

Универзитет у Београду  
Електротехнички факултет



Пројекат из предмета Теорија електричних кола  
**СИНХРОНА КОНТРОЛА ДРОНА И РАЧУНАРА  
ПОКРЕТИМА РУКЕ**

Ментор:

Поф. Др Дејан Тошић

Студент:

Владан Башић 2022/0395

У Београду  
септембар 2024.

# САДРЖАЈ

<b>1</b>	<b>Увод</b>	<b>2</b>
1.1	Инспирација за пројекат . . . . .	2
1.2	Tello - Паметни дрон . . . . .	2
1.3	Карактеристике дрона . . . . .	2
<b>2</b>	<b>Контролисање дрона помоћу покрета шаке</b>	<b>3</b>
2.1	КОРАК 1: Детектовање шаке . . . . .	3
2.2	КОРАК 2: Препознавање положаја . . . . .	4
2.3	КОРАК 3: Померање дрона . . . . .	4
2.4	ДОДАТАН КОРАК: Дисплеј . . . . .	4
<b>3</b>	<b>Алати</b>	<b>5</b>
3.1	Mediarpipe . . . . .	5
3.2	OpenCV . . . . .	5
3.3	DJITelloPy . . . . .	6
3.4	Pygame . . . . .	6
<b>4</b>	<b>Метод рада</b>	<b>6</b>
4.1	Одређивање позе шаке . . . . .	6
4.1.1	Напред-назад и лево-десно . . . . .	6
4.1.2	Горе-доле . . . . .	6
4.2	Дисплеј . . . . .	7
4.3	Контролисање дрона . . . . .	7
<b>5</b>	<b>Закључак</b>	<b>7</b>
<b>6</b>	<b>Даљи рад</b>	<b>8</b>
<b>7</b>	<b>Литература</b>	<b>8</b>

# 1 Увод

## 1.1 Инспирација за пројекат

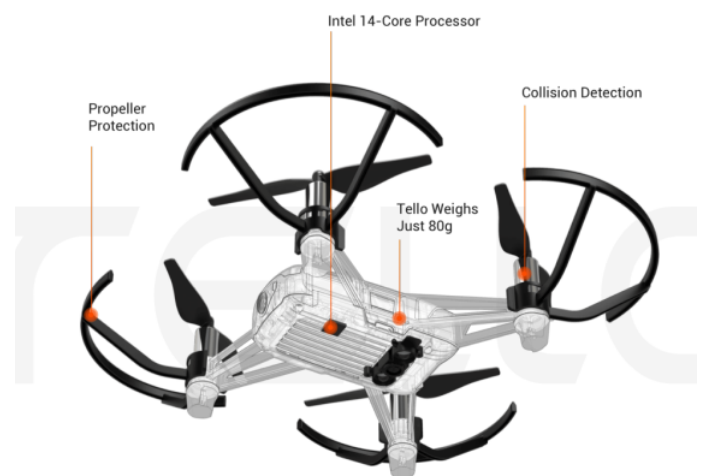
У данашње доба сталних иновација велик део фокуса технолошког напретка чине вештачка интелигенција и обрада слике. Ове области науке као и индустрија видео игара и њихових контролера биле су главне мотивације за овај пројекат. Покушавајући да спојим ове области у један пројекат добио сам идеју за овај пројекат.

## 1.2 Tello - Паметни дрон

Модел који је изабран за овај пројекат је **Tello TLW004** дрон који поседује електронску стабилизацију и Wi-Fi конекцију преко које може да прима команде и шаље податке.

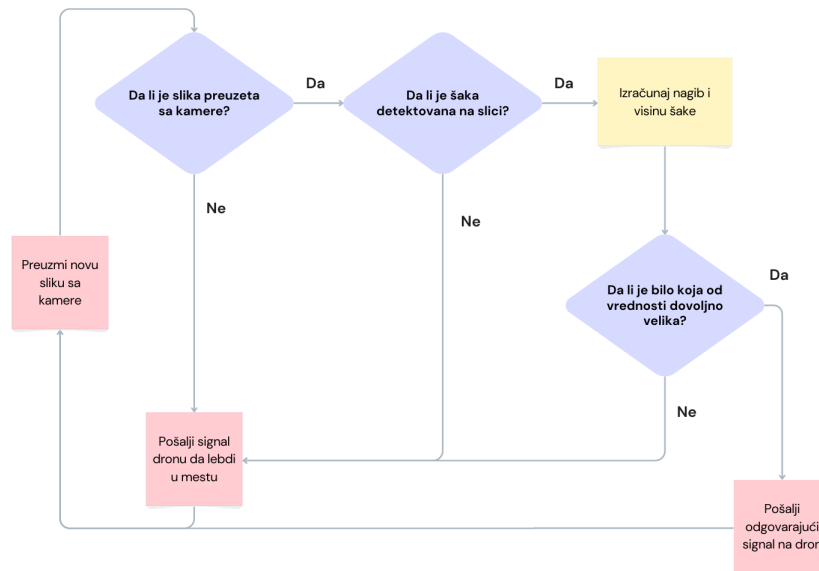
## 1.3 Карактеристике дрон-а

- Тежина: приближно 80g (са батеријом и пропелерима)
- Скидајућа батерија: 1.1Ah/3.8V
- Видео: HD720P@30
- Максимална даљина летења: 100m
- Максимална брзина: 8m/s
- Максимално време летења: 13min.
- Максимална висина летења: 10m
- Конекција: 2.4 GHz 802.11n Wi-Fi



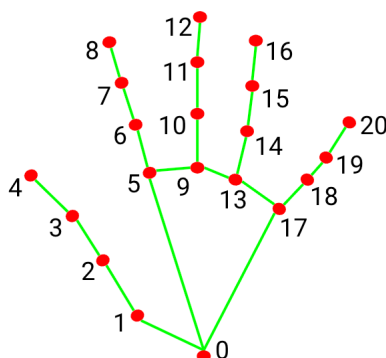
## 2 Контролисање дрона помоћу покрета шаке

У овом одељку биће изведен план и идеја пројекта. Идеја пројекта је да се померањем шаке која стоји у ваздуху изнад камере контролише дрон. Додељени покрети за контролу дрона су нагињање шаке напред и назад, лево и десно и подизање шаке на горе и спуштање шаке на доле. Препознавање ових покрета и комуникација са дроном су два главна дела овог пројекта.



### 2.1 КОРАК 1: Детектовање шаке

Потребно је детектовати шаку на слици и кодирати информације о њој тако да је могуће одредити нагнутост и удаљеност од камере. Најбољи начин да се то постигне је да се детектују шаке заједно са кључним тачкама шаке, које су представљене координатама. Кључне тачке шаке су тачке од интереса, начешће су то зглобови шаке.



- |                       |                       |
|-----------------------|-----------------------|
| 0. WRIST              | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC          | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP          | 13. RING_FINGER_MCP   |
| 3. THUMB_IP           | 14. RING_FINGER_PIP   |
| 4. THUMB_TIP          | 15. RING_FINGER_DIP   |
| 5. INDEX_FINGER_MCP   | 16. RING_FINGER_TIP   |
| 6. INDEX_FINGER_PIP   | 17. PINKY_MCP         |
| 7. INDEX_FINGER_DIP   | 18. PINKY_PIP         |
| 8. INDEX_FINGER_TIP   | 19. PINKY_DIP         |
| 9. MIDDLE_FINGER_MCP  | 20. PINKY_TIP         |
| 10. MIDDLE_FINGER_PIP |                       |

## 2.2 КОРАК 2: Препознавање положаја

Након што је шака препозната заједно са својим кључним тачкама потребно је да се, помоћу координата кључних тачака, одреди положај шаке. Одређују се три бредности: нагнутост напред-назад, нагнутост лево-десно и висина шаке.



## 2.3 КОРАК 3: Померање дрона

Након што је одређен положај шаке потребно је померити дрон. Дрон може да формира Wi-Fi конекцију са рачунаром преко које могу да размењују сигнале. Множењем сигнала позиције шаке одговарајућим скаларима добија се сигнал за померање дрона. Да би се избегло нестабилно константно померање при малим покретима шаке, јер шака не може бити саршено мирна, слаби сигнали се анулирају. Слаби сигнали се анулирају тако што се одређује праг активације. Ако дати сигнал има вредност која је мања од прага активације он се третира као нула сигнал. Ако сигнал има вредност која је већа од прага активације он се третира као разлика тог сигнала и прага активације.

$$f(x) = \begin{cases} 0, & x \leq \alpha \\ x - \alpha, & x \geq \alpha \end{cases}$$

## 2.4 ДОДАТАН КОРАК: Дисплеј

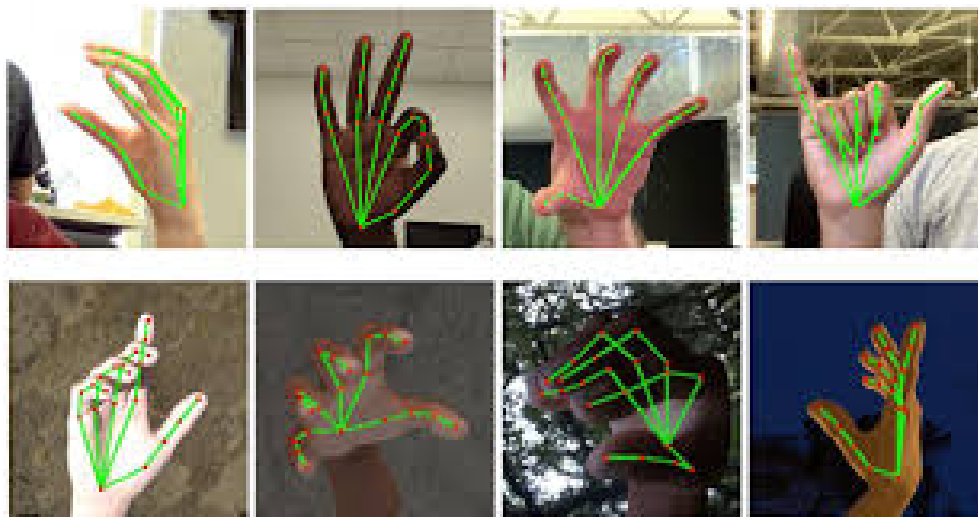
Иако није неопходан, дисплеј је јео користан додаток на овај систем. Имплементацијом дисплеја који на визуелан начин приказује сигнале који су послати дрону је јео корисна при тестирању и коришћењу система.

## 3 Алати

### 3.1 Mediapipe

Рачунарска визија је област програмирања која се бави добијањем информација из фотографије или видео снимака. Информације које могу бити добијене могу да буду од позиција на слици где се налази одређена боја до издвајања региона слике где се налази човек или неки други предмет. Рачунари су добри у решавању првог проблема како се на слици може јасно дефинисати тражена боја. Други проблем је скоро немогуће описати јер постоји превише варијабли. У таквим проблемима користе се статистичке методе које припадају области машинског учења. Циљ је да се скупи довољно података тако да мрежа може да "научи" да препозна задани објект. Овај поступак никада не даје савршене резултате али уз довољно велик сет података за тренирање и одговарајућ модел може да постигне довољно добре резултате.

Mediapipe је скуп више различитих модела машинског учења. Модел који се користи у овом раду је детектор кључних тачака шаке, који детектује све шаке на слици и дели их на 21 кључну тачку. Свака тачка означена је са три броја: x-координатом, y-координатом и z-координатом, где z-координата представља апроксимирану удаљеност по висини(дубини) од тачке 0 која обележава зглоб шаке.



### 3.2 OpenCV

OpenCV је широко коришћена Python библиотека за дигиталну обраду слике. Садржи функције за: учитавање и уписивање слика, мењање система боја, трансформацију слике, коришћење кернела, подршку снимања преко камере... У овом пројекту је коришћена за преузимање слике са камере.

### 3.3 DJITelloPy

Djitellopy је python библиотека за интеракцију са DJI Tello дроном. Омогућава контролу кретање дрона и приступ телеметријским подацима у реалном времену, као што су ниво батерије, време лета и још много тога. Поред тога, омогућава примање видео стрима са дрона у реалном времену и његову обраду користећи OpenCV библиотеку. Библиотека је изграђена на врху Tello-вог SDK-а који омогућава контролу дрона на ниском нивоу. У овом пројекту коришћена је само за слање контрола дрону.

### 3.4 Pygame

Pygame је Python библиотека за формирање екрана и цртање на њему. Примарно је коришћена за прављење видео игрица и физичких симулација. Библиотека такође омогућава читање уноса са тастатуре и миша. У овом пројекту коришћена је за формирање дисплеја и детекцију клика дугмића на тастатури ради покретања команди као што су узлетање, слетање, калибрација...

## 4 Метод рада

### 4.1 Одређивање позе шаке

Након детектовања кључних тачака шаке потребно је да се одреди у којој се пози налази шака. Посматра се нагињање напред-назад, лево-десно и померање горе-доле. Помоћу координата кључних тачака које су од интереса одредиће се нагиб и висина. Све добијене вредности нормализоване су на распон између -1 и 1 ради лакшег даљег рада са њима.

#### 4.1.1 Напред-назад и лево-десно

За нагињање напред-назад посматран је угао са хоризонталом који формира права која пролази кроз тачке 0 и 9. Овај угао рачуна се као:

$$\Theta_{FB} = \arctg \frac{z_9 - z_0}{\sqrt{(x_9 - x_0)^2 + (y_9 - y_0)^2}}$$

Угао левп-десно одређује се на исти начин само са тачкама 5 и 17.

$$\Theta_{LR} = \arctg \frac{z_{17} - z_5}{\sqrt{(x_{17} - x_5)^2 + (y_{17} - y_5)^2}}$$

Притискањем дугмета ENTER калибришу се осе и њихова тренутна вредност се користи за добијање нових референтних вредности. Ова функционалност омогућава калибрисање система на различите камере са различитим карактеристикама и различита окружења са различитим условима.

#### 4.1.2 Горе-доле

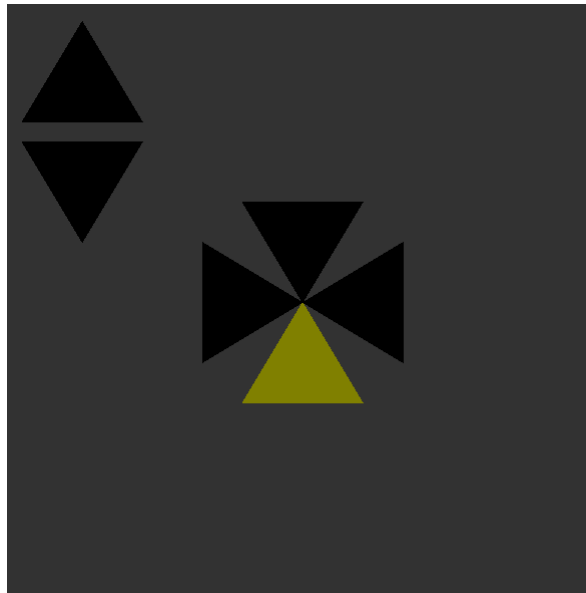
За разлику од углова нагиба, висину шаке није могуће одредити апсолутно због релативности z-координате (означава удаљеност тачке од зглоба а не од камере). Удаљеност шаке одређује се релативно у односу на дату тачку калибрације. Удаљеност

је представљена дистанцом између тачака 5 и 17. Када је на тастатури притиснуто дугме SPACE трентна удаљеност тачака је постављена као референтна. Када се шака приближава камери удаљеност расте и обрнуто. Да би се избегло лажно детектовање при ротирању шаке детекција промене висине шаке је дозвољена само ако није детектовано ротирање шаке, што значи да је могуће мењати висину дрона само ако је шака постављена хоризонтално.

$$d = \sqrt{(x_{17} - x_5)^2 + (y_{17} - y_5)^2 + (z_{17} - z_5)^2}$$

## 4.2 Дисплеј

Дисплеј детектованих команди се врши помоћу библиотеке *PyGame*. Ова библиотека омогућава формирање екрана и цртање по њему, као и детектовање притискања дугмића на тастатури. Формирани дисплеј сачињен је од 6 индикатора за смерове у којима се дрон може кретати. Када дрону није послат сигнал за кретање на дату страну одговарајући индикатор је црн. Када одговарајући сигнал постоји индикатор постаје жут. Већа сатурација жуте боје означава јачи сигнал.



## 4.3 Контролисање дрона

Контролисање дрона врши се преко *djitellopy* библиотеке. Помоћу ове библиотеке могуће је слати команде дрону преко рачунара који је са дроном повезан преко Wi-Fi конекције. Библиотека садржи функцију **send\_rc\_control** којом је могуће послати сигнал који oponaша сигнал са контролера (*joystick*). Ова функција прима 4 параметра који одређују колико дрон треба да се помера по осама (напред-назад, лево-десно, горе-доле и ротација око вертикалне осе). За сврхе овог пројекта није коришћена ротација око вертикалне осе и њен параметар се увек поставља на нула.

## 5 Закључак

Овај пројекат успешно је повезао технологију модерних контролера са обрадом слике, машинским учењем и контролом беспилотних летилица. Изложени су неки од



алата који су корисни за овакве типове пројеката у жељи да помогну читаоцима који са њима нису упознати и скрену пажњу на постојање поменутих алата.

## 6 Даљи рад

Главни проблем пројекта је кашњење између померања руке и реакције дрона на тај померај. Ово кашњење може се умањити даљом оптимизацијом кода, али најбољи начин да се кашњење умањи би био да се систем праћења руке замени. Уместо камере користила би се посебна рукавица са акцелерометром и микроконтролером. Микроконтролер би комуницирао са рачунаром који би комуницирао са дроном. Овај метод би представљао потпуно другачији пројекат са својим сетом изазова и проблема. На крају крајева, немогуће је потпуно уклонити кашњење зато што Wi-Fi конекција коју дрон користи за комуникацију је сама по себи спора и уводи значајно кашњење у систем.

## 7 Литература

- <https://github.com/google-ai-edge/mediapipe>
- [https://www.youtube.com/watch?v=NGQgRH2\\_kq8&t=547s](https://www.youtube.com/watch?v=NGQgRH2_kq8&t=547s)
- <https://www.youtube.com/watch?v=vD0kUHNdmKs&t=82s>
- <https://www.pygame.org/docs/>

Изворни код: **github.com**