a Y-ovej súradnice, ktoré určujú rozsah kontúry. Pri analýze obrazu sa často používa na detekciu objektov a určovanie ich polohy. Je jednoduchým spôsobom reprezentácie objektov v obraze a umožňuje rýchle a efektívne spracovanie vizuálnych dát.

1.9 Modul PyZbar

PyZbar je Pythonový modul (knížnica), ktorý poskytuje rozhranie pre prácu s knižnicou ZBar, ktorá bola prv vyvinutá v jazyku C. ZBar je open-source knižnica pre detekciu a dekodovanie QR a čiarových kódov.

1.10 Haar Cascade

Haar Cascade je populárny a efektívny algoritmus v oblasti detekcie objektov, ktorý sa využíva v rôznych aplikáciách v oblasti počítačového videnia a rozpoznávania vzorov. Najčastejšie sa Haar Cascade používa na detekciu tváří, ale dá sa prispôsobiť aj pre detekciu iných objektov, ako sú autá, zvieratá, oči a ďalšie.

Je dôležité poznamenať, že Haar Cascade je algoritmus detekcie objektov, nie rozpoznávania. Poskytuje informáciu o prítomnosti objektu v obraze, ale neidentifikuje jeho identitu. Pre rozpoznanie konkrétneho objektu by bolo potrebné použiť ďalšie techniky a metódy akými sú napríklad neurónové siete..

1.11 Vývoj Android aplikácií

1.11.1 Jazyk a prostredie

Android Studio je integrované vývojové prostredie (IDE) navrhnuté špeciálne pre vývoj aplikácií pre platformu Android. Je to hlavný nástroj, ktorým sa vývojári používajú pri tvorbe a úprave aplikácií pre mobilné zariadenia, ako sú smartfóny a tablety. Je vyvinuté spoločnosťou Google a poskytuje širokú škálu nástrojov a funkcií, ako napríklad emulátor zariadenia, ktoré uľahčujú proces vývoja aplikácií pre Android.

Kotlin je moderný programovací jazyk, ktorý bol vyvinutý spoločnosťou JetBrains. Bol oficiálne predstavený v roku 2011 a od roku 2017 je oficiálnym jazykom pre vývoj aplikácií pre platformu Android. Je to staticky typovaný jazyk, čo znamená, že všetky typy premenných musia byť deklarované vopred. Jeho syntax je jednoduchá, čitateľná a pripomína syntax jazyka Java, čo umožňuje vývojárom ľahký prechod z Javy na Kotlin.

1.11.2 Perzistencia dát

Room je knižnica v jazyku Kotlin, ktorá umožňuje jednoduchú prácu s lokálnou databázou. Je súčasťou Android Jetpacku, ktorý je sadou nástrojov od spoločnosti Google pre vývoj Android aplikácií. Cieľom Room-u je poskytnúť spoľahlivé a efektívne riešenie pre ukladanie dát na mobilných zariadeniach. Room poskytuje abstrakciu nad SQLite databázou a umožňuje vývojárom manipulovať s dátami pomocou objektovo-orientovaného rozhrania. Ponúka anotácie, ktoré umožňujú definovať štruktúru tabuliek a vzťahy medzi nimi priamo v kóde.

2. Návrh

2.1 Nová kategória Dron

Súčasťou zadania tejto bakalárskej práce je návrh novej disciplíny Dron pre Robocup Junior Slovensko 2023. Ako som v časti 1.2.1 napísal, doterajšia kategória dron nemala s autonómnym letom nič spoločné. Nová úloha riešiteľov disciplíny dron sa stále týka tematiky zachraňovania obete. Dron sa však už po novom musí orientovať autonómne, teda disciplínou sa nedá úspešne prejsť naprogramovaním letu drona „napevno“.

Úlohou drona je vzlietnuť zo štartovacej pozície, letieť po ceste, reagovať na rôzne geometrické útvary poukladané pozdĺž nej a posnažiť sa zachrániť obeť, odniesť ju na koniec cesty a úspešne pristáť na vyznačenej ploche.

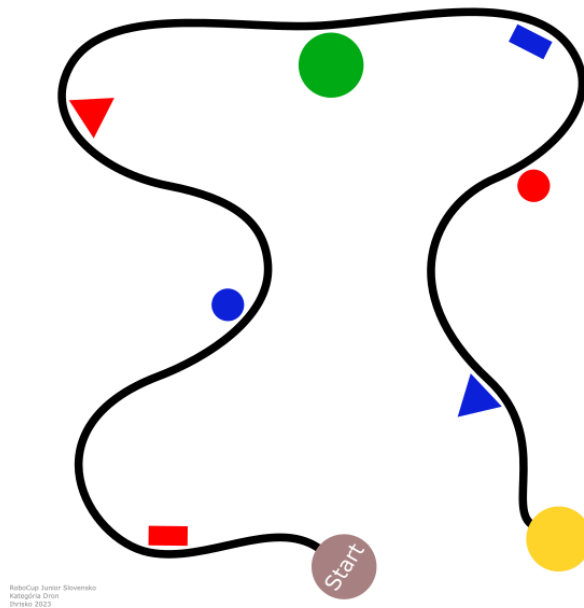
Mapa bude rozmerov 4 metrov štvorcových. Aby sa počítačovým videním dala cesta dobre rozoznávať, skladá sa z dvoch hlavných častí. Čierna čiara šírky 5 cm a svetlý podklad (optimálne biely), ktorý bude mať šírku aspoň 25 cm z oboch strán čiernej čiary. Objekty (geometrické útvary, konkrétne popísané nižšie) budú ponalepované na podklade, najviac 30 cm od čiernej čiary, tak aby bolo možné ich kamerou snímať.

Objekty (geometrické útvary) sú rôznych druhov, môžu to byť trojuholníky, obdĺžniky, ale aj kruhy viacerých farieb. Úlohou riešiteľa je naprogramovať drona tak, aby vedel rozoznať o aký konkrétny útvar sa jedná + akej farby bude a aby vykonal aktivitu korešpondujúcu s útvarom :

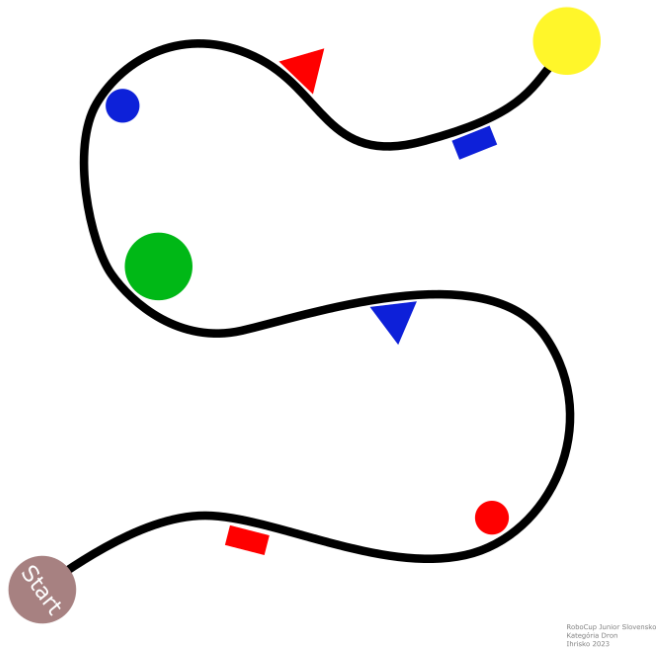
- Červený obdĺžnik (rozmery 24 cm x 12 cm) = dron začne obraz nahrávať, pokračuje v ceste
- Červený kruh (polomer kruhu je 10 cm) = dron zistí, že pred ním sa na ceste nachádza pomyselná prekážka a nadletí ju (bude po ceste pokračovať v ceste vo výške aspoň 2m kým neprekoná vzdialenosť 1m)
- Červený trojuholník (rovnostranný trojuholník s dĺžkou strany 15 cm) = dron sa otočí o 360 stupňov vľavo a pokračuje v ceste
- Modrý obdĺžnik (rozmery 24 cm x 12 cm) = dron prestane obraz nahrávať a video uloží, pokračuje v ceste

- Modrý kruh (polomer kruhu je 10 cm) = dron zistí, že pred ním sa na ceste nachádza pomyselná prekážka a podletí ju (bude po ceste pokračovať v ceste vo výške max 1m kým neprekoná vzdialenosť 1m)
- Modrý trojuholník (rovnostranný trojuholník s dĺžkou strany 15 cm) = dron sa otočí o 360 stupňov vpravo a pokračuje v ceste
- Zelený kruh (s polomerom 20 cm) = v strede sa nachádza obeť. Dron sa vycentruje presne do stredu kruhu, pokúsi sa zodvihnúť obeť, následne sa vráti na cestu a pokračuje v nej.
- Žltý kruh (polomer 20 cm) = dron si uvedomí, že je na konci cesty. Vycentruje sa do stredu a pristane.
- Každý druh útvaru sa vyskytne najviac jeden raz, ale poradie a rozmiestnenie nie je vopred známe.
- V novej súťažnej disciplíne Dron sú súťažiacim body pripisované podľa nasledujúcich pravidiel :
 - Odštartovanie – úspešné je také, kedy dron vzlietne, ostane stáť 5 sekúnd na mieste a následne sa pomaly začne hýbať vpred (5 bodov)
 - Presný let drona pozdĺž stredom čiernou čiarou – 15 bodov, za každé výrazné vybočenie z cesty -5 bodov (menšie nepresnosti budú tolerované)
 - Rotácia drona podľa zákrut – 10 bodov ak je dron nasmerovaný v smere čiernej čiary väčšinu času, 5 bodov ak dodržiava nasmerovanie s občasnými nepresnosťami, 0 bodov ak je dron väčšinu času nasmerovaný nesprávne
 - Ak team úspešne ukáže porote nahrávku, na ktorej začiatku je vidno červený obdĺžnik a na konci je vidno modrý – 10 bodov
 - Za každú otočku alebo úspešné vyhnutie sa prekážke (červené a modré kruhy/trojuholníky) – 5 bodov za každé
 - Pristátie – 10 bodov ak pristane v strede, 8 bodov ak nepristane v strede ale nevyčnieva, 5 bodov ak nejaká časť drona vyčnieva, 0 bodov ak pristane mimo kruhu (+10 bodov ak úspešne prinesie aj obeť)

Každý tím po znázornení a odprezentovaní svojho riešenia podstúpi aj takzvané technické interview. Počas interview bude od tímom zvoleného kapitána očakávané, že bude schopný odpovedať rozhodcovi na otázky ohľadom ich algoritmu a taktiež bude hodnotená aj dokumentácia kódu, pričom každý tím bude môcť za tento rozhovor dosiahnuť maximálne 30 bodov.



Obrázok 14: Navrhnutá mapa 1



Obrázok 15: Navrhnutá mapa 2

Na priložených obrázkoch vyššie sú zobrazené dve mapy. Pre ich vytvorenie sme využili vektorový grafický editor Inkscape. Výroba navrhnutých máp bola zasponzorovaná združením Robotika.sk.

2.2 Knižnica pre programovanie drona

Kľúčovou úlohou tejto práce je pripraviť balík software-u, ktorý budú riešitelia súťaže využívať na úspešné naprogramovanie zadania disciplíny Dron. Keďže veková kategória súťažiacich Robocup Junior Slovensko je tvorená žiakmi základných a stredných škôl na Slovensku, prehľadnosť a jednoduchosť kódu sú hlavnými požiadavkami pre tento software. Hlavnými dvoma prvkami tvorby tejto knižnice bude kombinovanie existujúcich technológií na komunikáciu s dronom – djiellopy a technológie na prácu s počítačovým videním – OpenCV. Cieľom tvorby tejto knižnice bude vytvoriť balík znovu použiteľných komponentov, ktoré budú do budúcnosti slúžiť nie len riešiteľom súťaže Robocup Junior Slovensko, ale aj pri rôznych iných projektoch využívajúcich drona Tello a počítačové videnie.

2.2.2 Návrh tried a skriptov

Zadefinujeme si niekoľko tried, ktorými budeme vedieť skonštruovať napríklad aj algoritmus 2.2.1

Abstraktná trieda Shape

Abstraktná trieda Shape bude reprezentovať n-uholník. Trieda bude niest' funkcionalitu rozpoznávania útvarov pomocou analýzy kontúr na obraze získaného z kamery. Shape bude abstrakciou ďalších tried ako napríklad Circle alebo Triangle, ktoré budú mať vlastné špecifikácie.

Trieda QRCode

Táto trieda, podobne ako abstraktná trieda Shape, bude obsahovať metódu, ktorá vyhodnotí či sa v obraze nachádza QR kód. Ak v obraze QR kód nájde, vráti z neho prečítané dáta vo forme Stringu.

Trieda ShapeRecognizer

V tejto triede budú inštanície tried konkrétnych objektov, ktoré chceme v obraze hľadať. Trieda bude obsahovať, ktorá pre všetky masky vyrobené pre daný obraz skontroluje, či sa v ňom daný geometrický útvar nenachádza.

Trieda PathRecorder

Trieda bude slúžiť na začatie a zastavenie nahrávania videa drona.

Trieda PathFollower

Bude to hlavná trieda, ktorá bude slúžiť na komunikáciu s dronom, respektíve posielanie príkazov dronu. Bude obsahovať inštancie tried PathRecorder, Shaperecognizer a Movement

Trieda MaskMaker

Táto trieda bude obsahovať statické metódy pre vytvorenie masiek potrebných farieb pre nami zadaný frame.

Trieda Movement

Bude to pomocná dátová trieda, kde budú uchovávané hodnoty potrebného pohybu drona.

Súbor utils.py

V súbore utils.py sa budú nachádzať viaceré všeobecne využiteľné funkcie, ako napríklad funkcia na porovnávanie dvoch obrazov, vypočítavanie potrebnej rotácie drona, či jeho potrebnej translácie.

2.2.3 Návrh jazyka pre QR kódy

Aby dron vedel, aké pohyby má vykonať pri detekcii QR kódov, je dôležité si zadať istý jazyk, ktorému bude rozumieť. Zadeľujme si preto pár príkazov :

F = pohyb vpred

B = pohyb vzad

L = translácia vľavo

R = translácia vpravo

U = pohyb nahor

D = pohyb dolu

RL = rotácia vľavo

RR = rotácia vpravo

LAND = pristátie

Príklad sekvencie príkazov pre drona :

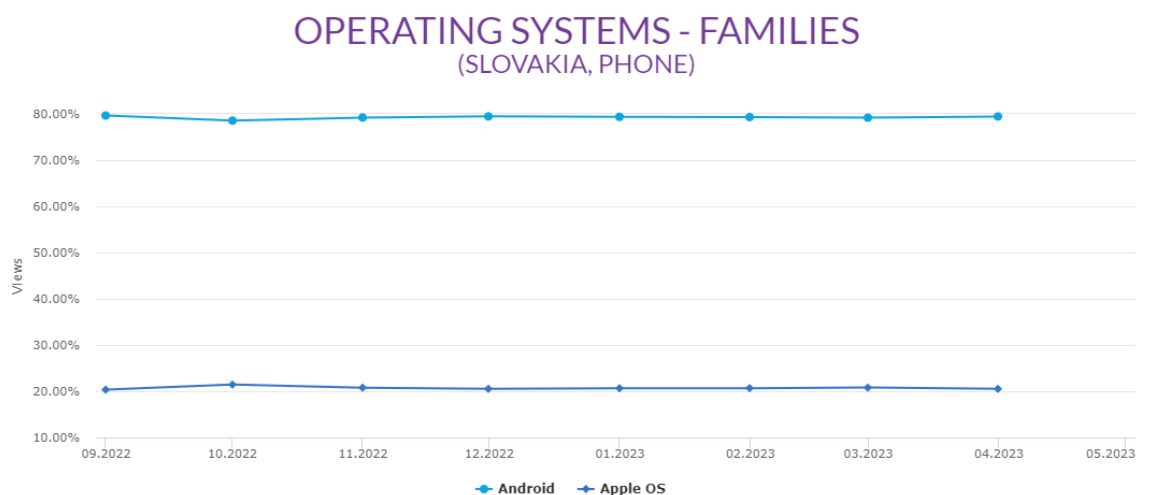
F 50+R 40+B 20

Dron bude plniť príkazy po poradí, to znamená že najprv poletí vpred 50cm, potom doprava o 40 cm a následne dozadu 20cm. Je dôležité aby sa syntax dodržiavala, teda smer a požadovaná vzdialenosť boli oddelené medzerou a nasledujúce príkazy oddelené symbolom +.

2.3 Android aplikácia pre rozhodcov súťaže Robocup Junior Slovensko

Každá disciplína súťaže má svojich rozhodcov. Títo rozhodcovia majú k dispozícii hárky A4 papierov, ktorá pre nich slúži ako databáza výsledkov disciplín. Spôsob zapisovania na papier je dlhodobo zaužívaný, no z praktického hľadiska nie veľmi optimálny. Pri nešťastnej náhode sa môžu papiere s výsledkami stratiť či nejakým spôsobom znehodnotiť. Po uzavretí hodnotenia súťaže sa výsledky všetkých disciplín manuálne zapisujú do Excel súborov, ktoré sa neskôr zverejňujú na webovej stránke súťaže. Cieľom bude tento proces digitalizovať a tým uľahčiť prácu rozhodcom.

Pri navrhovaní aplikácie prišiel do úvahy koncept webovej aplikácie, no ten bol eventuálne zavrhnutý z dôvodu, že nie každý človek má na svojom zariadení neobmedzený prístup na internet. Navyše podľa štatistík, ku dnešnému dňu na Slovensku pre smartfóny platí, že výrazne dominuje operačný systém Android [15]. Z týchto dôvodov som zvolil vyvinutie aplikácie pre rozhodcov v tejto doméne.



Obrázok 16: Graf znázorňujúci zastúpenie operačných systémov v smartfónoch na Slovensku prevzatý z [15]

2.3.1 Šablóna hodnotenia

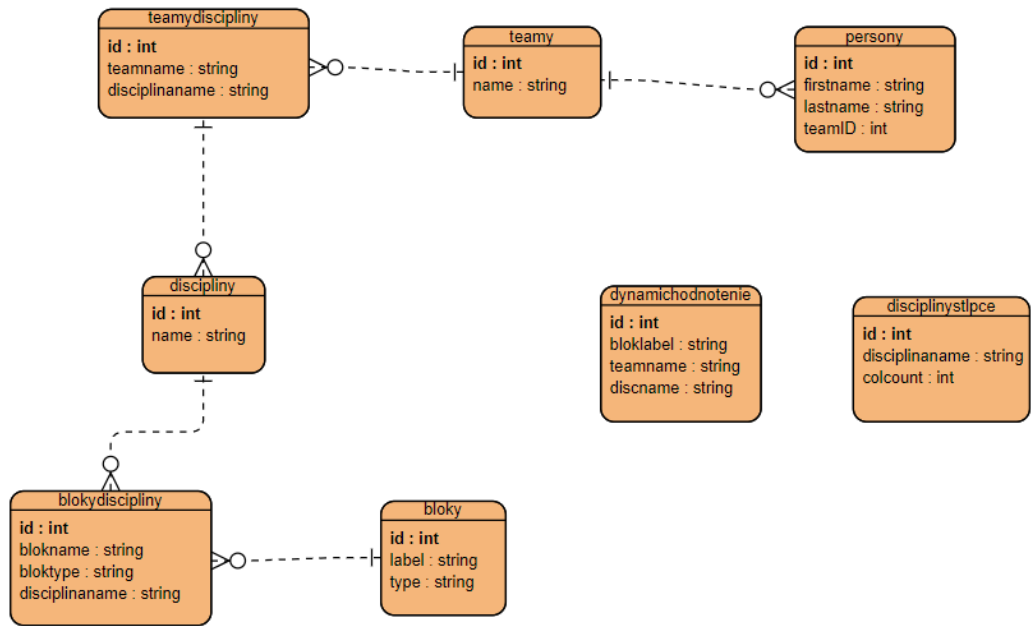
Rôzne disciplíny majú rôzne body ktoré sa hodnotia. Pri jednej disciplíne sa môže hodnotiť presnosť robota, pri druhej môže byť väčší dôraz na jeho rýchlosť. Tieto rôzne celky, ktorých body sú rozličné pre každú disciplínu, budeme označovať ako šablóny. Ako jednoduchý príklad šablóny si môžete pozrieť spôsob bodovania mnou vytvorenej novej disciplíny Dron v časti 2.1.

2.3.2 Požiadavky funkcionalít aplikácie

Vopred známe mená tímov a disciplín, spolu s informáciou ktorý tím sa zúčastní ktorej disciplíny sú zverejnené na stránke súťaže. Stránka taktiež umožňujú stiahnuť si CSV súbor s týmito informáciami. Požiadavky pre aplikáciu sú nasledovné :

1. Aplikácia bude schopná spracovať CSV súbor s informáciami uvedenými vyššie a uložiť si ich.
2. Aplikácia bude schopná spracovať konfiguračný súbor, v ktorom budú pozapísované šablóny hodnotení disciplín.
3. Aplikácia umožní užívateľovi vybrať si tím a disciplínu, pre ktorú bude chcieť hodnotenie vytvoriť.
4. Po vybratí tímu a disciplíny užívateľ bude môcť vyplniť bodovací formulár, ktorý bol dynamicky vygenerovaný podľa šablóny konkrétnej disciplíny a ktorý sa do zariadenia uloží.
5. Aplikácia umožní exportovať výsledky disciplíny do súboru vo formáte CSV, ktorý sa uloží medzi dokumenty zariadenia.
6. Aplikácia bude umožňovať perzistenciu dát, t.j. pri uzamknutí či vypnutí smartfónu sa uložené dáta nestratia.

2.3.3 Zjednodušený Entitno-Relačný model databázy aplikácie



Obrázok 17: Vizualizácia databázy aplikácie

3. Implementácia

V tejto kapitole si objasníme aplikovanie algoritmov počítačového videnia, pozrieme sa na ukážky kódu projektu, analyzujeme problémy ktoré pri implementovaní vznikli a vysvetlíme fungovanie drona.

3.1 Aplikovanie knižnice djitellopy

3.1.1 Pripojenie sa na drona

Prvým najdôležitejším krokom je naviazanie komunikácie s dronom. Vďaka existujúcej knižnici djitellopy nám je komunikácia s dronom veľmi uľahčená. Za použitia knižnice djitellopy vytvoríme inštanciu triedy Tello. Jednoduchým zavolaním metódy `.connect()` sa na drona pripojíme. Potom zavoláme metódu `.streamon()`, ktorá spustí zdieľanie videa zo zabudovanej kamery drona, čím nám umožní jeho zachytávanie a zobrazenie v reálnom čase.

```
from djitellopy import tello
tello = tello.Tello()
tello.connect()
tello.streamon()
```

Obrázok 18: Znáznornenie kódu pre pripojenie sa na drona

3.1.2 Získavanie obrazu z drona

Pre získanie aktuálneho obrazu z videokamery využívame v cykle metódu knižnice djitellopy :

```
while True:
    curr_frame = tello.get_frame_read().frame
```

Obrázok 19: Znáznornenie kódu pre získanie snímok kamery

Do premennej `curr_frame` sa nám zapíše volaním znázornenej metódy aktuálny obraz. Premenná `curr_frame` je v tomto bode typu 2D numpy poľa. Keďže je obraz reprezentovaný dvojrozmerným poľom, vieme naň následne aplikovať ďalšie chcené operácie.

3.2 Problematika nasmerovania kamery

Kamera drona Tello je staticky nasmerovaná dopredu. Splniť nové zadanie disciplíny Dron by preto bolo nemožné. Bolo potrebné nájsť nejaký spôsob, ktorým by sme docielili, aby dron vedel snímať to čo je pod ním. Šikovný spôsob ako tento problém obísť, je využitie technológií 3D tlače. Na internete sú dostupné modely držiakov pre drona Tello, ktoré slúžia práve pre túto problematiku. [16] Na tento držiak vieme umiestniť malé zrkadielko, prípadne malý ústrižok zrkadlovej pásky, vďaka ktorému dron vidí priamo pod seba.

3.3 Problematika Vision Positioning System-u drona Tello

Ako je popísané v časti 1.4.4, VPS je výhodou pri štandardom používaní Tella, no v našom prípade, kedy je pohyb drona ovládaný príkazmi vyhodnotenými podľa obrazu, skôr nevýhodou. Na začiatku testovania môjho algoritmu sa dron správal veľmi neočakávane. Pričom kontrolné konzolové výpisy príkazov tvrdili, že mnou navrhnutý algoritmus funguje, dron veľmi často strácal koordináciu. Napokon som si však všimol koreláciu pri všetkých chybách letu drona. Podobnosť bola v tom, že dron sa stabilizoval vždy až v bode, kde končil rozsah mapy (časť 2.1). Keďže VPS prestáva fungovať v momentoch, kedy je plocha pod dronom reflexná či takmer monochromatická, navrhnutá mapa potrebovala isté vylepšenia. Nimi boli objekty s textúrou, farieb ktoré by neprekážali pri vyhodnocovaní geometrických útvarov či cesty.



Obrázok 20: Ukážka pridaných predmetov na mapu pre vyriešenie problému 3.3

3.4 Rozoznávanie farebných geometrických objektov

Triedu Shape, popísanú v časti 2.2.2, sme aplikovali podľa návrhu. Dôležitá je jej metóda `is_in_frame(self, masked_frame)`, ktorej argument `masked_frame` je predspracovaný obraz prahovaním pre istú farbu. Návratovou hodnotou danej metódy je boolean a funguje nasledovne:

1. Pre nájdenie kontúr, respektíve hrán v obraze používame OpenCV funkciu `cv2.findContours()`. Jej argumentami sú daný obraz ktorý chceme analyzovať, `cv2.RETR_TREE`, ktoré označuje všetky kontúry a následne `cv2.CHAIN_APPROX_SIMPLE`, ktoré značí ukladanie rohových bodov nájdených kontúr. Návratové hodnoty sú nami chcené kontúry a hierarchia kontúr, ktorá udáva vzťahy medzi nimi.

2. Prechádzame cyklom zistené kontúry v našom obraze.

3. Aby sme spresnili vyhodnocovanie nájdeného objektu v obraze, využijeme `cv2.contourArea()` pre každú kontúru, aby sme sa zaoberali len prvkami väčších rozmerov.

4. Použijeme výpočet obvodu uzatvorených kontúr

5. Následne využijeme funkciu `cv2.approxPolyDP()`, ktorej argumenty sú nami zvolená kontúra, presnosť aproximácie (čím menšia konštanta, tým vyššia presnosť a naopak čím väčšia konštanta, tým hrubšia aproximácia) a finálne hodnota `True` ktorá značí, že kontúra je uzavretá.

6. Keďže návratová hodnota funkcie `cv2.approxPolyDP()` je pole bodov, to, o aký geometrický útvar sa jedná vieme aproximovať dĺžkou tohto poľa.

```

class Shape:
    def __init__(self):
        self.id = "Abstract"

    def get_angles(self):
        pass

    def is_in_frame(self, masked_frame):
        count_of_angles = self.get_angles()
        contours, hierarchy = cv2.findContours(masked_frame,
                                              cv2.RETR_TREE,
                                              cv2.CHAIN_APPROX_SIMPLE)

        for c in contours:
            area = cv2.contourArea(c)
            if area > 3000:
                peri = cv2.arcLength(c, True)
                approx = cv2.approxPolyDP(c, 0.03 * peri, True)
                pocet_uhlov = len(approx)
                if type(count_of_angles) == range:
                    if pocet_uhlov in count_of_angles:
                        return True

                elif pocet_uhlov == count_of_angles:
                    return True
        return False

```

Obrázok 21: Ukážka kódu triedy Shape

3.5 Pohyb drona po čiare

Popis algoritmu sledovania čiary dronom si vieme definovať v nasledovných bodoch :

1. Získanie potrebnej masky pre farbu čiary. (časť 1.8.1 kapitoly Východiská)
2. Analýza obrazu a lokalizácia najväčšej kontúry nachádzajúcej sa v maskovanom obraze
3. Výpočet súradníc stredu najväčšej kontúry maskovaného obrazu.
4. Výpočet súradníc minimálneho ohraničujúceho obdĺžnika (časť 1.8.5 kapitoly Východiská) najväčšej kontúry nachádzajúcej sa v maskovanom obraze.
5. Aproximácia uhla potrebnej rotácie drona
6. Zaslanie potrebných vypočítaných hodnôt dronu

3.6 Súbor funkcií

V súbore utils.py sa nachádzajú kľúčové funkcie aj pre riešenie novej disciplíny dron. Medzi nich patria napríklad :

- Funkcia `compare_two_frames(first, second)`, ktorej parametre reprezentujú dva obrázky. Jej návratová hodnota je boolean `True` ak sú obrázky rozličné, `False` ak snímky zhodujú. Využitie táto funkcia nájde napríklad pri nekonečnom cykle. Aby sme sa vyhli opakovaným operáciám na tom istom obraze, stačí, že sa potrebné výpočty budú vykonávať v rýchlosti FPS zdroja snímok.
- Funkcia `get_begin_end_of_bounding(x, y, w, h, masked_frame)`, kde parametre `x,y,w,h` predstavujú súradnice minimálneho ohraničujúceho obdĺžnika niektorej kontúry a `masked_frame` predstavuje binárny obraz. Návratovou hodnotou je n-tica, ktorej prvé dva prvky reprezentujú súradnice najvyššie položeného nenulového bodu, druhé dva toho najnižšie. Funkcia slúži ako výpomoc pri aproximovaní potrebného uhla rotácie drona.

3.7 Aplikácia pre rozhodcov – implementácia

Implementácia aplikácie postupovala podľa požiadaviek funkcionalít, časť 2.3.3 kapitoly Návrh.

3.7.1 Perzistencia dát aplikácie

Pre perzistenciu dát aplikácie využívame Room databázový systém opísaný v časti 1.9.2. kapitoly Východiská.

3.7.2 Spracovanie CSV súboru s informáciami o disciplínach a tímo

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Team	Konštrukci	Ruka	Dron	Futbal/Sor	Futbal/Sor	Futbal 1N	Futbal Sim	RescueLin	RescueMa	Rescue Er	RescueCo	Tanec/Dance
2													
3	!focus()		prihlásení					prihlásení	prihlásení	prihlásení	prihlásení	prihlásení	
4	AutoPrikratki					prihlásení							
5	Belasí			prihlásení									
6	Ben Dover												prihlásení
7	Bern964_r	prihlásení							prihlásení				
8	Bern964_t	prihlásení							prihlásení				
9	Bikeri	prihlásení							prihlásení				
10	Bobalky	prihlásení											
11	CASTLEBOT										prihlásení		
12	ELEKTRO Rescue									prihlásení			
13	Flores	prihlásení											prihlásení
14	GALEJE An	prihlásení							prihlásení				
15	GJH QBB				prihlásení								
16	Gothic Serpents			prihlásení									
17	Guľvas										prihlásení		
18	Inštrumen	prihlásení											
19	Inžinieri	prihlásení							prihlásení				
20	KickTronZ				prihlásení								
21	Lego Paná	prihlásení											

Obrázok 22: Formát zápisu dát súboru

V prvom riadku súboru, teda hlavičke, sa nachádzajú názvy stĺpcov. Prvý stĺpec v tomto riadku vždy reprezentuje meno tímu, nasledujúce mená disciplín súťaže. Ak je niektorá bunka súboru označená „prihlásení“, znamená to, že konkrétny tím sa registroval na danú disciplínu a chystá sa ju riešiť.

```
fun readCSVFile(): MutableList<Pair<String, String>> {
    val reader = BufferedReader(InputStreamReader(FileInputStream(pathh)))
    val header = reader.readLine()
    val headerList = header.split(";").toMutableList()
    Log.i("teamy",headerList.toString())
    headerList.removeAt(0)
    var line: String? = reader.readLine()
    var j = 0
    var arrTeamDisc = mutableListOf<Pair<String, String>>()
    while (line != null) {
        if (line.isNotBlank()) {
            val currLineList = line.split(";").toMutableList()
            val team = currLineList[0]
            currLineList.removeAt(0)
            for (i in currLineList) {
                if (i != "") {
                    arrTeamDisc.add(
                        Pair(
                            team.replace("\"", ""),
                            headerList[j].replace("\"", "")
                        )
                    )
                }
                j++
            }
        }
        j = 0
        line = reader.readLine()
    }
    return arrTeamDisc
}
```

Obrázok 23: Ukážka metódy readCSVFile()

Pre prečítanie a uloženie informácií zo zvoleného súboru sme vytvorili triedu CSVReaderTeamsDisc.. Algoritmus jej metódy readCSVFile() vyhodnotí pre každý tím na aké disciplíny je prihlásený a povytvára súvisiace páry. Návratová hodnota funkcie je pole dvojíc, ktorých prvý člen reprezentuje meno tímu a druhý meno disciplíny.

3.7.3 Spracovanie konfiguračného súboru so šablónami hodnotení

V súťaži Robocup Junior Slovensko má každá disciplína rozličné šablóny. Tieto body nemusia byť v budúcich ročníkoch súťaže identické s aktuálnymi, a preto bolo potrebné od aplikácie odlúčiť preddefinované generovanie používateľského rozhrania. Zvolili sme preto prístup konfiguračného súboru, ktorý nesie informácie o šablónach. Ako formát konfiguračného súboru sme zvolili JSON, ktorý je populárny pre svoju jednoduchosť a ľahkú čitateľnosť pre ľudí aj počítače. Tento súbor nesie informáciu názvov disciplín a ich potrebných prvkoch. Súbor sa dá meniť, teda je možné pridávať či odoberať celé disciplíny a nastavovať im potrebné body.

```
{
  "discipliny": [
    {
      "name": "Dron",
      "description": "Lorem Ipsum",
      "usedBlocks": [
        {
          "type": "plusminus",
          "label": "Výrazné vybočenia"
        },
        {
          "type": "trimoznost",
          "label": "360/Nadlet/Podlet"
        },
        {
          "type": "numinput",
          "label": "Interview"
        },
        {
          "type": "checkbox",
          "label": "Recording"
        }
      ]
    },
    {
      "name": "Ruka",
      "description": "Lorem ipsum",
      "usedBlocks": [
        {
          "type": "plusminus",
          "label": "Test Ruky"
        }
      ]
    }
  ]
}
```

Obrázok 24: Ukážka konfiguračného súboru JSON pre disciplíny Dron a Ruka

3.7.4 Vlastné bloky

V doméne tvorby Android aplikácií je základným stavebným blokom používaným na tvorbu užívateľského rozhrania takzvaný „View“. „View“ predstavuje jeden vizuálny element, ako napríklad tlačidlo, textové pole, obrázok, zoznam a mnoho ďalších. Pri vývoji sme si vytvorili vlastné typy „Views“.

```
class PlusMinus(context: Context) : AbstractView(context) {  
  
    override var label = ""  
    override var currVal = 0  
    private val size = 130  
  
    private val verticalLayout = LinearLayout(context)  
    private val linearLayout = LinearLayout(context)  
    private val plusko = ImageButton(context)  
    private val minusko = ImageButton(context)  
    private val currValView = TextView(context)  
    private val popis = TextView(context)
```

Obrázok 25: vlastná trieda PlusMinus

Trieda PlusMinus zobrazená na obrázku 25 využíva viacero štandardných stavebných blokov užívateľského rozhrania ako napríklad ImageButton a TextView.



Obrázok 26: Vizualizácia vlastného stavebného bloku PlusMinus

3.7.5 Use case : Vytvorenie nového hodnotenia

Po výbere želaného tímu a disciplíny, pre ktoré chce rozhodca vytvoriť nové hodnotenie, bude po stlačení tlačidla „pridať hodnotenie“ presunutý na novú obrazovku. Na tejto obrazovke sa zobrazia dynamicky vygenerované body hodnotenia danej disciplíny podľa šablóny preddefinovanej v konfiguračnom súbore (časť 3.7.2.). Po vyplnení želaných informácií užívateľ stlačí tlačidlo „Uložiť“

hodnotenie“, ktorým sa do korešpondujúcich tabuliek databázy uložia potrebné informácie. Užívateľ bude taktiež vrátený na hlavnú obrazovku aplikácie.

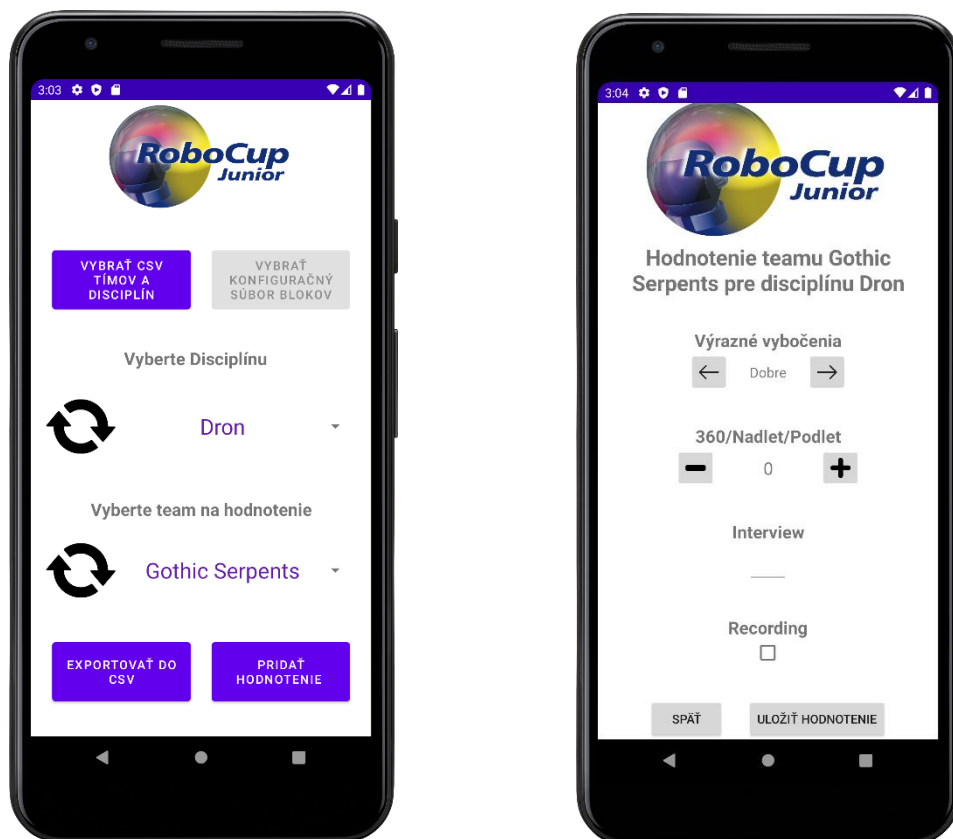
3.7.6 Export hodnotení disciplíny do súboru CSV

Na hlavnej obrazovke aplikácie sa nachádza tlačidlo „Exportovať do CSV“. Stlačením tlačidla sa do priečinka /Documents/robocup zariadenia uloží súbor s hodnoteniami vybranej disciplíny. Ukážkou výstupného súboru hodnotení kategórie Dron je nasledujúceho formátu :

	A	B	C	D	E	F
1	Meno Teamu ▾	Výrazné vybočenia ▾	360/Nadlet/Podlet ▾	Interview ▾	Recording ▾	Suma bodov ▾
2	!focus()	2	5	0	5	12
3	Gothic Serpents	1	5	0	0	6
4						

Obrázok 27: Ukážka výstupného CSV súboru

3.7.7 Používateľské rozhranie aplikácia



Obrázok 28: Ukážka hlavnej stránky aplikácie (vľavo) a stránky pre hodnotenie (vpravo)

3.7.8 Komplikácie s aplikáciou

Aplikácia je kompilovateľná pre verzie Android API 26 a vyššie. Pri novších verziách Androidu však nastávajú časté problémy pri práci so súbormi externého úložiska smartfónu. Novšie systémy majú prísnejšie podmienky povolení a v prípade Android API 33 vývojárom už nie je vôbec povolené pristupovať k bežným priečinkom externej pamäte ako napríklad /Documents. Ani po otestovaní viacerých spôsobov sme túto problematiku úplne nevyriešili a preto pri budúcich verziách aplikácie budú potrebné mierne úpravy práce s úložiskom pre udržanie životnosti aplikácie.

4. Priebeh kategórie Dron ročníka 2023 Robocup Junior Slovensko

Súťaž Robocup Junior Slovensko sa tohtoročne konala na Strednej priemyselnej škole elektrotechnickej v Prešove v termínoch 12.4.-14.4. Nová kategória Dron mala dve oficiálne kolá.

Prvé kolo sa odohrávalo 12.4. Počas tohto dňa mali súťažiaci dovolené testovať let drona na prvej, tréningovej mape bez straty bodov. Súťažiaci mali pre vyriešenie kategórie Dron k dispozícii celú knižnicu funkcií a tried, ktorá bola ešte pred súťažou zverejnená v repozitári oficiálnej softvérovej podpory pre Robocup Junior Slovensko na Github-e. Jej využitie však nebolo nutnou podmienkou úspešného splnenia disciplíny. Počas tréningového dňa mali súťažiaci dovolené konzultácie pri riešení s vedúcim kategórie.

Druhé kolo sa odohrávalo 13.4. Oficiálna hodnotená časť kategórie Dron sa začala o 11:00, do tohto bodu mali ešte tímy priestor svoje algoritmy testovať. Následne boli mapy vymenené. Hodnotenie tímov bolo založené na zadaných podmienkach časti 2.1. kapitoly Návrh. Každý tím mal 5 pokusov, pričom hodnotený bol najlepší z nich. Každý zo zúčastnených tímov po predvedení svojho riešenia podstúpil aj technické interview, počas ktorého tím odpovedal na dodatočné otázky ohľadom ich riešenia. Na súťaži bol pri hodnotení taktiež otestovaný prototyp aplikácie pre rozhodcov.

5. Záver

Úlohou tejto práce bolo navrhnuť a vytvoriť knižnicu pre robotického drona Tello s využitím počítačového videnia, navrhnuť nové výzvy na obohatenie kategórie Dron súťaže Robocup Junior Slovensko a aktívne sa zúčastniť hodnotenia a následného znázornenia vzorového riešenia letu drona v novej rozšírenej kategórii.

Prv bola potrebná analýza architektúry drona Tello a nájdenie prieniku možností jeho kombinovania s počítačovým videním. Po vyhodnotení možností navrhnutá knižnica disponuje viacerými zaujímavými funkcionalitami pre získavanie potrebných informácií či už pre pohyb drona alebo analýzou obrazu. Implementovaním knižnice je riešiteľ schopný vyvinúť algoritmus napríklad riešenia časti 2.1 kapitoly Návrh, docieľiť pohyb drona detekciou tváre, alebo dekodovaním QR kódu v obraze.

Významným bodom práce bolo taktiež vyvinutie aplikácie pre rozhodcov. Aplikácia počas vývoja prešla viacerými verziami. Na počiatku slúžila aplikácia výhradne pre zápis a export hodnotení pre novú súťažnú kategóriu dron. Finálna verzia aplikácia umožňuje rozhodcom ktorejkoľvek z disciplín zapisovať potrebné dáta. Aplikácia je schopná spracovať dva druhy súborov. Prvý súbor vo formáte CSV disponuje informáciami o zápise tímov na disciplíny súťaže. V druhom súbore, ktorý má formát JSON, sa nachádzajú konfigurácie disciplín, podľa ktorých aplikácia vytvorí stavebné bloky používateľského rozhrania konkrétneho hodnotenia.

Výsledný produkt práce je balík, ktorý poskytuje ako súťažiacim, tak aj budúcim projektom základ pre rozvoj v tejto doméne. Je viacero zaujímavých možností, akými by sa knižnica dala do budúcna rozširovať. Jedným z nich je napríklad pridanie detekcie rozličných bitmáp v obraze. Pre tento účel existujú algoritmy slúžiace na zadefinovanie významných oblastí (ang. keypoints) [24] žiadaného objektu, ktoré sú následne vyhľadávané v zadanom obraze. Zhody významných oblastí sa počítajú, a podľa tohto zráťania vieme vyhodnocovať výskyt objektu v obraze. Pridaním takejto funkcionality by dron vedel reagovať nie len na geometrické útvary rôznych farieb, ale aj na ľubovoľný objekt ktorého významné oblasti máme vopred analyzované. Ďalším možným rozšírením knižnice by mohlo byť využitie sledovacích algoritmov [17]. Aj po odovzdaní práce máme v pláne na rozširovaní možných funkcionalít produktu pokračovať.

6. Literatúra

- [1] Všeobecné informácie o súťaži Robocup Junior, dostupné na : <https://junior.robocup.org/> [cit. 05.01.2023]
- [2] Všeobecné informácie o súťaži Robocup Junior Slovensko, dostupné na : <https://robotika.sk/rcj/> [cit. 17.12.2022]
- [3] Všeobecné informácie o kategórii dron ročníkov 2020-2022 súťaže Robocup Junior Slovensko, dostupné na : [https://wiki.robotika.sk/robowiki/index.php?title=RoboCup_Junior - Dron 2020-2022](https://wiki.robotika.sk/robowiki/index.php?title=RoboCup_Junior_-_Dron_2020-2022) [cit. 14.01.2023]
- [4] Prehľad o súťaži MathWorks Minidrone Competition, dostupné na : <https://www.mathworks.com/videos/series/mathworks-minidrone-competition.html> [cit. 30.04.2023]
- [5] Murtaza's Workshop - Robotics and AI , Drone Programming With Python Course | 3 Hours | Including x4 Projects | Computer Vision, 01.01.2021, Youtube, dostupné na : https://www.youtube.com/watch?v=LmEcyQnfpDA&t=8763s&ab_channel=Murtaza%27sWorkshop-RoboticsandAI [cit. 19.02.2023]
- [6] Mishra, B., Garg, D., Narang, P., & Mishra, V. (2020). Drone-surveillance for search and rescue in natural disaster. *Computer Communications*, 156, 1–10.
- [7] Tokyo Institute of Technology, “Listening” drone helps find victims needing rescue in disasters, zverejnené : December 22, 2017, dostupné na : <https://www.titech.ac.jp/english/news/2017/040159> [cit. 17.03.2023]
- [8] Acecore, Acecore Technologies announces speaker drone with vehicle takeoff, zverejnené : 15 April, 2020, dostupné na : <https://acecoretechnologies.com/news-speaker-drone/> [cit. 18.03.2023]
- [9] Ryze Technology, Tello User Manual v1.2, dostupné na : https://dl-cdn.ryzerobotics.com/downloads/Tello/20180404/Tello_User_Manual_V1.2_EN.pdf [cit. 22.05.2023]
- [10] RNDr. Elena Šikudová, PhD., Počítačové videnie – Computer Vision, dostupné na : <https://dai.fmph.uniba.sk/~vittek/proseminar/microsites/pgv/Elena%20%C5%A0ikudov%C3%A1%202011/videnie.html> [cit. 12.05.2023]

- [11] E. Šikudová, Z. Černeková, W. Benešová, Z. Haladová, and J. Kučerová, Počítačové videnie Detekcia a rozpoznávanie objektov. Wikina Praha, 2013, p. 397.
- [12] Öztürk, Ş., & Akdemir, B. (2015). Comparison of Edge Detection Algorithms for Texture Analysis on Glass Production. Procedia - Social and Behavioral Sciences, 195, 2675–2682.
- [13] Damià Fuentes Escoté and Jakob Löw, GitHub repozitár knižnice DjiTelloPy dostupné na : <https://github.com/damiafuentes/DJITelloPy> [cit. 20.02.2023]
- [14] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.
- [15] Gemius, 01/09/2022 - 30/04/2023, dostupné na : <https://ranking.gemius.com/sk/ranking/systems/> [cit. 24.05.2023]
- [16] skopecant, Tello mirror clip, 21/06/2020, dostupné na : <https://www.thingiverse.com/thing:4488454> [cit. 23.02.2023]
- [17] GAZAR, Ehsan, Object Tracking with OpenCV, 06/08/2018, dostupné na : <https://ehsangazar.com/object-tracking-with-opencv-fd18ccdd7369> [cit. 02.05.2023]
- [18] Robocup Junior, prevzaté z : <https://junior.robocup.org/rescue/> [cit. 20.05.2023]
- [19] Robocup Junior Australia, prevzaté z: <https://www.robocupjunior.org.au/soccer/> [cit. 20.05.2023]
- [20] Robocup Junior Australia, prevzaté z : <https://www.robocupjunior.org.au/past-performance-videos/> [cit. 20.05.2023]
- [21] Vizualizácia mapy starej kategórie Dron, prevzaté z : https://wiki.robotika.sk/robowiki/index.php?title=RoboCup_Junior_-_Dron_2020-2022 [cit. 20.05.2023]
- [22] Murtaza Hassan, kód riešenia problému sledovania čiary, dostupné po registrácii na : <https://www.computervision.zone/topic/line-follower/> [cit. 19.02.2023]
- [23] Obrázok prevzatý z : <https://tello.oneoffcoder.com/parts.html> [cit. 05.05.2023]
- [24] Mrinal Walia, Object Detection vs. Image Classification vs. Keypoint Detection, SEP 28, 2022, dostupné na : <https://blog.roboflow.com/object-detection-vs-image-classification-vs-keypoint-detection/> [cit. 21.05.2023]

7. Prílohy

Zdrojový kód vyvinutej knižnice, inštalačný súbor Android aplikácie, návod k používaniu tejto aplikácie a testovacie súbory k nej sú dostupné vo forme elektronickej prílohy v systéme AIS2. Zdrojový kód celej práce, spolu s návodmi na inštaláciu a použitie, sa nachádza v GitHub repozitári dostupný na : <https://github.com/PetoVerci/Tellovision-ReffApp> .