# Úvod

Technológia autonómnych dronov sa v posledných rokoch výrazne rozvíja a nachádza uplatnenie v mnohých odvetviach, vrátane logistiky, monitoringu, poľnohospodárstva a záchranných operácií. Schopnosť dronu pohybovať sa samostatne bez priamej intervencie človeka otvára nové možnosti pre automatizáciu úloh a zlepšovanie efektivity rôznych procesov.

V rámci maturitnej práce sa zameriavam na vývoj autonómneho dronu, ktorý pomocou počítačového videnia dokáže sledovať čiernu čiaru na zemi, rozpoznať a reagovať na rôzne farebné útvary, a vykonať definované úlohy vrátane prepravy balíka.

Cieľom mojej maturitnej práce je navrhnúť a implementovať program pre dron, ktorý bude schopný splniť tieto komplexné úlohy a následne bezpečne pristáť na určenom mieste. Realizácia tohto projektu si vyžaduje kombináciu rôznych technológií a znalostí, vrátane programovania v jazyku Python, použitia knižnice OpenCV na spracovanie obrazu, a 3D tlače na výrobu potrebných súčiastok.

Motiváciou pre túto tému je preukázanie schopnosti spojiť moderné technológie s praktickým riešením reálnych problémov. S narastajúcim dopytom po autonómnych systémoch je dôležité pochopiť, ako fungujú a ako ich možno implementovať tak, aby vykonávali presné úlohy, či už na pomoc v priemysle, alebo v iných oblastiach spoločnosti.

# TEORETICKÁ ČASŤ

# 1 Použité technológie

## 1.1.Hardware

### 1.1.1. Dron DJI Tello

Dron, ktorý je možné programovať v programovacom jazyku Python a disponuje senzormi a kamerou. Je vybavený aj integrovaným Wi-Fi modulom, ktorý umožňuje pripojenie k externým zariadeniam na diaľkové ovládanie alebo sledovanie jeho stavu.

## 1.2.Software

### 1.2.1. Programovací jazyk Python

Python je interpretovaný programovací jazyk, ktorý je taktiež známy svojou jednoduchosťou. Je populárny v oblasti počítačového videnia, strojového učene a robotiky. Je ideálny aj pre vývoj aplikácií a softvéru na autonómne riadenie dronov. Vďaka jeho flexibilite a rozsiahlej komunite vývojárov je Python voľbou pre rýchly vývoj a implementáciu nových funkcií v systémoch pre autonómne zariadenia.

### 1.2.2. Knižnica OpenCV

Knižnica OpenCV poskytuje širokú škálu nástrojov a funkcií na spracovanie obrazu a počítacové videnie. V práci je táto knižnica primárne využitá na detekciu farieb a útvarov v obraze, čo je kľúčové pre autonómne riadenie dronu.

### 1.2.3. Knižnica Tkinter

Tkinter je zabudovaná knižnica pre Python na tvorbu jednoduchého grafického rozhrania. Knižnica je použitá pre jednoduchú interakciu užívateľa s programom a poskytuje informácie o stave dronu.

## 1.3. Iné technológie

### 1.3.1. 3D tlač

3D tlač je využitá na vytvorenie mechanizmu pre uchopenie balíka, ktorý dron musí prepraviť na koniec svojej dráhy. 3D tlač nám umožňuje rýchly vývoj prototypov a presné

vytvorenie dielov na mieru. 3D tlač nám teda zabezpečí bezpečné a efektívne uchytenie a manipuláciu s balíkom počas letu.

# PRAKTICKÁ ČASŤ

# 2 Počítačové videnie

Počítačové videnie je jedna z najdôležitejších častí programu. Umožňuje dronu analyzovať obrazové dáta a na základe nich vykonávať autonómne rozhodnutia. Táto technológia simuluje schopnosť ľudského oka a mozgu identifikovať, rozpoznávať a interpretovať vizuálne informácie, čo je dôležité pre presnú navigáciu a plnenie úloh drona.

## 1.4.Rozpoznávanie farieb

V tejto časti je opísaná implementácia triedy ColorRecognition, ktorá je zodpovedná za rozpoznávanie farieb z obrazu. Obraz je pôvodne zachytávaný v BGR farebnom formáte, ale pre lepšie spracovanie sa obraz konvertuje do HSV farebného formátu.

### 1.4.1. Inicializácia triedy

Konštruktor triedy má zadefinovaný jeden povinný parameter, cestu JSON súboru, v ktorom sú uložené jednotlivé názvy a rozsahy farieb. V rámci inicializácie triedy sa pomocou metódy load_color_bounds načítajú názvy a rozsahy farieb.

JSON súbor obsahuje pole objektov, respektíve pole farieb. Každá farba obsahuje názov farby, pole troch hodnôt, ktoré predstavujú dolnú hranicu rozsahu farby v HSV farebnom formáte a druhé pole troch hodnôt ktoré predstavujú hornú hranicu rozsahu farby v HSV farebnom priestore.

### 1.4.2. Rozpoznávanie jednotlivých farieb

Metóda detect_colors je zodpovedná za identifikáciu farieb v danom obraze. Vstupný parameter tejto metódy je snímka vo formáte N-dimenzionálneho poľa, z ktorej sa najskôr spraví duplikát aby sa zachovala pôvodná snímka. Následne sa konvertuje z BGR farebného formátu do HSV farebného formátu. HSV formát sa používa, pretože lepšie reprezentuje farebné odtiene, sýtosť a jas. Tento formát nám teda umožňuje lepšie a presnejšie rozpoznať jednotlivé farby.

Metóda následne prechádza cez zoznam preddefinovaných rozsahov a názvov farieb. Pre každý rozsah farby sa vytvorí maska pomocou metódy apply_mask. Táto maska nám identifikuje a znázorni jednotlivé pixely v obraze, ktoré zodpovedajú danej farbe, respektíve sa nachádza v danom rozsahu farby. Po aplikovaní masky využijeme funkciu findContours z knižnice OpenCV pre vyhľadanie kontúr vo výslednej maske, čím sa identifikujú časti obrazu, ktoré obsahujú rozpoznanú farbu.

Každá nájdená kontúra, ktorej plocha je väčšia ako nami zadefinovaná hodnota 500 pixelov, je pre nás dôležitá, a tak jej obvod sa zvýrazní pomocou funkcie boundingRect z knižnice OpenCV. Táto funkcia nám umožní získať súradnice a rozmery obdĺžnika, ktorý znázorňuje polohu danej oblasti farby. Informácie o rozpoznanej farbe a jej pozícií v obraze sa uložia do zoznamu detected_colors vo forme slovníka s názvom farby a súradnicami obdĺžnika. Zoznam je na konci metódy vrátený ako výstup metódy detect_colors.

## 1.5.Rozpoznávanie útvarov

V tejto časti je podrobne opísaná implementácia triedy ShapeRecognition, ktorá je zodpovedná za rozpoznávanie útvarov v obraze. Trieda poskytuje metódy na spracovanie obrazu, identifikáciu útvarov a určenie typu tvaru na základe počtu vrcholov.

### 1.5.1. Detekcia útvarov

Metóda detect_shapes nám slúži na detekciu útvarov v obraze. Jej vstupný prameter je snímka obrazu vo formáte N-dimenzionálneho poľa, na ktorom chceme útvary detekovať. Najskôr si spravíme kópiu snímky, aby sme zachovali pôvodnú snímku. Ďalší krok, ktorý metóda vykoná je konvertovanie snímky do odtieňov sivej pre jednoduchšie spracovanie. Konvertovanie snímky do odtieňov sivej vieme spraviť pomocou funkcie cvtColor z knižnice OpenCV. Následne musíme snímku rozmazať pomocou funkcie GaussianBlur z knižnice OpenCV, aby sme zjemnili detaily a znížili šum, ktorý sa vyskytne pri zachytávaní obrazu kamerou.

Hlavný bod metódy je detekcia hrán. Detekciu hrán vieme docieliť pomocou funkcie Canny z knižnice OpenCV. Následne pomocou funkcie findContours z knižnice OpenCV vyhľadáme kontúry v obraze. Pre každú nájdenu kontúru sa zavolá nami ďalšia vytvorená funkcia identify_shape, ktorá určí typ úvaru. Ak je útvar identifikovaný, nakreslí sa obrys

na pôvodný obraz pomocou funkcie rectangle z knižnice OpenCV a nad tvarom sa zobrazí aj jeho názov.

### 1.5.2. Identifikácia útvarov

Metóda identify_shapes, ktorú používa metóda detect_shapes, slúži na určenie typu útvaru na základe počtu vrcholov kontúry a jej následnej aproximácii. Pomocou arcLength z knižnice OpenCV sa vypočíta obvod kontúry. Funkcia approxPolyDP z knižnice OpenCV aplikuje aproximáciu kontúry na základe daného zlomku obvodu (0.04 * peri), čím určí približný počet vrcholov.

Nakoniec metóda určí podľa vrcholov daný útvar. Pre tri vrcholy nám metóda vráti trojuholník, pre štyri nám metóda vráti štvorec alebo obdĺžnik a pre viac než štyri vrcholy nám vráti kruh. V prípade že metóda deteguje štyri vrcholy, tak metóda vypočíta pomer strán obdĺžnika a na základe neho určí či ide o štvorec alebo obdĺžnik. Metóda výstupnú hodnotu vracia ako reťazec, ktorý sa potom použije v metóde detect_shapes na označenie a zobrazenie útvaru v obraze.

# 3 Sledovanie čiary

Sledovanie čiary je ďalším kľúčovým aspektom. Táto funkcia umožní dronu detegovať a sledovať preddefinovanú líniu na zemi, ktorá naviguje po určenej trase.