

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ІВАСЕЧКО Андрій Вікторович

**Метод та засоби розпізнавання зображень для системи
ведення статистики гри в настільному тенісі / Image
recognition method and tools for the keeping statistics
system in the of table tennis**

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КНм-21
А.В. Івасечко

Науковий керівник:
к.т.н., доцент О.Р. Осолінський

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
Завідувач кафедри
_____ М.П. Комар

ТЕРНОПІЛЬ - 2023

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «магістр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
М.П. Комар
«___» _____ 20__р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Івасечкові Андрію Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Метод та засоби розпізнавання зображень для системи ведення статистики гри в настільному тенісі / Image recognition method and tools for the keeping statistics system in the of table tennis

керівник роботи к.т.н., доцент О.Р. Осолінський

затверджені наказом по університету від 8 грудня 2022 року № 491.

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити

- огляд предметної області;
- аналіз існуючих систем спортивної аналітики на базі технології комп'ютерного зору;
- аналіз найближчих досліджень на тему використання комп'ютерного зору в спорті
- постановка задачі дослідження;
- дослідження алгоритму відстеження об'єкта за допомогою пошуку кольорової плями;
- розробити алгоритм системи для відстеження тенісного м'яча та ведення рахунку в настільному тенісі;
- дослідження засобів реалізації запропонованого алгоритму;
- реалізація алгоритму системи для відстеження тенісного м'яча та ведення рахунку в настільному тенісі
- результати експериментальних досліджень та порівняння з відомими рішеннями.

5. Перелік графічного матеріалу у роботі
- демонстрація роботи програми;
 - вікна системи з результатами тестування;

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області та відомих рішень у сфері використання комп'ютерного зору в спорті	12.2022 р. – 03.2023 р.	
2	Метод відстеження об'єктів та виконання суддівських процесів в настільному тенісі	03.2023 р. – 05.2023 р.	
3	Реалізація та результати експериментальних досліджень алгоритму відстеження об'єктів та виконання суддівських процесів в настільному тенісі	05.2023 р. – 11.2023 р.	
4	Повне завершення та представлення кваліфікаційної роботи на кафедрі	01.12.2023 р.	

Студент _____ А.В. Івасечко
підпис

Керівник роботи _____ к.т.н., доцент Осолінський
підпис

РЕЗЮМЕ

Кваліфікаційна робота на тему «Метод та засоби розпізнавання зображень для системи ведення статистики гри в настільному тенісі» на здобуття освітнього ступеня «Магістр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 102 сторінки і містить 11 ілюстрації, 4 таблиць, 3 додатки та 46 використаних джерел.

Метою даної кваліфікаційної роботи є розробка програмного модуля "Електронний суддя для пінг-понгу", який використовує передові технології комп'ютерного зору та машинного навчання для автоматичного визначення результатів гри.

Методи досліджень: методи обробки зображень, комп'ютерний зір, математична статистика, теорія штучних нейронних мереж, методи комп'ютерного моделювання.

Результати дослідження: вдосконалено та адаптовано метод відслідковування м'яча у відеопотоці для настільного тенісу. Запропоновано концепцію системи електронного судді для настільного тенісу та алгоритм функціонування програмного забезпечення ведення рахунку.

Результати роботи можуть застосовуватися, як допоміжна система ведення рахунку в настільному тенісі для підвищення точності та рівня суддівства в змаганнях аматорського рівня.

Ключові слова: НАСТІЛЬНИЙ ТЕНІС, ПІНГ-ПОНГ, СИСТЕМА ЕЛЕКТРОННОГО СУДДІВСТВА, RASBERRY PI, PYTHON, КОМП'ЮТЕРНИЙ ЗІР.

ABSTRACT

Qualification work on the topic «Image recognition method and tools for the keeping statistics system in the of table tennis» for Master's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 102 pages and it contains 11 figures, 4 tablei, 3 annexes and 46 sources.

The purpose of this qualification work is to development of the "Electronic Ping-Pong Referee" software module, which uses advanced computer vision and machine learning technologies to automatically determine game results.

Research methods: image processing methods, computer vision, mathematical statistics, theory of artificial neural networks, computer modeling methods.

Research results: the method of tracking the ball in a video stream for table tennis was improved and adapted. The concept of an electronic referee system for table tennis and an algorithm for the software are proposed.

The results can be used as an auxiliary scoring system in table tennis to improve the accuracy and level of judging in amateur competitions.

Keywords: TABLE TENNIS, PING PONG, ELECTRONIC JUDGING SYSTEM, RASBERRY PI, PYTHON, COMPUTER VISION.

ЗМІСТ

Вступ	8
1 Аналіз предметної області та відомих рішень у сфері використання комп'ютерного зору в спорті	14
1.1 Поняття комп'ютерного зору	14
1.2 Можливості застосування комп'ютерного зору в спорті	15
1.3 Машинне навчання	17
1.4 Розпізнавання образів.....	18
1.5 Комп'ютерний зір у інших видах спорту	20
1.6 Комерційні системи на основі комп'ютерного зору	24
1.7 Наукові дослідження стосовно комп'ютерного зору в настільному тенісі	31
1.8 Постановка завдання	32
Висновки до розділу 1:	35
2 Метод відстеження об'єктів та виконання суддівських процесів в настільному тенісі	37
2.1 Концепція системи.....	37
2.2 Архітектура системи.....	39
2.3. Пропонований метод	41
Висновки до розділу 2:	50
3 Реалізація та результати експериментальних досліджень алгоритму Відстеження об'єктів та виконання суддівських процесів в настільному тенісі	52
3.1 Підбір технічного забезпечення	52
3.2 Підбір програмного забезпечення.....	54
3.3 Проектування інтерфейсу	54
3.4 Опис роботи системи.....	57
3.5 Тестування	60
3.6 Порівняння з аналогами	67

Висновки до розділу 3:.....	70
Висновки	72
Додаток А код програми відстеження об'єктів та виконання суддівських процесів в настільному тенісі	78
Додаток Б Апробація отриманих результатів	82

ВСТУП

Обґрунтування вибору теми та її актуальність для науки і практики:

Розвиток програмного забезпечення та технологій штучного інтелекту набуває все більшого значення в різних галузях життя. Спортивні ігри не є винятком, адже вони є важливою частиною в житті суспільства, необхідним аспектом фізичному розвитку та здорового способу життя. У цьому контексті розробка програмного модуля "Електронний суддя для пінг-понгу" має особливу актуальність.

Пінг-понг - це популярна спортивна гра, яка вимагає від гравця хорошої координації, швидкості та реакції. Однак, під час гри виникають ситуації, коли необхідно об'єктивно визначити правильність удару, враховувати правила гри та забезпечити чесність результатів. Традиційно це роблять судді, проте їх наявність не завжди можлива або ефективна. Тому розробка програмного модуля, який замінить суддю та забезпечить автоматичне визначення результатів, є важливим завданням.

Формулювання мети роботи і завдань, які необхідно вирішити для досягнення поставленої мети:

Метою данго дослідження є розробка програмного модуля "Електронний суддя для пінг-понгу", який використовує передові технології комп'ютерного зору та машинного навчання для автоматичного визначення результатів гри. Для досягнення цієї мети необхідно вирішити такі завдання:

- Розробити алгоритми комп'ютерного зору та машинного навчання для аналізу руху м'яча та ракеток гравців.
- Реалізувати програмний модуль, який здатний обробляти відеозапис гри та вести рахунок як під час матчу, так і загальний рахунок по раундах.

- Провести тестування та оцінку ефективності розробленого модуля на різних ігрових ситуаціях.

- Визначення об'єкта і предмета дослідження:

Об'єктом дослідження є процес розробки програмного модуля "Електронний суддя для пінг-понгу", який використовує комп'ютерний зір та машинне навчання для автоматичного визначення результатів гри.

Предметом дослідження є розробка алгоритмів обробки відеозапису гри, визначення параметрів руху м'яча, а також реалізація програмного модуля для ведення рахунку в пінг-понгу.

Методи дослідження:

- Аналіз наукової літератури та публікацій з області комп'ютерного зору, машинного навчання та спортивних ігор.

- Експериментальне дослідження на реальних відеозаписах пінг-понгу для аналізу руху м'яча та ракеток гравців.

- Розробка методу обробки відео та аналізу гри.

- Реалізація алгоритмів обробки відео та аналізу гри.

- Тестування та оцінка ефективності розробленого програмного модуля на різних ігрових ситуаціях.

Опис інформаційної бази наукового дослідження:

В ході роботи використовуватимуться наукові статті, публікації та інформація з області комп'ютерного зору, обробки відео та спортивних ігор. Також будуть використовуватись відеозаписи пінг-понгу для експериментального дослідження та аналізу руху м'яча та ракеток гравців.

Наукова новизна отриманих результатів полягає у розробці портативної системи, яка використовує стандартну відеокамеру і мікрокомп'ютер Raspberry Pi для відстеження м'яча, а також спеціально розроблений алгоритм для ведення рахунку і визначення правильності подачі та ударів. Ця система дозволяє не тільки забезпечити більш справедливий та об'єктивний контроль за ходом гри, але й знижує

залежність від людського фактора у суддівстві, забезпечуючи більш точні та швидкі рішення.

Практичне значення одержаних результатів:

Розробка програмного модуля "Електронний суддя для пінг-понгу" має велике практичне значення. Він може використовуватись у спортивних залах, турнірах та тренуваннях з пінг-понгу для автоматичного визначення правильності ударів та рахунку, що забезпечує більш об'єктивну гру та уникнення конфліктів між гравцями. Крім того, розроблений модуль може бути використаний як основа для подальшого розвитку та впровадження подібних систем у інші спортивні ігри, сприяючи автоматизації та покращенню суддівських процедур.

Апробація результатів дослідження.

Основні теоретичні положення та практичні результати даного дослідження обговорювалися на IV Міжнародній студентській науковій конференції 01.12.23 - Луцьк, Україна, та ВОТТП – «Вимірювальна та обчислювальна техніка в технологічних процесах» №3 Хмельницький, Україна

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВІДОМИХ РІШЕНЬ У СФЕРІ ВИКОРИСТАННЯ КОМП'ЮТЕРНОГО ЗОРУ В СПОРТІ

1.1 Поняття комп'ютерного зору

Комп'ютерний зір, або ж Computer Vision - це науковий напрям, який включає в себе дослідження і розробку алгоритмів та методів для розуміння та аналізу зображень та відео, що виконуються комп'ютерами. Він займається створенням систем та програм, які можуть "бачити" та "розуміти" світ навколо нас, аналізувати зображення, розпізнавати об'єкти, рух, розуміти контекст та приймати рішення на основі отриманих даних.

Область застосування даної технології є дуже широкою і розмаїтою. Вона охоплює такі сфери, як автономні транспортні засоби, безпека та відеоспостереження, медицина, розпізнавання облич, виробничі процеси, віртуальна реальність та багато інших, до яких належить спорт в тому числі належить. Використання технології комп'ютерного зору якого дає можливість розширити межі людських можливостей, полегшити рутинну роботу, забезпечити безпеку та покращити якість життя.

Одним з ключових елементів комп'ютерного зору є обробка зображень. Завдяки алгоритмам обробки зображень ми можемо виконувати такі операції, як фільтрація, розмиття, збільшення чи зменшення розміру, зберігання та відтворення зображень. Також важливим аспектом є виявлення об'єктів та розпізнавання образів, яке може включати в себе визначення контуру, виявлення різних особливостей об'єктів, класифікацію та ідентифікацію.

Одним з найважливіших завдань в комп'ютерному зорі є розпізнавання об'єктів. Це процес, який дозволяє ідентифікувати та класифікувати об'єкти на зображеннях або відео. Він використовується в різних сферах, наприклад, у медицині для розпізнавання патологій на медичних зображеннях, в автомобільній промисловості для розпізнавання дорожніх знаків або в робототехніці для навігації та взаємодії з навколишнім середовищем.

Для досягнення поставлених завдань у галузі комп'ютерного зору використовуються різні методи та техніки. Одним з них є машинне навчання, яке дозволяє комп'ютеру "навчитися" розпізнавати об'єкти шляхом аналізу великої кількості даних. Інші методи включають в себе фільтрацію зображень, перетворення зображень, статистичний аналіз та багато інших.

У даній роботі буде розроблено програмний модуль "Розумний суддя для пінг-понгу" на основі технології комп'ютерного зору. Мета роботи полягає в створенні системи, яка здатна аналізувати відеозаписи пінг-понгу та автоматично визначати результати гри на основі виявлення руху м'яча та розпізнавання правил гри. Це може бути корисним для спортивних тренерів та арбітрів, які зможуть отримувати об'єктивну інформацію про результати гри без піддачі впливу суб'єктивного фактору.

Отримані результати даної роботи матимуть практичне значення для спортивної галузі, забезпечуючи більш точну та об'єктивну оцінку результатів гри в пінг-понг на аматорських турнірах та тренуваннях. Крім того, цей проект дозволить дослідити можливості комп'ютерного зору в спортивних застосуваннях та розширити границі його застосування[1].

1.2 Можливості застосування комп'ютерного зору в спорті

Комп'ютерний зір, або computer vision, є однією з найбільш важливих галузей комп'ютерних наук, що займається обробкою та аналізом зображень і відео з метою розпізнавання об'єктів, вимірювання властивостей та виявлення різних шаблонів та патернів. В спорті, комп'ютерний зір використовується для розробки систем, які можуть аналізувати та інтерпретувати відеозаписи спортивних подій, що дозволяє спортсменам, тренерам та суддям отримати об'єктивну інформацію про хід гри та виявити помилки або потенційні можливості для покращення[2].

Одним з основних застосувань комп'ютерного зору в спорті є визначення позиції та руху об'єктів на полі гри. Наприклад, в футболі комп'ютерний зір

може бути використаний для визначення офсайду, фіксації м'яча та руху гравців, а також виявлення порушень правил, таких як рука чи офсайд. За допомогою високошвидкісних камер та алгоритмів комп'ютерного зору, система може автоматично визначати ці параметри та повідомляти суддю або тренера про потенційні порушення. Це поліпшує точність суддівських рішень та допомагає уникнути суперечок та конфліктів.

Крім того, комп'ютерний зір може бути використаний для аналізу та оцінки техніки виконання рухів у різних видах спорту. Наприклад, в гімнастиці або фігурному катанні на коньках система комп'ютерного зору може виявляти позицію тіла спортсмена, кут нахилу та рухи кінцівок з високою точністю. Це дозволяє тренерам та спортсменам аналізувати техніку виконання рухів, виявляти недоліки та працювати над їх вдосконаленням. Такі системи можуть надати об'єктивну оцінку технічних елементів та забезпечити обратній зв'язок для покращення виконання спортивних рухів.

Також, комп'ютерний зір може бути використаний для створення віртуальних тренувальних середовищ, де спортсмени можуть симулювати різні ситуації та тренувати свої навички. Наприклад, в гольфі або тенісі система комп'ютерного зору може моделювати різні поля, погодні умови та рухи м'яча для тренування точності та стратегії гри. Це дозволяє спортсменам тренуватися в реалістичних умовах та реагувати на різні сценарії гри.

Загалом, використання комп'ютерного зору в спорті має великий потенціал для покращення тренувань, аналізу гри та досягнення кращих результатів. Він допомагає спортсменам удосконалювати свої навички, тренерам - ефективніше працювати з командою, а глядачам - насолоджуватися більш захоплюючими та емоційними спортивними змаганнями. За допомогою комп'ютерного зору в спорті, можна зробити спортивні події більш об'єктивними, захоплюючими та доступними для аналізу та дослідження[3].

1.3 Машинне навчання

Машинне навчання, також відоме як machine learning - це галузь, що займається розробкою та розумінням методів, які дозволяють комп'ютерам "навчатися" і покращувати свою продуктивність у виконанні різних завдань, використовуючи дані. Алгоритми машинного навчання використовують навчальні дані для побудови моделей, які можуть здійснювати прогнози та приймати рішення без явного програмування. Ці алгоритми широко застосовуються у багатьох галузях, таких як медицина, фільтрація електронної пошти, розпізнавання мови, сільське господарство та комп'ютерний зір. Машинне навчання пов'язане з обчислювальною статистикою і використовує методи математичної оптимізації. Воно також може використовувати дані та нейронні мережі для моделювання роботи біологічного мозку. У бізнес-середовищі машинне навчання також відоме як прогнозна аналітика[4].

Алгоритми навчання ґрунтуються на припущенні, що стратегії, алгоритми та висновки, що показали гарні результати в минулому, імовірно, продовжуватимуть працювати ефективно у майбутньому. Це може бути очевидним, як у прикладі з сходженням сонця, або більш складним, коли розглядаються можливі нюанси та варіації. Програми машинного навчання здатні виконувати завдання без прямого програмування. Вони вчаться на основі наданих даних з метою вирішення конкретних завдань. Для простих завдань можна створити алгоритми, які крок за кроком вказують комп'ютеру, як вирішити проблему, і не потрібне додаткове навчання. Однак, для складніших завдань створення необхідних алгоритмів вручну може бути складним завданням. У таких випадках ефективніше допомогти комп'ютеру самостійно розробити свій алгоритм, замість того, щоб програмісти вказували кожен окремий крок.

Машинне навчання використовує різні підходи для навчання комп'ютерів виконувати завдання, для яких відсутні повністю задовільні алгоритми. У випадках, коли існує велика кількість можливих відповідей,

один підхід полягає в тому, щоб позначити деякі з правильних відповідей як вірні. Ці дані можуть бути використані для навчання комп'ютера та покращення алгоритмів, які визначають правильні відповіді. Наприклад, для навчання системи розпізнавання цифр часто використовуються набори даних, такі як OpenTTGames. У даному наборі даних для завдань комп'ютерного зору в настільному тенісі містяться п'ять навчальних відео та сім тестових відео. Кожне відео містить файли з розміткою координат м'яча, а також папку з масками сегментації. Всього набір даних містить 4271 ручно анотованих подій трьох класів: відскоки м'яча, удари по сітці та порожні події[5][6].

1.4 Розпізнавання образів

Як люди, ми маємо природну здатність розпізнавати закономірності та порівнювати їх зі збереженою пам'яттю. В контексті розпізнавання образів, це означає, що ми можемо запам'ятовувати та відновлювати шаблони, коли ми їх бачимо знову і знову. У сфері машинного навчання розпізнавання образів включає порівняння інформації з бази даних з вхідними даними. Іншими словами, моделі використовують те, що вони вивчили, щоб ефективно виявляти спільні особливості.

Незважаючи на те, що існують певні спільні елементи, такі як класифікація зображень для розпізнавання образів, комп'ютерний зір і розпізнавання образів в цілому мають свої відмінності. Розпізнавання образів займається обробкою різних типів даних і стосується автоматичного виявлення шаблонів, тоді як комп'ютерний зір фокусується на обробці зображень, виявленні об'єктів, класифікації зображень і сегментації, не використовуючи виключно розпізнавання шаблонів.

Як один із основних складових елементів комп'ютерного зору, розпізнавання образів має за мету емулювати здібності людського мозку. Можна розглядати його як здатність моделей передбачати невидимі дані шляхом ідентифікації повторюваних закономірностей. Це може бути

застосовано до будь-якого формату даних, таких як зображення, відео, текст, аудіо і т.д.

В основі складності розпізнавання образів лежить аналіз вхідних даних, виділення шаблонів і їх порівняння зі збереженими даними. Процес може бути розбитий на дві фази: експлораторну, де алгоритми досліджують шаблони, і описову, де алгоритми групують та описують знайдені закономірності в контексті вихідних даних[7].

Алгоритм розпізнавання образів складається з таких етапів:

1. Збір даних може бути значним кроком для досягнення високої точності розпізнавання. Використання високоякісних наборів даних є необхідним, і відкриті набори даних можуть значно скоротити час, який потрібно для їх збору. Проте, важливо приділяти увагу контролю якості даних. У випадках, коли ручний збір даних неможливий, можна створити синтетичні набори даних.

2. Попередня обробка даних включає видалення шуму та домішок для створення повніших наборів даних і підвищення точності прогнозів. Згладжування і нормалізація використовуються для виправлення варіацій у напрямку та інтенсивності освітлення. Цей крок допомагає створити значущі та легко інтерпретовані дані для моделей.

3. Вилучення ознак передбачає перетворення вхідних даних у вектор ознак, що представляє зменшену версію набору ознак. Це допомагає вирішити проблему високої розмірності вхідних даних шляхом виділення лише релевантної інформації. Важливо, щоб обрані функції були стійкими до спотворень або маніпуляцій. Після цього вибрані об'єкти піддаються класифікації.

4. На етапі ж класифікації вилучені ознаки порівнюються з відповідними шаблонами, що дозволяє призначити їм відповідні класи. У процедурі навчання класифікатори використовують попередні знання про категорії шаблонів для розрізнення різних класів. Неконтрольоване навчання використовує властивості даних для визначення параметрів моделі.

Класифікація шаблонів зазвичай слідує постобробці, де робляться рішення щодо використання результатів для керування системою.

Одним з найважливіших аспектів у розпізнаванні образів є вибір підходящих алгоритмів. Наступні шість загальних методів розпізнавання можуть бути використані в даному процесі:

1. Статистичний алгоритм: Використовує статистичні методи для навчання на основі прикладів та розробки правил для майбутніх спостережень.

2. Структурний алгоритм: Враховує складні взаємозв'язки між елементами та застосовує ієрархічний підхід для розпізнавання складних образів.

3. Нейронна мережа: Використовує штучні нейронні мережі, що дозволяють більш гнучке розпізнавання та класифікацію шляхом виявлення закономірностей.

4. Відповідність шаблону: Застосовується для порівняння двох сутностей одного типу та визначення їх подібності на основі збережених шаблонів.

5. Нечіткий алгоритм: Використовує нечітку логіку для розпізнавання образів, що відображають невизначеність та нечіткі компоненти.

6. Гібридний алгоритм: Комбінує різні методи розпізнавання, використовуючи кілька класифікаторів та простори ознак, для отримання найкращих результатів[8].

Ці методи можуть бути використані окремо або в поєднанні, залежно від конкретних потреб та характеру завдання розпізнавання образів.

1.5 Комп'ютерний зір у інших видах спорту

Потенціал технології комп'ютерного зору є значно більшим ніж його застосування лише в настільному тенісі. Було проведено значне дослідження щодо застосування CV для виявлення, відстеження та аналізу кидка м'яча в

різних видів спорту. Багато з цих алгоритмів залежать від контексту і вимагають урахування екологічних умов, якості відео або особливостей спортивної події. Наступні статті не мають прямого зв'язку з настільним тенісом, але їх варто розглянути з метою можливого впровадження в галузі настільного тенісу. Зворотно, результати обговорення CV у настільному тенісі можуть розширити потенціал застосування CV в інших видах спорту.

Був вивчений зв'язок між застосовністю алгоритмів у різних видах спорту, при цьому загальні характеристики, такі як кількість гравців, швидкість м'яча, розмір, колір, площа та розмір ігрового поля, залишаються актуальними. Порівняння комбінацій алгоритмів у різних видах спорту дозволяє зрозуміти складність проблемної області та надає ідеї про те, як інші дослідники розв'язували подібні складності[9].

1.5.1 Комп'ютерний зір в настільному тенісі

Аналіз відео в реальному часі та виявлення об'єктів інтересу є поширеним в світовому спорті. Проте в настільному тенісі, одному з найпопулярніших видів спорту, загалом відсутні дослідження та інвестиції в технології виявлення м'яча на основі комп'ютерного зору. Поза людською грою, комп'ютерний зір регулярно використовується в технічних завданнях, пов'язаних з навчанням роботів грати в настільний теніс, як запропонували Чжан та ін..Хоча основи технології обробки можуть бути використані повторно, вимоги та обмеження між проектуванням робота та безпечним виявленням об'єктів інтересу для аналізу результативності та підтримки рішень судді значно відрізняються.

Навіть на останніх Олімпійських іграх 2020-го року високошвидкісні камери використовувалися для сповільнення гри з метою задоволення глядачів, проте не було жодної спроби покращити цю технологію для надання даних гравцям, тренерам або суддям. Потенційні переваги, які це може принести для спорту, значні. Наприклад, молодому гравцеві, який тренується, щоб поліпшити швидкість, напрямок, точку контакту зі столом або законність

подачі. Додатковим застосуванням може бути можливість тренера спостерігати й розвивати кількох гравців одночасно, можливо, навіть дистанційно за допомогою онлайн-застосунків. Навіть під час клубного турніру, мати доступ до систем підтримки прийняття рішень, що пропонують неупереджені дані та аналіз, буде корисним. Вихідні дані на основі резюме з недорогих монокулярних відеореєстраторів, ретельно розташовані поруч з параграфом, дозволяють негайно отримати доступ до даних і мати достатню точність, щоб принести користь. Саме в цій області знаходиться фокус цього дослідження[10].

1.5.2 Комп'ютерний зір в баскетболі

Було запропоноване рішення для баскетболу, яке поєднує використання кольорової гістограми для виділення м'яча з використанням кругового перетворення Хафа для виявлення м'яча. Далі застосовується рекурсивне відстеження м'яча у тривимірному просторі за допомогою каліброваних камер. Процес відстеження включає пошук тривимірного простору навколо попередньо відомої позиції м'яча шляхом оцінки його двовимірних представлень, отриманих окремими камерами. Це рішення було протестоване на відео з роздільною здатністю 720x576 пікселів та частотою захоплення кадрів 25 Гц. Порівнюючи це рішення з застосуванням у настільному тенісі, можна сказати, що воно не стикалося з додатковими складнощами, які виникають у випадку швидкорухливого м'яча з частими змінами траєкторії, граючи на меншій площі та маючи менший діаметр. Крім того, висока вартість використання кількох камер ускладнює впровадження цього рішення. Однак, використані алгоритми комп'ютерного зору дають орієнтир для оцінки обробки зображення[10].

Альтернативний двовимірний підхід був запропонований Чакраборті та Мехр[4], в якому кожний кадр відео сегментується за допомогою алгоритму розрізнення кадрів для ідентифікації набору об'єктів-кандидатів. Потім проводиться фільтрація кандидатів, основана на їх розмірі відносно відстані

до камери, формі (враховуючи деформацію м'яча, яка може виникнути фізично або через артефакти відео) і компактності, що визначається як площа найменшого кола, яке обведене навколо кандидат-кулі. Для всіх кандидатів визначаються траєкторії, і справжня траєкторія м'яча-кандидата може бути встановлена на основі його фізичного руху. Під час перекриття траєкторій використовується інтерполяція. Нарешті, отримана траєкторія накладається на вихідні кадри.

Важливо відзначити, що в статті звертається увага на проблему розрахунку різниці між двома кадрами при ідентифікації швидкого м'яча. Запропоноване рішення полягає в застосуванні техніки трьохкадрової диференціації. У випадках, коли спостерігаються значні розриви в значеннях інтенсивності, також використовується додатковий метод виявлення країв. Для тестування були використані відеодані зі стандартної відеокамери Canon™ з роздільною здатністю 640x480 пікселів та частотою захоплення кадрів 30 Гц.

Результати дослідження показали точність виявлення м'яча на рівні 96,22% та точність відстеження на рівні 100%. У нашому сценарії настільного тенісу можна очікувати хороших результатів, якщо розмір м'яча значний в порівнянні з розміром кадру, а обмеження післяобробки для відстеження м'яча відповідають нашим очікуванням[11].

1.5.3 Комп'ютерний зір в бейсболі

Центральним елементом дослідження Теобальта та ін[12] є використання недорогих камер зі стробоскопічною технологією для зйомки бейсбольних траєкторій. Бейсбол був обраний через його велику швидкість та для вимірювання початкової швидкості, осі обертання та частоти обертання. Для цього на кулю були наклеєні оптичні маркери різних кольорів. Дослідження проводилися в темній кімнаті з використанням цифрової камери Olympus™ Camedia C5050 (Olympus America Inc., n.d.), яка була налаштована

на тривалу експозицію, а також з використанням стробоскопа з тривалістю 20 мкс і частотою 80 Гц, який функціонував як затвор.

Для сегментації фону за кольором та створення бінарної маски використовувався фреймворк OpenCV[13]. Центр кулі визначався шляхом побудови еліпсів на основі граничних точок та застосування кругового перетворення Хафа (СНТ) навколо центру еліпса для додаткової уточненості. Зафіксована швидкість обертання м'яча становила 1600 обертів на хвилину (об/хв), що відповідає швидкості понад 80 миль на годину. Оскільки недоступні точні дані щодо руху м'яча на землі, перевірка проводилась за допомогою фізичного моделювання. Остаточна точність використання чотирьох камер для отримання 3D зображення становила від 18% до 15% діаметра бейсбольного м'яча.

Хоча робота проводилась в затемненій кімнаті, що може бути складним для більшості гравців настільного тенісу, специфікації запису, якість камер та їх конфігурації мають велике значення для даного дослідження. Важливим аспектом є детальний опис у статті всіх конструктивних параметрів для експериментів, що дозволяє повторити дослідження та отримати корисну інформацію для подальших досліджень. Це ключове знання для опису використання комп'ютерного зору в настільному тенісі[12].

1.6 Комерційні системи на основі комп'ютерного зору

На сьогоднішній день алгоритми комп'ютерного зору (CV) широко використовуються для відстеження гравців, транспортних засобів і/або м'яча в різних видах спорту. Це зазвичай використовується для підтримки рішень суддів і задоволення запитів команд або гравців щодо перегляду і зміни прийнятих рішень суддів або арбітрів. При огляді літератури щодо резюме та спортивних технологій, необхідно коротко оцінити також системи, призначені для комерційного використання. Хоча деталі використовуваних алгоритмів часто є конфіденційними з питань інтелектуальної власності, наявність таких

систем надає деякі вказівки щодо використовуваного обладнання та програмного забезпечення, що можуть бути корисними в цій галузі досліджень резюме та настільного тенісу. Вимоги до таких систем також вказують на очікування щодо рішень, заснованих на CV, що застосовуються у спорті.

Більшість комерційних систем CV для спорту зосереджені на підтримці рішень суддів, аналізі ефективності або анотаціях для трансляцій. Це програми не просто виявляють м'яч, але також генерують форму об'єкта, його розмір, розташування та достатню кількість даних для аналізу руху. Це важливі аспекти для застосувань у сфері аналізу ефективності, автоматичного анотування, коучингу та підтримки рішень. Нижче наведено загальний опис основних комерційних рішень щодо резюме, які стосуються даної роботи[14].

1.6.1 Технологія «Hawk-Eye»

«Hawk-Eye» - це система на основі комп'ютерного зору (CV), яка була розроблена з 1999 року (Hawk-Eye Innovations Ltd, 2013) і пізніше придбана компанією Sony у 2011 році. Спочатку вона була розроблена для крикету як система суддівства, але зараз вона використовується в різних видах спорту, включаючи теніс, футбол, снукер і недавно легку атлетику. У всіх цих видах спорту «Hawk-Eye» вимагає високошвидкісні камери, які будуть правильно налаштовані (наприклад, шість камер у крикеті, сім у футболі), розміщені навколо арени, поля або корта. Дана система зарекомендувала себе як система з високою точністю та широким застосуванням у спорті, в тих моментах коли потрібно мінімізувати кількість помилок при прийнятті суддівських рішень та в спірних моментах. Вона сприяє мінімізації неточностей у суддівських рішеннях та збільшує ефективність суддівських процесів в спортивних змаганнях

Вона виконує такі функції:

1. Визначення траєкторії руху: Система здатна точно визначати положення об'єкта та траєкторію його руху. У великому тенісі, наприклад,

вона використовується для визначення положення м'яча в межах корту, або поза ними.

2. Підтвердження або виклик рішень: Hawk-Eye здатна використовуватись для визначення точного результату в спірних моментах. У тенісі, наприклад, гравці можуть викликати рішення арбітра, якщо вони не погоджуються з оцінкою системи щодо того, чи м'яч впав на лінію корта, чи за його межами, тобто аут.

3. Статистика та аналіз гри: Дана технологія також здатна надавати статистичні дані про рух об'єктів, їхню швидкість, траєкторію та інші параметри для аналізу гри.

У випадку з тенісом, який по свої суті дуже близький до настільного, обробка вхідних даних в системі «Hawk-Eye» пройшла чотири етапи:

- 1) 2D-інформація з кожної камери використовується для обробки CV з метою виявлення центру м'яча,
- 2) 3D-дані обчислюються за допомогою триангуляції з усіх доступних камер,
- 3) ці 3D-дані обробляються для отримання точної трекінгової інформації про рух м'яча[15].

1.6.2 Технологія «SportVU»

Система розроблена компанією Stats LLC у Чикаго[9] і призначена для відстеження гравців та прогнозування 3D-відеоданих. Це рішення для аналізу, відоме як SportVU (2012), зосереджується на відстеженні гравців і створенні статистики гри. Його застосовують для вимірювання відстані, швидкості та відношення (відстань) до м'яча в різних видах спорту, включаючи футбол, баскетбол і американський футбол. Серед його клієнтів є ЗМІ, трансляційні компанії та організації з коучингу.

Існують два варіанти впровадження системи: базовий і розширений. У базовому варіанті встановлюють три невеликі високошвидкісні камери в одному місці, розташовані на декілька сантиметрів одна від одної. Програмне

забезпечення ідентифікує об'єкти в полі зору та отримує дані про їх 3D-позиціонування, створюючи потік рухів кожного об'єкта в реальному часі і карту активності матчів. Розширене рішення передбачає використання шести високошвидкісних камер, розташованих у двох місцях у залі, як триплети. Камери можуть бути стаціонарними або портативними, кожна з них фіксує третину поля, і всі шість камер об'єднуються для створення панорамного огляду. Відстеження м'яча здійснюється з частотою 25 кадрів на секунду (FPS), а відстеження гравців - з частотою 15 FPS. Четвертий датчик використовується для виявлення нових об'єктів, таких як заміни гравців. Оператор системи повинен початково ідентифікувати гравців, судді та інших учасників.

Наразі SportVU не застосовується у виді спорту, такому як настільний теніс, оскільки швидкість гри та розміри поля в цих видів спорту значно відрізняються[16].

1.6.3 «Sport Vision»

Спортивна компанія SportVision[17] розробила низку інструментів і методів для відстеження об'єктів у різних видах спорту, включаючи бейсбол, американський футбол, автоспорт NASCAR і конні перегони. Вони надають реально-часові показники та покращені графічні зображення для подій у прямому ефірі. Спочатку їхніми клієнтами були переважно медіа-компанії, але востаннє споживачі також отримують доступ до цих технологій через портативні пристрої.

Один з продуктів SportVision, відомий як PITCH/fx, використовується у бейсболі для вимірювання швидкості кидка, обертання м'яча, швидкості та висоти руху м'яча в 60 точках під час його польоту. Для цього використовуються три високошвидкісні камери, розташовані на кожному стадіоні. Ця технологія застосовується для трансляційної накладної графіки та для тренувань, дозволяючи виміряти середню швидкість м'яча з точністю до 2,5 см, що дорівнює приблизно 93,05 миль/год. Хоча швидкість та обертання

м'яча в бейсболі мають деякі схожі характеристики з настільним тенісом, ця технологія не використовується в настільному тенісі.

1.6.4 Opta Sports

Opta Sports[8] - це одна з провідних компаній у галузі спортивної аналітики та статистики. Вони спеціалізуються на зборі, аналізі та наданні деталізованої статистики для різних видів спорту. Opta надає важливі дані для команд, тренерів, аналітиків, ЗМІ та спортивних фанатів з метою розуміння та аналізу спортивних подій.

Основні функції Opta Sports включають:

1. Збір статистики в реальному часі: Opta використовує високоточні технології, такі як оптичне відстеження та системи GPS, щоб відстежувати рух гравців та рух м'яча під час спортивних подій в реальному часі. Це дозволяє надавати докладну статистику та аналіз навіть під час гри.

2. Різні види спорту: Opta працює в різних видів спорту, включаючи футбол (соккер), регбі, крикет, баскетбол, хокей, американський футбол та інші. Вони надають адаптовану статистику для кожного виду спорту.

3. Деталізована статистика: Opta збирає широкий спектр статистичних даних, включаючи інформацію про голи, передачі, штрафи, кидки, різноманітні показники професіоналізму гравців, володіння м'ячем, атаки, оборону та багато іншого.

4. Використання в спортивному аналізі та бізнесі: Інформація, надана Opta, корисна для тренувального процесу команд, стратегічного аналізу, а також для створення спортивних програм, ставок і багатьох інших галузей.

5. Статистика в реальному часі та передача даних: Opta надає послуги статистики в реальному часі для ЗМІ та онлайн-платформ, також є функція надавання API для інтеграції своїх даних в різні додатки та веб-сайти[18].

1.6.5 Zebra MotionWorks

Zebra MotionWorks[19] - це технологічна платформа для відстеження руху гравців та обладнання в спортивних іграх, зокрема в американському футболі, американському футболі для відкритого повітря, регбі, хокеї на траві, гандболі, лакросі та інших колективних видів спорту. За допомогою системи Zebra MotionWorks можна отримувати високоточну інформацію про рух гравців і м'яча на полі, що дозволяє командам, тренерам і аналітикам отримувати цінні дані для стратегічного аналізу та покращення виступів.

Основні характеристики та можливості Zebra MotionWorks включають:

1. Висока точність вимірювань: Система використовує високоточні сенсори та технології відстеження для вимірювання руху гравців та обладнання з великою точністю. Це дозволяє збирати докладні дані про рух, швидкість, прискорення та інші параметри.

2. Система в реальному часі: Zebra MotionWorks надає дані в режимі реального часу, що дозволяє тренерам та аналітикам слідкувати за подіями на полі під час гри і приймати швидкі стратегічні рішення.

3. Аналіз даних: Платформа включає в себе інструменти для аналізу зібраних даних. Це дозволяє тренерам і аналітикам вивчати рух гравців, визначати стратегії та піддавати оцінці ефективність команди.

4. Інтеграція з іншими системами: Zebra MotionWorks може інтегруватися з іншими системами аналізу та відеоспостереження для отримання комплексної інформації про гру.

5. Використання в тренуваннях і підготовці: Дані, зібрані системою, можуть бути використані для покращення тренувального процесу, збільшення фізичної підготовки гравців та розробки стратегії.

6. Оптимізація стратегії: Збирати дані з реального полі гри дозволяє тренерам розробляти більш ефективні гравчині тактики та стратегії[19].

1.6.7 Kinexon

Kinexon[20] - це компанія, яка розробляє інноваційні системи з аналітики та відстеження руху для спортивних команд, тренерів і аналітиків. Вони спеціалізуються на використанні сучасних технологій для збору даних про рух гравців та обладнання на полі, що дозволяє зрозуміти і поліпшити спортивні виступи.

Основні характеристики та можливості системи Kinexon включають:

1. Висока точність вимірювань: Системи Kinexon використовують високоточні сенсори та технології відстеження, такі як радіочастотна ідентифікація (RFID) та ультразвук, для вимірювання руху гравців, м'ячів та іншого обладнання з великою точністю.
2. Реальний час: Дані, зібрані системою Kinexon, доступні в режимі реального часу, що дозволяє тренерам та аналітикам відстежувати гру в реальному часі та реагувати на події на полі.
3. Аналіз даних: Компанія надає інструменти для аналізу зібраних даних, включаючи інформацію про рух, швидкість, прискорення та інші параметри. Це дозволяє тренерам і аналітикам розробляти стратегії та оптимізовувати тренувальний процес.
4. Інтеграція з іншими системами: Системи Kinexon можуть легко інтегруватися з іншими системами аналізу даних та відеоспостереження для отримання комплексної інформації про гру.
5. Використання в різних видах спорту: Kinexon використовується в різних видів спорту, включаючи футбол, баскетбол, хокей, американський футбол, регбі, бейсбол та інші. Вони працюють з командами і спортивними організаціями по всьому світу.
6. Оптимізація тренувань і підготовки: Дані, зібрані системою Kinexon, можуть бути використані для покращення фізичної підготовки гравців та розробки ефективних тренувальних програм.[20].

1.7 Наукові дослідження стосовно комп'ютерного зору в настільному тенісі

Дослідження Tao Ning та інших [21] запропонувало метод визначення місця приземлення м'яча в настільному тенісі, заснований на інформації просторового домену. Цей підхід знижує залежність від обладнання для збору візуальних даних і підвищує швидкість обробки, використовуючи динамічне порогове значення кольору та метод чотирьох кадрів для визначення точки падіння м'яча.

Патрік Вонг та інші [22] представили багатовидову, недорогу систему суддівства для настільного тенісу. Використовуючи багатоагентську систему (MAS), ця система виявляє та відстежує м'яч, створюючи 3D-представлення його траєкторії та аналізуючи ключові аспекти гри, що має потенціал реформувати спосіб арбітражу матчів, особливо для аматорських гравців.

Наяра Роша та інші [23] зосередились на розробці системи на основі комп'ютерного зору для аналізу тенісних матчів. Вони використовують машинне навчання та морфологічні операції для визначення позиції м'яча, ліній корту та розташування гравців, а також для аналізу моментів удару м'яча. Дослідження [24] вводить нову систему VAR (відео-асистент судді), засновану на Kinect і Raspberry Pi, для настільного тенісу. Основна ідея полягає в створенні портативної системи, яка може бути використана з будь-яким типом LCD-систем і допомагати суддям під час гри для кращого підрахунку м'ячів та підвищення якості гри.

Дослідження [25] зосереджується на швидкому відстеженні м'яча в настільному тенісі з використанням недорогого обладнання. Розроблені ефективні алгоритми дозволяють точно відстежувати м'яч навіть у складних умовах, як-от перекриття об'єктами та змішування кольорів, сприяючи розвитку автоматичної суддівської системи.

У роботі Пареша Р. Камбле та інших [26] представлено докладний огляд досліджень щодо відстеження м'яча у спортивних відео. Основна увага

приділяється виявленню та відстеженню м'яча, а також розглядаються виклики, пов'язані з цим завданням, та обговорюються можливі шляхи вдосконалення процесу відстеження.

Дослідження [28] пропонує підхід до виявлення швидко рухомих, невеликих м'ячів у реальному часі, базуючись на моделі об'єктного виявлення YOLOv3. При цьому застосовуються конкретні налаштування архітектури мережі та процесу навчання для покращення точності та швидкості виявлення.

У дослідженні [29] аналізуються методи комп'ютерного зору для оцінки позицій гравців у паделі, що є ключовим для аналізу ефективності гри. Порівняння точності різних сучасних методів показало, що найкращі результати досягаються за допомогою методів оцінки позиції з використанням глибоких згорткових нейронних мереж.

У роботі [30] пропонується підхід до цифрового спортивного тренування, заснований на обробці цифрових відеозображень, що дозволяє зменшити залежність від досвіду тренерів у процесі тренувань з настільного тенісу. Розроблена система використовує технології захоплення руху та аналізу характеристик руху для покращення процесу тренувань.

Дослідження Sanusi et al.[31] зосереджене на розробці системи "Table Tennis Tutor" (T3), яка використовує мультимодальні дані та нейронні мережі для класифікації правильних та неправильних ударів настільного тенісу. Система включає в себе смартфон із вбудованими датчиками руху та камеру Microsoft Kinect для відстеження позиції тіла гравця.

1.8 Постановка завдання

У вище згаданих дослідженнях розглядається застосування алгоритмів комп'ютерного зору в спорті, зокрема для відстеження м'яча та гравців. Системи, такі як "Hawk-Eye", "SportVU", "SportVision", "Opta Sports", "Zebra MotionWorks" та "Kinexon", використовують високошвидкісні камери та сенсори для точного відстеження м'яча та гравців у різних видах спорту.

Дослідження Tao Ning та інших запропонувало метод для визначення місця приземлення м'яча в настільному тенісі, заснований на інформації просторового домену. Патрік Вонг та інші представили багатовидову систему суддівства для настільного тенісу, яка може аналізувати 3D-траєкторію м'яча. Інші дослідження, такі як [15], [16], [17], [18] та [19], фокусуються на розробці новітніх систем для відстеження м'яча та аналізу гри з використанням машинного навчання, глибоких нейронних мереж та обробки цифрових відеозображень.

Однак, у цих дослідженнях існують виклики та недоліки, які включають складність відстеження в умовах перекриття та змішування кольорів, обмежену точність при визначенні швидко рухомих об'єктів, а також технічні та фінансові обмеження, пов'язані з використанням високоточного обладнання. Необхідність узагальнення алгоритмів для різних видів спорту та розширення можливостей недорогих систем є ключовими аспектами, які потребують подальших досліджень та розвитку.

У контексті даного дослідження, яке зосереджено на розробці системи для відстеження м'яча в настільному тенісі, важливо вивчити аналогічні розробки в цій області (Таблиця 1). Для цього розглянуто декілька найбільш відповідних досліджень, які використовують комп'ютерний зір та інші технології для точного відстеження руху м'яча або елементів гри.

Таблиця 1 - Порівняльний аналіз найближчих досліджень

Автор(и)	Назва	Основні моменти	Методологія	Особливості
Tao Ning, Changcheng Wang, Meng Fu, Xiaodong Duan [11]	Виявлення місця приземлення м'яча в настільному тенісі	Використання інформації просторового домену для визначення точки приземлення м'яча, зменшення	Динамічне порогове значення кольору, метод чотирьох кадрів, просторова Евклідова	Висока точність (78.5%), швидкість обробки 45.3 fps.

		залежності від візуального обладнання.	відстань, екстракція ключових кадрів.	
Patrick Wong, Hnin Myint, Adrian Hopgood [12]	Багатовидова автоматична система суддівства в настільному тенісі	Низьковартісна, портативна, багатовидова система для суддівства в настільному тенісі.	Багатоагентська система (MAS), багатовидова камера, відстеження м'яча, 3D-моделювання траєкторії м'яча.	Мінімальна затримка, потенціал для паралельної архітектури, масштабованість.
Khaleel Asyraaf Mat Sanusi, Daniele Di Mitri, Bibeg Limbu, Roland Klemke [20]	Table Tennis Tutor: Forehand Strokes Classification Based on Multimodal Data and Neural Networks	Розробка системи для класифікації ударів настільного тенісу, використання мультимодальних даних та нейронних мереж для навчання початківців.	Використання смартфона з датчиками руху та Microsoft Kinect для відстеження позиції тіла, рекурентні нейронні мережі для класифікації ударів.	Потенціал для реального часу зворотного зв'язку, підтримка навчання початківців, використання доступних технологій.

У зв'язку з ростом популярності настільного тенісу і зростаючими вимогами до об'єктивності суддівства, метою даного дослідження є розробка інноваційної системи для автоматичного відстеження м'яча і рахунку в настільному тенісі. Дана система, на відміну від існуючих комерційних рішень, як то [11, 12, 13], є більш доступною і простою у використанні, орієнтована на аматорський рівень і може бути впроваджена без необхідності використання складного та дорогого обладнання.

Новизною роботи є розробка портативної системи, яка використовує стандартну відеокамеру і мікрокомп'ютер Raspberry Pi для відстеження м'яча, а також спеціально розроблений алгоритм для ведення рахунку і визначення

правильності подачі та ударів. Ця система дозволяє не тільки забезпечити більш справедливий та об'єктивний контроль за ходом гри, але й знижує залежність від людського фактора у суддівстві, забезпечуючи більш точні та швидкі рішення.

Таким чином, запропонована система відкриває нові перспективи для аматорського спорту, дозволяючи гравцям настільного тенісу зосередитися на грі, а не на суддівстві, і сприяє підвищенню інтересу та залученості до цього виду спорту.

Виявлення та відстеження м'ячів або кулястих об'єктів за допомогою обробки зображень для спортивних подій з використанням Python – на сьогоднішній день є популярною задачею для дослідників, які займаються розробкою таких систем на базі недорогих мікрокомп'ютерів [1].

В більшості випадків використовується камера [2] для зйомки відеозаписів м'яча в реальному часі, які потім обробляються за допомогою бібліотеки Python OpenCV [3]. Така популярність пов'язана з універсальністю ПО, яке може легко портуватись на різні комп'ютерні системи. Але перед тим як почати відслідковувати м'яч, потрібно правильно знайти кольорові плями (тенісний м'яч теж відноситься до цієї категорії) певної кольорової гами.

В установці з однією камерою відстеження об'єктів відбувається шляхом віднімання фону з подальшим встановленням відповідності регіону, що враховує різні ознаки, такі як швидкість об'єкта, його розміри та відстань до обмежувальної рамки[4]-[5].

Висновки до розділу 1:

У даному розділі були представлені всі відповідні поняття та термінологія, які в подальшому необхідними для розуміння дипломної роботи. Він включав в себе всі фундаментальні поняття щодо комп'ютерного зору, його використання в спорті, та, що більш важливе, його, перспектив в даному

напрямі, машинне навчання та один з його розділів - розпізнавання образів та його алгоритми.

Тож можна підвести підсумки і сказати, що цифрові технології в спорті задіяні на дуже низькому рівні, проте як технологій на базі комп'ютерного зору, так і технології на базі машинного навчання, а саме розпізнаванні образів мають великі перспективи в даному напрямі, що з практичної точки зору, що з наукової.

2 МЕТОД ВІДСТЕЖЕННЯ ОБ'ЄКТІВ ТА ВИКОНАННЯ СУДДІВСЬКИХ ПРОЦЕСІВ В НАСТІЛЬНОМУ ТЕНІСІ

2.1 Концепція системи

Суть пропонованого методу полягає в відстеженні об'єкта за допомогою технології пошуку кольорової плями, конкретного заданого кольору, або діапазону кольорів, яким повинен відповідати колір відстежуваного об'єкта, фіксованого, або максимального розміру, конкретної форми, або просто плями.

В даному випадку відстежуваним об'єктом є м'яч для пінг-понгу [6] , який представляє собою кулю діаметром 40мм оранжевого кольору, котра рухається на великій швидкості(до 30м/с) за непередбачуваною траєкторією із великою кількістю відскоків і в цілому на кадрах знятих більш дешевими камерами дуже сильно розмивається.

В розробці даного методу можна виділити такі ключові етапи:

- Завантаження потоку

Однією з ключових функцій завантаження потоку є можливість роботи з зображеннями в різних форматах, таких як JPEG, PNG, BMP тощо. Також, OpenCV дозволяє відкривати та обробляти відеопотоки з вебкамер або відеофайлів, виконувати операції з обробки та аналізу кадрів у реальному часі.

Пакет також надає засоби для виявлення об'єктів, розпізнавання паттернів, класифікації зображень та аналізу відеопотоків.

Процес захоплення відеопотоку починається зі створення об'єкту захоплення за допомогою виклику методу `cv2.VideoCapture()`, де вибирається джерело відеопотоку. Для того щоб захопити відео з вбудованої камери, вказується її номер у функції. Наприклад, `cap = cv2.VideoCapture(0)` захопить відео з першої доступної камери. Потім виконується читання кадрів за допомогою функції «`read()`» об'єкта `VideoCapture`, котра дозволяє отримати

новий кадр з відеопотоку. Цей метод повертає два значення: перше - `ret`, яке показує, чи був успішно прочитаний кадр, а друге - `frame`, що містить сам кадр.

- Створення бінарного зображення.

Створення бінарного зображення полягає у перетворенні кольорового чи чорно-білого зображення на зображення, де пікселі приймають тільки два значення: чорний або білий, з метою виділення або відокремлення певних областей на основі заданих порогових значень яскравості чи насиченості, в нашому випадку даному діапазону повинні відповідати колір та відтінки оранжевого тенісного м'ячика, він має наступні значення: Hue мінімальне = 25, Saturation мінімальне = 25, Value мінімальне = 125, Hue максимальне = 50, Saturation максимальне 50, та Value максимальне 255.

Використовується дане перетворення за допомогою функції пакета Cv2 «`cv2.inRange()`».

- Застосування операцій морфології.

Для покращення якості знаходження контурів та об'єктів можна застосовувати операції морфології, такі як згладжування, та розширювання, які застосовуються для виокремлення областей інтересу, видалення шуму, відновлення об'єктів чи для покращення якості зображень. Їх застосування дозволяє підготувати дані для подальшого використання у програмах комп'ютерного зору, що позитивно впливає на якість обробки.

- Згладжування (Erosion) - це операція яка зменшує розмір об'єктів на зображенні шляхом видалення зовнішніх пікселів по контуру об'єкта. Це використовується для видалення шуму, створення границь об'єктів, або зменшення товщини контуру. Виконується за допомогою функції «`cv2.erode()`».

- Розширення (Dilation): На відміну від ерозії, дана операція збільшує розмір об'єктів на зображенні шляхом додавання пікселів до контуру об'єкта. Вона корисна для заповнення малих отворів, об'єднання об'єктів та збільшення товщини контуру. Йому відповідає функція «`cv2.dilate()`».

- Знаходження контурів в бінарному зображенні.

Після попередніх підготувань можна безпосередньо задати параметри для пошуку об'єкта (тенісний м'ячик) та виконати його. Знаходження контурів в бінарному зображенні проводиться за допомогою функції «cv2.findContours()», де задаються параметри форми шуканого об'єкта – коло, це виконується за допомогою функції «cv2.circle()». З метою відсіювання дрібних точок, або об'єктів задаються мінімальний та максимальний радіус об'єкта, цикл повторюється до тієї пори поки його не буде знайдено, або коли роботу програми не перерве користувач.

- Виділення об'єкта.

Коли об'єкт знайдено поотрібно це візуалізувати для користувача, для цього виконується виділення. Периметр обводиться жовтим колом, а його геометричний центр червоною точкою. Так як тенісний м'яч рухається зі швидкістю до 40 м/с з нелінійною траєкторією то людському оку встежити за ним досить важко, тож, додатково виконується візуалізації траєкторії руху: між кількома останніми точками розташування центру малюється червона лінія, за допомогою функції «cv2.line()».

- Суддівські процеси.

Одним з найскладніших етапів роботи буде створити такий алгоритм який зможе максимально ефективно вести рахунок як по очках так і загальний по раундах. Для цього потрібно створити високоефективну систему розпізнавання «голових ситуацій», тобто ситуацій коли гравцю повинно зарахуватись очко.

- Інтерфейс користувача.

Інтерфейс забезпечує взаємодію користувача та програми, виводить оброблене зображення, рахунок.

2.2 Архітектура системи

Пропонована система з автоматичного суддівства в настільному тенісі використовує інноваційний підхід до обробки та аналізу відеоданих. Вона

інтегрує передові технології візуального розпізнавання та алгоритми відстеження для точного ведення рахунку та суддівства в матчах настільного тенісу. Архітектура запропонованої системи представлена на рисунку 1.

Камера (див.рис.1 блок 1) передає відео в цифровому вигляді на мікрокомп'ютер Raspberry Pi (див.рис.1 блок 2), для подальшої обробки. Першим етапом є захоплення відео (див.рис.1 блок 2.1) та ділення його на кадри(фрейми), далі з отриманими даними починає взаємодіяти система розпізнавання зображення (див.рис.1 блок 2.2), яка аналізує кожен кадр як фото з метою забезпечити трекінг відстежуваного об'єкта – тенісного м'ячика, встановлює його положення. Тепер отримані з отриманими даними буде працювати суддівський алгоритм (див.рис.1 блок 2.3), який повинен забезпечити виконання суддівських процесів, а саме відстеження моментів коли гравці отримують очки, та ведення рахунку. В кінцевому результаті отримане оброблене зображення та рахунок виводяться в вигляді вікна з таблом, що й становить користувацький інтерфейс (див.рис.1 блок 3).

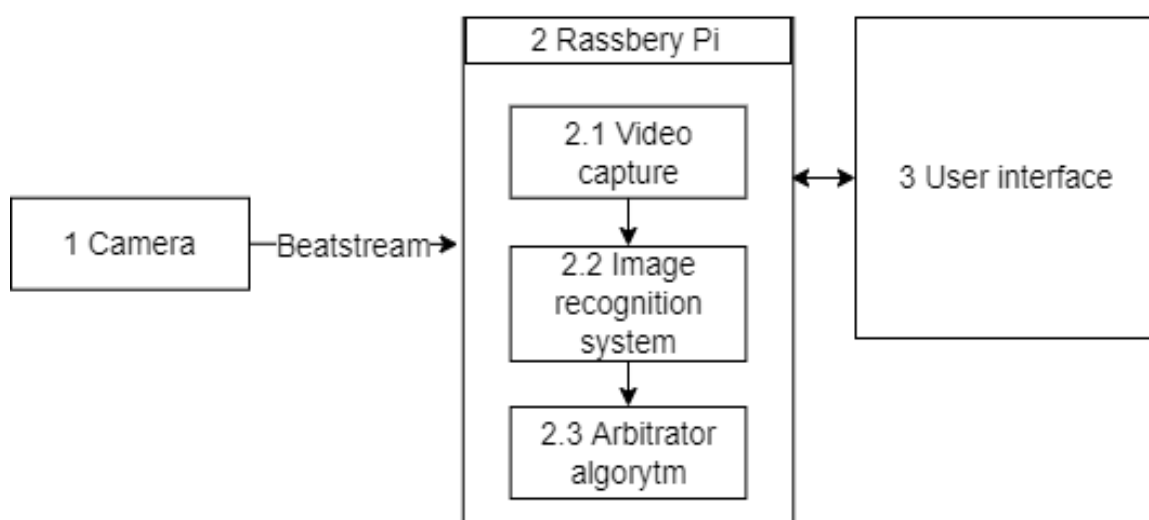


Рисунок 2.1 – Узагальнена архітектура системи

2.3. Пропонований метод

Для реалізації системи відстеження м'яча в настільному тенісі розроблено наступний метод, представлений алгоритмом (рис.2) та поділений на декілька етапів:

Етап 1. Підготовка. Задаються змінні з рахунком «Player1», «Player2» (див. рис.2 блок 1), за допомогою пакету Tkinter створюється вікно програми (див. рис.2, блок 2), до якого додаються віджети для виводу обробленого відеопотоку та рахунку (див. рис.2 блок 3), виводиться рахунок (див. рис.2 блок 4), за замовчуванням, «0:0».

Етап 2 Оголошення функцій «process_video()» (див. рис.2 блок 5), яка відповідає за трекінг та арбітражні процеси. Виконується ініціалізація парсера аргументів (див. рис.2 блок 6). Використовуючи пакет argparse, ми створили парсер аргументів, що дозволяє програмі приймати відео як аргумент. Це означає, що користувач може передати шлях до відеофайлу, який буде використовуватися для подальшої обробки. Виконується сам парсинг (див. рис.2 блок 7), якщо відеофайлу не має, буде використовуватись зображення з камери.

Обробка часу представлена на рисунку 2 блок 8. За допомогою функції time ми отримуємо точний час для кожного захопленого кадру. Це важливо для синхронізації подій у відео.

Етап 3. Ініціюється цикл While (див. рис.2 блок 9) та починається процес захоплення відео. Використовуючи функцію VideoCapture з пакету cv2, програма захоплює відео (див. рис.2 блок 10). Це перший крок у обробці відеоданих.

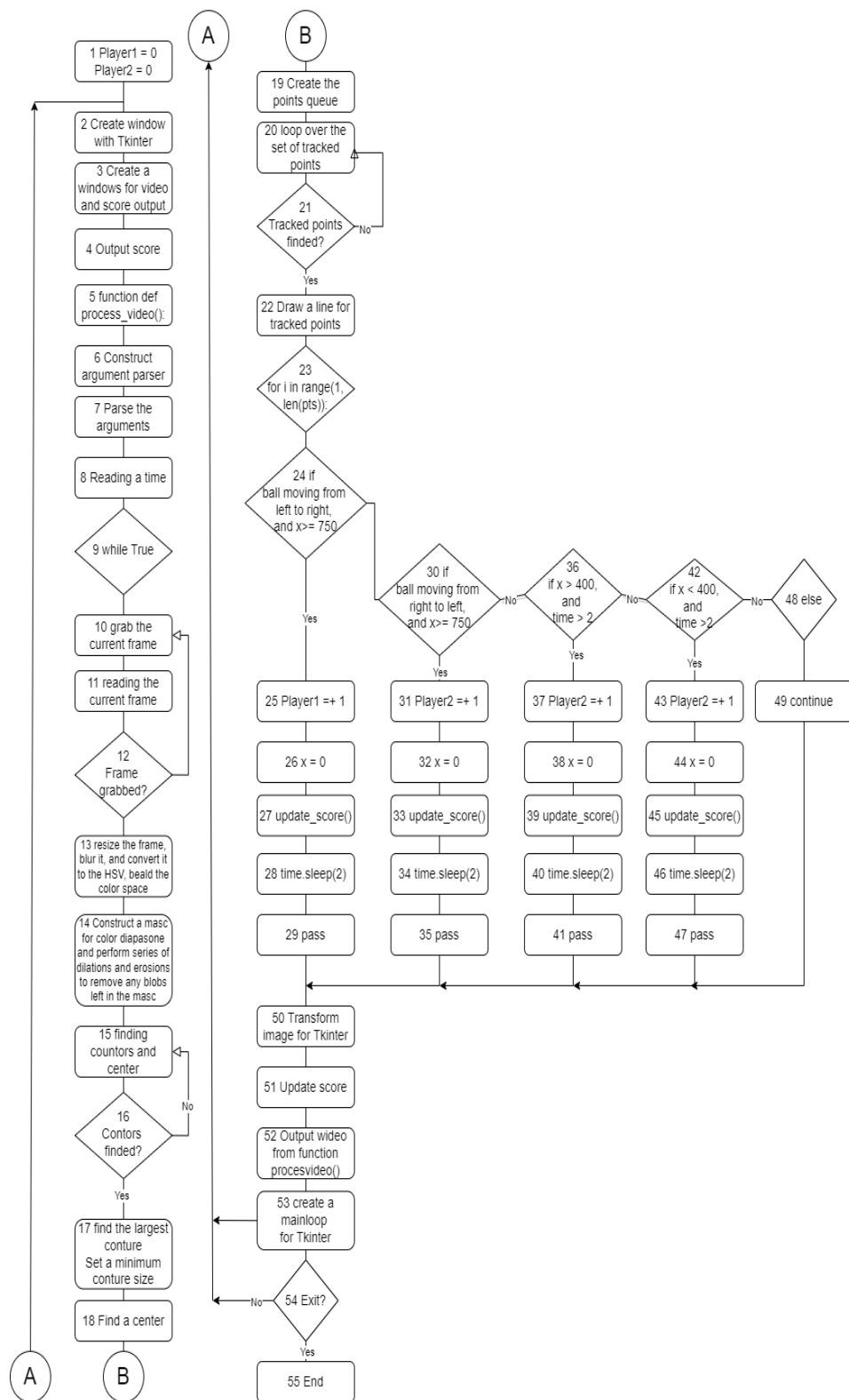


Рисунок 2.2 – Алгоритм системи «Електронний суддя для пінг-понгу»

Читання і обробка кадру. Захоплюємо кадр з отриманого відеопотоку(див. рис.2 блок 11) та перевіряємо, чи кадр успішно захоплено(див. рис.2 блок 12). Якщо так, то відбувається перетворення

розміру кадру, його розмиття та переведення з палітри RGB у палітру HSV (рис.3) за допомогою функцій `resize`, `GaussianBlur` та `cvtColor` (див. рис.2 блок 13).

Тут варто зосередити увагу на тому, що з себе представляють обидві палітри.

RGB - це спосіб представлення кольорів у просторі RGB (Red, Green, Blue). У цій моделі будь-який колір можна утворити шляхом комбінації трьох основних кольорів: червоного (Red), зеленого (Green) та синього (Blue), її можна побачити на рисунку 3.

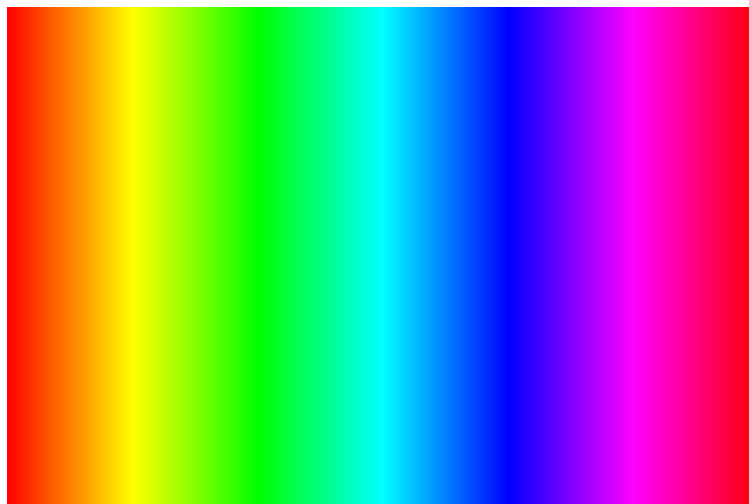


Рисунок 2.3 – Палітра RGB

Кожен колір у палітрі RGB виражається числовими значеннями для червоного, зеленого та синього каналів. Кожен канал приймає значення від 0 до 255, де 0 представляє відсутність колірного складу, а 255 - максимальний інтенсивний колір у цьому каналі.

Наприклад, білий колір представлений як (255, 255, 255), де червоний, зелений та синій канали мають максимальне значення. Чорний колір відповідає (0, 0, 0), що означає відсутність колірного складу у всіх каналах.

Ця модель широко використовується в комп'ютерній графіці, фотографії, відео та інших сферах, де кольори виражаються числово для подальшої обробки та відображення.

HSV (Hue, Saturation, Value) або тон, насиченість та значення - це інша модель кольору, яка відрізняється від RGB, її схему можна побачити на рисунку 4.

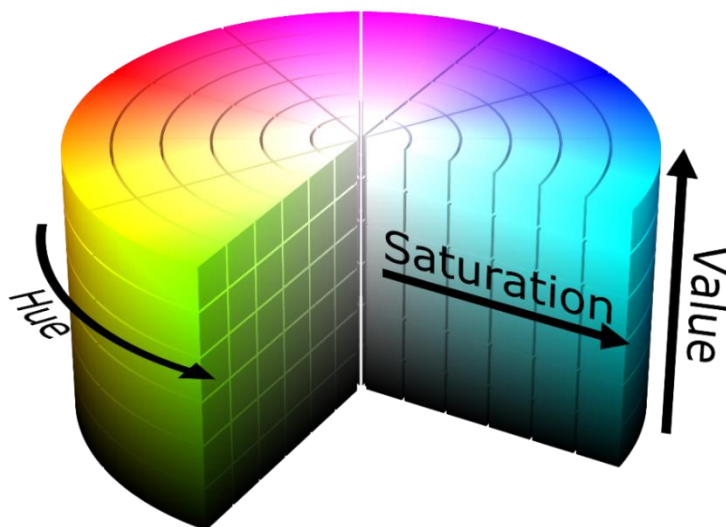


Рисунок 2.4 – Палітра HSV[20]

У цій моделі кожен колір представляється трьома параметрами:

- Відтінок (Hue): Це параметр, який визначає тип кольору (наприклад, червоний, синій, зелений тощо) у градації колірного кола. Він вимірюється в градусах від 0 до 360.
- Насиченість (Saturation) вимірює насиченість кольору. Низьке значення насиченості призводить до відтінків сірого, тоді як висока насиченість вказує на чистоту кольору. Вимірюється від 0 до 100%.
- Значення (Value): Вимірює яскравість або світлоту кольору. Значення V вказує на яскравість кольору, де 0 - це чорний колір і 100 - це повна яскравість. Також вимірюється від 0 до 100%.

Дана палітра має наступні переваги над палітрою RGB:

- Інтуїтивне розуміння: HSV спрощує сприйняття кольорів, розділяючи їх на тон, насиченість та значення. Це робить кольори більш доступними для маніпулювання та розуміння у графічних програмах.

- Фільтрація кольорів: Використання HSV полегшує відбір кольорів за конкретними параметрами, такими як тон, насиченість або значення.

- Простота налаштування насиченості та яскравості: Зміна цих параметрів у HSV спрощує управління яскравістю та насиченістю кольорів без необхідності регулювання окремих значень червоного, зеленого та синього, як це відбувається в RGB.

- Зручність в розпізнаванні відтінків: В HSV відтінок виражений як окремий параметр, що полегшує розрізнення та роботу з кольорами в залежності від їхнього типу.

Зважаючи на вище вказані переваги дана палітра більше підходить для виконання нашого завдання, та допоможе покращити якість виявлення об'єкта.

Етап 3 Конструювання маски(див. рис.2 блок 14). Для виявлення м'яча створюється маска з бінарним зображенням для ізоляції певного діапазону кольорів, використовуючи функцію `inRange` з параметрами `lower = (25,25,125)` та `upper = (50,50,255)`. Потім маска піддається серії ерозій та розширень для видалення будь-яких плям. Конструювання маски для виявлення м'яча в комп'ютерному зорі включає наступні математичні кроки:

- Перетворення Кольору: вихідне зображення I перетворюється з RGB в HSV формат (Hue, Saturation, Value), щоб спростити виявлення кольорів.

- Визначення Діапазону Кольорів: встановлюються порогові значення для виявлення кольору м'яча. Для цього використовуються два вектори: нижній поріг $lower = (H_{\min}, S_{\min}, V_{\min})$ і верхній поріг $upper = (H_{\max}, S_{\max}, V_{\max})$.

- Створення Маски: застосовується функція `inRange` до HSV-зображення для створення маски M , яка виділяє області, кольори яких лежать

в заданому діапазоні:

$$M_{(x,y)} = \begin{cases} 1 \text{ якщо } H_{min} \leq H_{(x,y)} \\ \text{та } S_{min} \leq S_{(x,y)} \leq S_{max} \\ \text{та } V_{min} \leq V_{(x,y)} \leq V_{max} \\ 0 \text{ інакше} \end{cases} \quad (1)$$

де $H(x,y), S(x,y), V(x,y)$ - це відтінок, насиченість та яскравість пікселя у точці

(x,y) на HSV-зображенні.

$H_{min}, S_{min}, V_{min}$ - нижні порогові значення для відтінку, насиченості та яскравості.

$H_{max}, S_{max}, V_{max}$ - верхні порогові значення..

Маска M піддається серії морфологічних операцій, таких як ерозія та розширення, для видалення шумів. Ерозія та розширення виконуються шляхом застосування структурного елемента S до маски M . Ерозія E визначається як:

$$E(M) = M \ominus S$$

Розширення D визначається як:

$$D(M) = M \oplus S$$



Рисунок 2.5 – М'ячик в бінарному зображенні

Етап 6. Пошук кольорової плями. Наступним кроком є пошук кольорової плями заданого діапазону (див. рис.2 блок 15). Виконується перевірка, якщо плями не знайдено, припускається, що м'яч відсутній на столі, і пошук продовжується (див. рис.2 блок 16).

Пошук відбувається шляхом аналізу областей у масці M , де є значення 1 (білий колір), що вказує на присутність м'яча.

Етап 7. Відстеження м'яча. Після виявлення м'яча на столі здійснюється відстеження його контурів (див. рис.2 блок 17) та центру (див. рис.2 блок 18). М'яч виділяється жовтим кольором, а центр - червоним. Також здійснюється відстеження попередніх точок знаходження центру м'яча (tracked points).

Етап 8. Візуалізація траєкторії м'яча. Задається функція для набору "tracked points" (див. рис.2 блок 19). Здійснюється відстеження точок знаходження центру м'ячика (див. рис.2 блок 20), тоді відбувається перевірка, якщо вони не знаходяться то м'ячик не на столі і функція повторюється доти поки він не повернеться (див. рис.2 блок 21) в гру. Для кращого розуміння траєкторії м'яча між точкою положення м'яча на даному фреймі та аналогічною точкою попереднього фрейму малюється червона лінія (див. рис.2 блок 22).

- Математично цей процес може бути описаний наступним чином:
- Визначення положення м'яча: У кожному кадрі фіксується положення м'яча як координати (x, y) . Для кадру i це положення позначається як (x_i, y_i) .
- Визначення положення м'яча у наступному кадрі: Аналогічно, для наступного кадру $i+1$ фіксується положення м'яча як (x_{i+1}, y_{i+1})
- Побудова лінії: Траєкторія м'яча візуалізується шляхом побудови лінії, яка з'єднує ці дві точки. Математично це можна виразити як лінію, що з'єднує точки (x_i, y_i) та (x_{i+1}, y_{i+1}) .
- Цей процес повторюється для кожної пари послідовних кадрів, що дозволяє створити неперервну лінію, яка відображає траєкторію руху м'яча протягом всього відеозапису. Червона лінія, що візуалізує траєкторію,

допомагає користувачеві краще розуміти динаміку руху м'яча в ігровому процесі.

Етап 9. Обрахунок результатів матчу. Оголошується цикл For(див. рис.2 блок 23) для умови «for i in range(1, len(pts)):», котра означає, що позиція м'ячика зараз відстежується. Алгоритм ведення рахунку в системі автоматичного суддівства настільного тенісу починається з визначення траєкторії м'яча шляхом аналізу його координат (x, y) у кожному кадрі, перевіряючи чи перетинає м'яч віртуальну лінію, встановлену на середині столу уздовж осі Y.

Далі слідує опис «головних ситуацій», тобто таких коли гравцям зараховуються очки, вони складаються з 4х умов:

1. Якщо м'ячик рухається з ліва на право та x, тобто його координати більше за 750(див. рис.2 блок 24), тоді лівому гравцю(Player1) зараховується очко(див. рис.2 блок 25), значення x обнуляється(див. рис.2 блок 26), викликається функція для оновлення рахунку «score.update» (див. рис.2 блок 27), за допомогою функції «sleep» пакету «time» встановлюється затримка на 2 секунди(див. рис.2 блок 28), цикл пропускається(див. рис.2 блок 29).

2. Якщо м'ячик рухається з права на ліво та x, тобто його координати менше за 50(див. рис.2 блок 30), тоді правому гравцю(Player2) зараховується очко(див. рис.2 блок 31), значення x обнуляється(див. рис.2 блок 32), викликається функція для оновлення рахунку «score.update» (див. рис.2 блок 33), за допомогою функції «sleep» пакету «time» встановлюється затримка на 2 секунди(див. рис.2 блок 34), цикл пропускається(див. рис.2 блок 35).

3. Якщо м'ячик знаходиться на половині правого гравця довше 2х секунд(див. рис.2 блок 36) то аналогічним чином лівому гравцеві зараховується очко(див. рис.2 блок 37), значення x обнуляється(див. рис.2 блок 38), викликається функція для оновлення рахунку «score.update» (див. рис.2 блок 39), за допомогою функції «sleep» пакету «time» встановлюється затримка на 2 секунди(див. рис.2 блок 40), цикл пропускається(див. рис.2 блок 41).

4. Якщо м'ячик знаходиться на половині лівого гравця довше 2х секунд(див. рис.2 блок 42) то аналогічним чином лівому гравцеві зараховується очко(див. рис.2 блок 43), значення x обнуляється(див. рис.2 блок 44), викликається функція для оновлення рахунку «score.update» (див. рис.2 блок 45), за допомогою функції «sleep» пакету «time» встановлюється затримка на 2 секунди(див. рис.2 блок 46), цикл пропускається(див. рис.2 блок 47).

Затримка на 2 секунди необхідна для того щоб м'ячик встиг зникнути зі столу так як в протилежному випадку

Рахунок зараховується в залежності від напрямку руху м'яча: якщо м'яч рухається з ліва на право і перетинає іншу лінію, котра знаходиться біля краю фрейму(див. рис.2 блок 24) очко отримує лівий гравець, тоді як рух з права на ліво приносить очко правому гравцеві(див. рис.2 блок 30). Додатково, якщо м'яч залишається на половині столу одного з гравців довше ніж 2 секунди, очко зараховується супернику(див. рис.2 блок 36, 42).

Рахунок оновлюється згідно цих правил, а гра завершується, коли один з гравців досягає встановленої кількості очок.

Математично алгоритм пошуку «голових ситуацій» можна описати наступним чином:

Нехай $P(x, y, t)$ описує позицію м'яча в часі t , де x та y — координати м'яча.

L_y — координата лінії на осі Y , що ділить стіл на дві частини.

Очки зараховуються за наступними правилами:

Якщо $P(x, y, t)$ перетинає L_y з ліва на право, очки_лівого $+= 1$, .

Якщо $P(x, y, t)$ перетинає L_y з права на ліво, очки_правого $+= 1$, .

Якщо $P(x, y, t)$ знаходиться на одній половині столу більше 2 секунд, зараховується очко протилежному гравцеві.

Етап 10. Відображення результатів. За допомогою функції «cvtColor» та функцій «fromarray», «PhotoImage», пакету «PIL» виконується трансформування зображення для його виводу в інтерфейс(див. рис.2 блок 50).

Задається функція для оновлення табла з результатами(див. рис.2 блок 51), яке виводилось на початку роботи програми(див. рис.2 блок 4), виводиться оброблене відео(див. рис.2 блок 52). Для відображення вікна постійно, а не на мить створюється петля за допомогою функції «Tkinter mainloop()»(див. рис.2 блок 53), дана дія необхідна для відображення інтерфейсу. Виконується перевірка(див. рис.2 блок 54), якщо здійснюється закриття вікна програми то її робота припиняється(див. рис.2 блок 55), звільнюється камера, якщо ні то програма продовжує свою роботу і цикл запускається заново, починаючи з переходу до функції створення вікна інтерфейсу(див. рис.2 блок 2), на кожній ітерації програми воно створюється заново, в силу специфіки роботи пакету, проте це не помітно так як цикл робить шістдесят ітерацій протягом секунди, тобто ми отримуємо стандартні для людського ока 60 FPS.

Отже, основні компоненти цього методу включають ініціалізацію обробки відеоданих, перетворення зображень з RGB в HSV формат для ефективнішого виявлення м'яча, створення маски та відстеження траєкторії м'яча. Використовуючи ці дані, метод автоматично визначає напрямок руху м'яча і зараховує очки згідно з правилами настільного тенісу.

Висновки до розділу 2:

В даному розділі було запропоновано метод розробки системи електронного суддівства в настільному тенісі. Сучасний підхід до відстеження м'ячів або кулястих об'єктів в спортивних заходах, використовуючи обробку зображень та програмування з використанням Python, що представляє значний інтерес для дослідників і практиків. Особливо актуальним стає використання недорогих мікрокомп'ютерів для створення таких систем, що сприяє поширенню та доступності цієї технології.

Однак перед початком відстеження м'яча, важливо правильно визначити кольорові плями або області, які відповідають м'ячу (наприклад, тенісний м'яч), що вимагає певних навичок та підготовки.

Пропонований метод полягає у виявленні заданої кольорової плями або діапазону кольорів, що відповідає м'ячу. В цій реалізації застосовується пошук оранжевого тенісного м'яча діаметром 40мм, який рухається з великою швидкістю та непередбачуваною траєкторією. Основні етапи реалізації включають завантаження відеопотоку, перетворення кольорового простору, визначення діапазону кольорів, створення бінарного зображення, морфологічні операції, знаходження контурів та відображення об'єкта.

3 РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ АЛГОРИТМУ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ ТА ВИКОНАННЯ СУДДІВСЬКИХ ПРОЦЕСІВ В НАСТІЛЬНОМУ ТЕНІСІ

3.1 Підбір технічного забезпечення

Система аналізу, відстеження м'яча та ведення рахунку в настільному тенісі, хоч і здатна працювати на майже будь яких комп'ютерах та ноутбуках, базується на використанні мікрокомп'ютера Raspberry Pi 4, що є центральним елементом усієї архітектури. Raspberry Pi 4 має систему на кристалі (SoC) BCM2711, яка включає в себе чотири ядра процесора ARM Cortex-A72 з тактовою частотою 1,5 ГГц, вироблену за 28-нанометровим технологічним процесом.

Завдяки підтримці стандартного ARM GIC, Raspberry Pi 4[32] (вартість ~100\$ набору) (Рис.4) здатний ефективно працювати з системами віртуалізації, що дало можливість серйозно розширити можливості використання даного обладнання. Об'єм оперативної пам'яті у 4 Гб LPDDR4 є цілком достатнім для використання сучасних додатків метою яких є обробка відео та відстеження об'єктів. Він має низьке енергоспоживання, що дозволяє навіть не вимикати його, але при цьому забезпечує достатню обчислювальну потужність та підтримку необхідного програмного забезпечення для обробки відеоданих в реальному часі. Всі ці фактори роблять його максимально ефективним для виконання даного завдання.

Система оснащена камерою для Raspberry Pi Camera Module 3, яка має наступні характеристики:

- Роздільна здатність: 11.9MP
- 12-мегапіксельний датчик зображення Sony IMX708 CMOS і оснащений системою зворотнього підсвічування
- Розмір пікселя становить 1.4x1.4 мкм.
- Розширення зображення: 4608 × 2592 пікселів.
- Кут огляду: 75 градусів

- Загальні режими відео: 480p120, 720p100, 1080p50(в системі використовується 480p120 так як в нашому випадку швидкодія буде грати більшу роль ніж якість зображення)

- Роз'єм кабелю: FPC 15 × 1 мм
- Вихід: RAW10
- Фазове автофокусування, що збільшує швидкість фокусу.
- Вбудована 2D динамічна корекція дефектів пікселів
- 2-провідне послідовне управління механізмом фокусування
- Розмір матриці: 7,4 мм
- 2-провідний послідовний зв'язок
- Послідовний вихід даних CSI-2
- Режим HDR (до 3 мегапікселів)

Додатково камера оснащується фокусною лінзою для налаштування фокусної відстані під конкретну ситуацію.

Вартість даної камери складає 25\$, що робить її досить доступною[32].



Рисунок 3.1 – Мікрокомп'ютер Raspberry Pi 4 з камерою [31]

Для забезпечення стабільного освітлення використовується лампа денного світла.

3.2 Підбір програмного забезпечення

Мікрокомп'ютер оснащений операційною системою Ubuntu MATE, котра представляє з себе версію Ubuntu розроблену спеціально для архітектури мікрокомп'ютера Raspberry PI.

Для реалізації даного проекту було обрано мову програмування Python, котра має велику кількість пакетів та засобів для підтримки технології комп'ютерного зору.

Використовується наступні пакети та бібліотеки:

- Collections[33]
- Numpy[34]
- Argparse[35]
- Imutils[36]
- cv2[37]
- time[38]
- pandas as pd[39]
- matplotlib[40]
- tkinter[41]
- PIL[42]

3.3 Проектування інтерфейсу

Інтерфейс програми забезпечує комфорту взаємодію користувача та програми. До нього відносять всі елементи які користувач здатен бачити та з якими здатен взаємодіяти. В програмі «електронний суддя для пінг-понгу буде використовуватись графічний інтерфейс.

Інтерфейс повинен відповідати таким вимогам:

- Забезпечити вивід рахунку під час раунду;
- Забезпечити вивід загального рахунку по раундах;

- Відображення обробленого відео, для того щоб користувач міг бачити; як працює програма, чи відстежується м'яч та яка була траєкторія удару;
- Інформація яка виводиться повинна мати комфортний для сприйняття вигляд;
- Кнопка для обнулення рахунку;
- Комфортне вимкнення програми;

Макет програми можна побачити на рисунку 3.2.

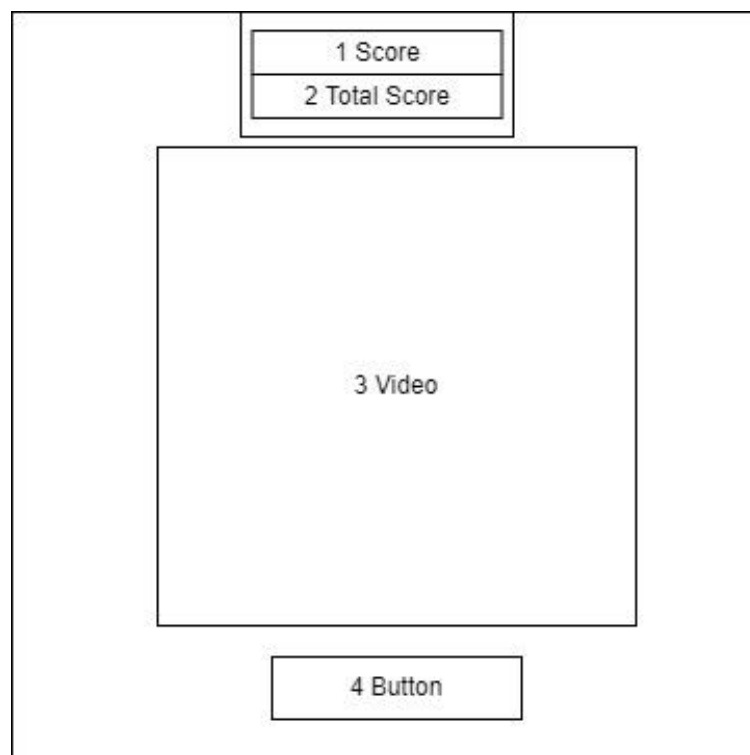


Рисунок 3.2 – Макет програми

На рисунку знаходяться:

- віджет для виводу рахунку(див. рис.3.2 блок 1);
- віджет для виводу загального рахунку пораундах(див. рис.3.2 блок 2);
- віджет для виведення обробленого відео(див. рис.3.2 блок 3);
- кнопка для обнулення результатів(див. рис.3.2 блок 4);

Для реалізації інтерфейсу було обрано модуль Tkinter, який має весь необхідний набір команд для виконання даного завдання, також для перетворення формату відео на той, який підтримується даним пакетом було використано пакет PIL. Вигляд вікна програми можна побачити на рисунку 3.3.

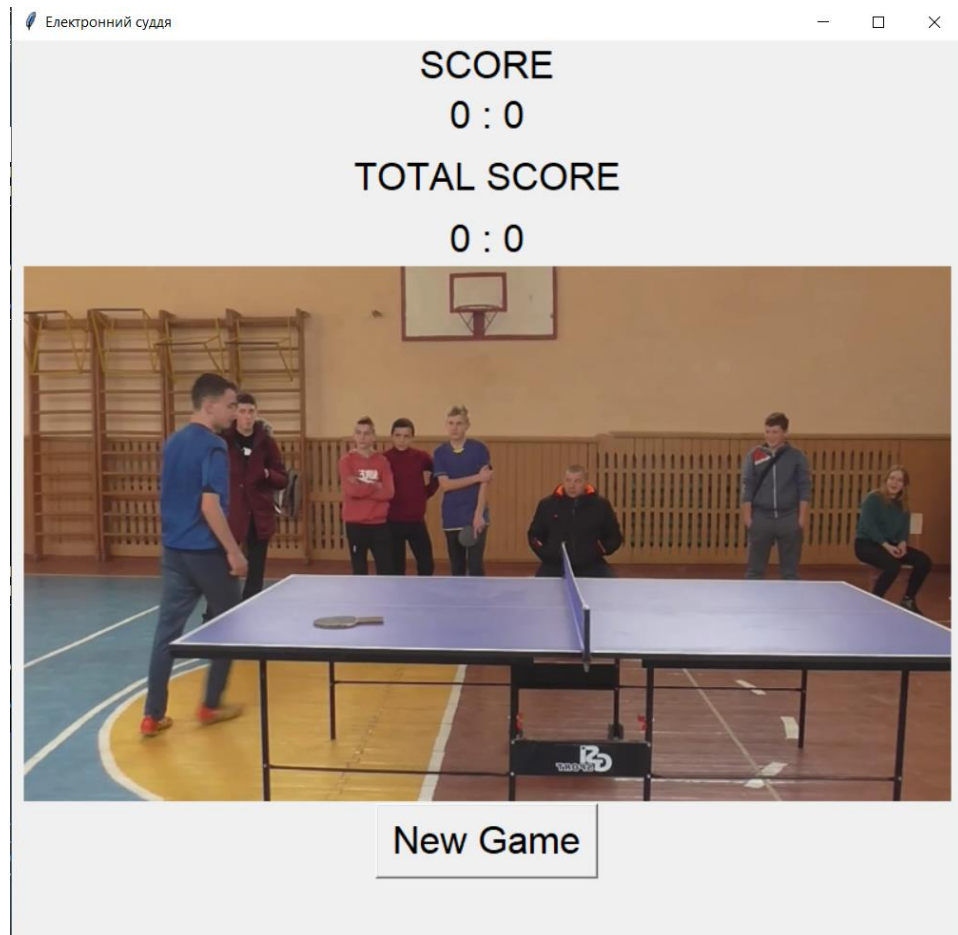


Рисунок 3.3 – Інтерфейс програми

Віджети для виведення рахунку, чи тексту створені за допомогою функції «Label» «pack», працює це наступним чином. Перший віджет виводить текст «Score», другий виводить переведені в текстовий формат змінні Player1, Player 2 які містять значення рахунку гравців, за замовчуванням 0:0, для оновлення рахунку існує спеціальна функцій «update_score», котра викликається при зміні рахунку, тобто при збільшенні значення змінної Player1, або Player2.

Далі знову знаходиться виводить текстовий віджет «Total Score», за яким слідує віджет з виведеннями змінних Player1total та Player2total.

Після них слідує віджет для виведення відео, який підлаштовується під розміри відео. Кожен фрейм відео виводиться як зображення, проте цикл програми виконується 60-80 разів за секунду, тому людина його бачить в хорошій якості 60-80 FPS.

Знизу знаходиться кнопка «New Game» яка при натисканні обнуляє рахунок гравців, тобто змінює значення змінних Player1, Player2, Player1total та Player1total на 0.

Шрифт обрано «Helvetica» нежирний, розмір 24.

Вгорі вікна програми є назва «Електронний суддя» та стандартні кнопки для керування вікном, які дозволяють згорнути та розгорнути його, відкривати на весь екран та закривати, при закритті вікна програма припиняє свою роботу звільняючи всі ресурси комп'ютера. Також саме вікно можна вільно збільшувати, зменшувати та переміщати на екрані.

3.4 Опис роботи системи

Камера встановлюється збоку від стола для настільного тенісу. Це розташування дозволяє камері знімати всю площину столу перпендикулярно до його поперечних сторін, що є критично важливим для сьогоденної версії реалізації алгоритму обрахунку. Проте як ми знаємо колір об'єкта залежить не тільки від нього, але й від кольору світла яке він відбиває, тому для забезпечення успішного і стабільного відстежування м'яча потрібне статичне освітлення, спеціально для цього поруч з камерою встановлюється лампа денного світла, котра компенсує змінне освітлення в приміщенні, забезпечуючи стабільне візуальне сприйняття ігрового поля системою.

В якості бази для системи виступає мікрокомп'ютер Raspberry Pi 4, до якого під'єднуються камера та монітор для виводу інформації, також, оскільки всі розрахунки виконуються на мікрокомп'ютері, то після оптимізації інтерфейсу система здатна працювати навіть з мобільними пристроями. Таким чином, на смартфоні будуть відображатися лише результати.

Система починає свою роботу з побудування вікна інтерфейсу та додавання віджетів для виводу отриманих в процесі гри результатів, таких як виведення обробленого зображення та рахунку і створення кнопки для обнулення результатів.

Потім слідує ініціалізація через пакет `Argparse`, який дозволяє обробляти командний рядок і передавати відео як аргумент. Це забезпечує гнучкість у виборі вхідного відеопотоку для аналізу. Після цього, за допомогою функції `VideoCapture` з пакету `cv2`, програма захоплює відео, яке потім обробляється для визначення точного часу кожного кадру використовується пакет `time`. Обробка включає перетворення розміру кадру, виконання розмиття та стискання з метою зменшити похибку, побудова маскт з бінарним зображенням на основі палітри HSV [20] (Рис.3), що є ключовим для наступного етапу виявлення м'яча.

Для виявлення м'яча в системі задається специфічний діапазон кольорів через функцію `inRange`, що дозволяє створити бінарне зображення тільки з областями, що відповідають кольору м'яча. Операції морфології, такі як згладжування та розширення, застосовуються для покращення якості знаходження контурів. Після виявлення м'яча, використовуючи функцію `cv2.findContours()`, система ідентифікує його контури та центр. М'яч на зображенні виділяється колом з позначенням центру червоною точкою для візуального відстеження.

Візуалізація процесу відстеження є виконується для кращої демонстрації роботи системи та надання можливості краще відстежувати траєкторію м'яча в спірних моментах, чи при аналізі гри (Рис.5). Між поточним і попереднім положенням м'яча малюється червона лінія, що дозволяє користувачеві чітко бачити траєкторію руху м'яча. Це забезпечує глибоке розуміння динаміки гри. Нарешті, програма завершується, звільняючи ресурси, використані під час обробки, для припинення роботи програми достатньо просто закрити вікно натиснувши на стандартну кнопку «хрести». Така архітектура програмного

забезпечення не лише покращує точність відстеження м'яча, але й забезпечує користувачеві інтуїтивне і зручне використання системи.



Рисунок 3.4 – Відображення результату

Алгоритм ведення рахунку в системі є ключовою частиною для точного відстеження результатів матчу. Хоча цей алгоритм все ще знаходиться в процесі розробки, його основний принцип полягає в аналізі траєкторії м'яча для визначення набраних очок. Коли м'яч рухається вздовж столу з права на ліво та перетинає визначену лінію на осі Y , правому гравцеві зараховується очко. Аналогічно, при русі з ліва на право, очки зараховуються лівому гравцеві.

Також для випадків коли м'яч залишається на столі, або падає не пересікаючи вище згадану лінію було додано функцію яка зараховує очко, якщо м'яч залишається на половині протилежного гравця довше двох секунд. Ці механізми дозволяють автоматично відстежувати рахунок матчу, знижуючи залежність від суддівських рішень та помилок.

Інтерфейс, наразі реалізовано у простій формі з використанням пакету Tkinter[22]. Цей інтерфейс виводить кількість очок на екрані разом із зображенням з камери.

Такий підхід до ведення рахунку та інтерфейсу має потенціал значно поліпшити досвід користувача, роблячи процес відстеження і аналізу настільного тенісу більш точним, зрозумілим і доступним. Це важливий крок до створення ефективної та надійної системи для автоматизації арбітражу в настільному тенісі.

3.5 Тестування

Перевірка роботи системи електронного суддівства для аматорського настільного тенісу заснована на комп'ютерному зорі була здійснена за допомогою аналізу десяти відеозаписів різних шкільних матчів. Ці записи були обрані для оцінювання здатності системи ефективно слідкувати за рухами м'яча, коректно фіксувати важливі події у грі та точно рахувати очки під час різноманітних ігрових ситуацій. Основна мета тестування полягала у виявленні можливих помилок системи, перевірці точності відстеження м'яча та оцінці надійності алгоритму підрахунку рахунку.

За результатами тестування середня точність відстеження м'яча для десяти матчів склала 97,8%, що підтверджує ефективність використаного алгоритму відстеження. Система здатна достатньо точно визначати положення м'яча навіть при швидких і раптових змінах траєкторії, які є типовими для настільного тенісу.

Далі було проведено тестування нашої системи на підрахунок результатів матчів системи з HSV палітрою(рис.6, табл. 1) та з RGB(рис.7, табл. 2)

3.5.1 Тестування системи з HSV палітрою

Тестування нашої системи автоматичного відстеження м'яча та рахунку в аматорському настільному тенісі показало хороші результати, які демонструють високу точність та надійність системи. У десяти різних тестах система достатньо точно відтворювала реальний рахунок гри, що свідчить про її непогану ефективність. Середня точність відстеження (табл.2) для лівого

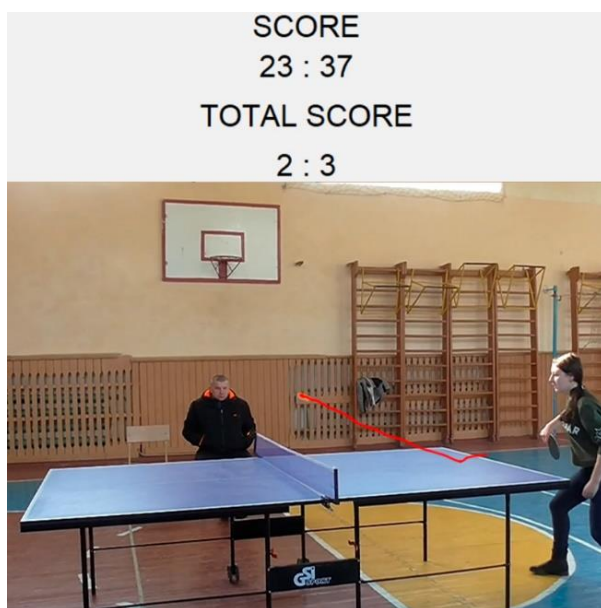
гравця склала 95.37%, а для правого гравця — 94.41%, зі середньою точністю по всіх матчах у 94.89%. Ці результати підтверджують хорошу здатність системи фіксувати удари та відповідний рахунок у реальному часі.



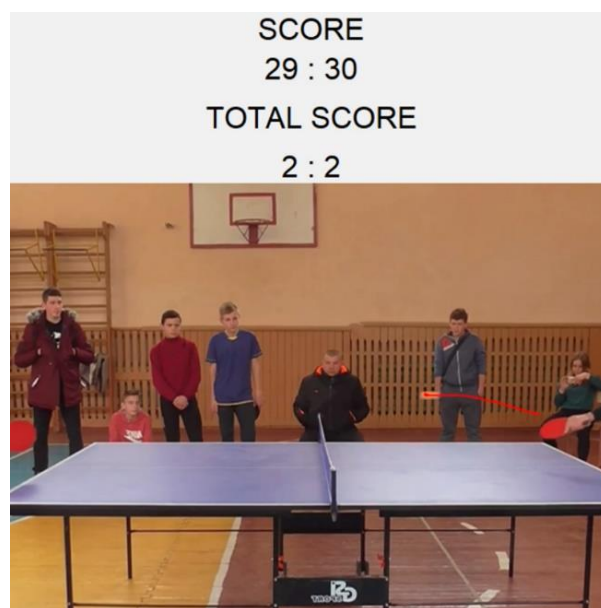
a) Тест 1



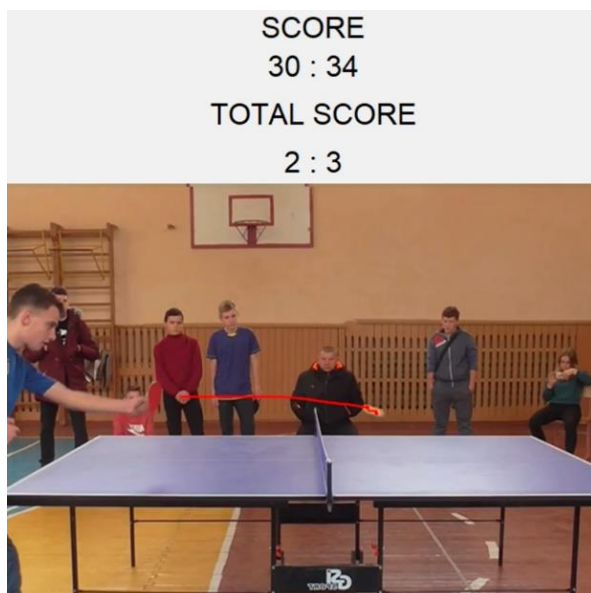
b) Тест 2



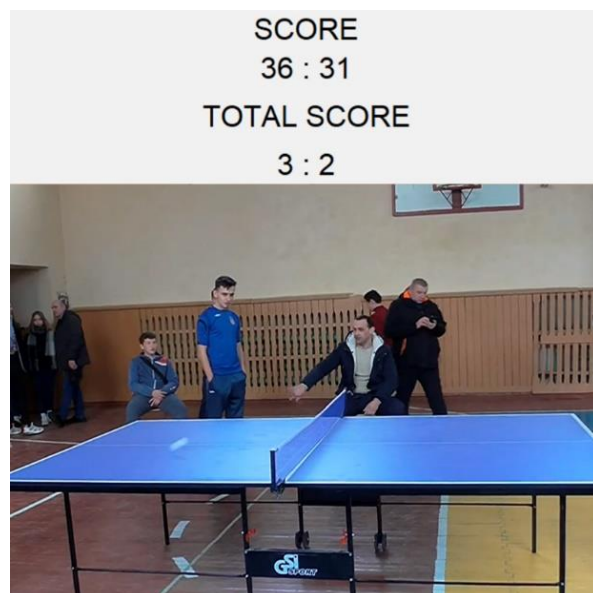
c) Тест 3



d) Тест 4



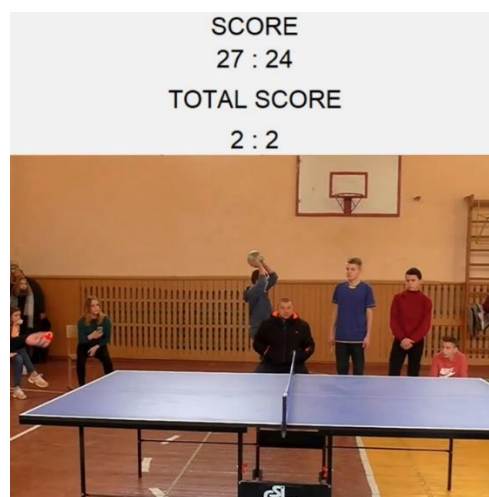
e) Тест 5



f) Тест 6



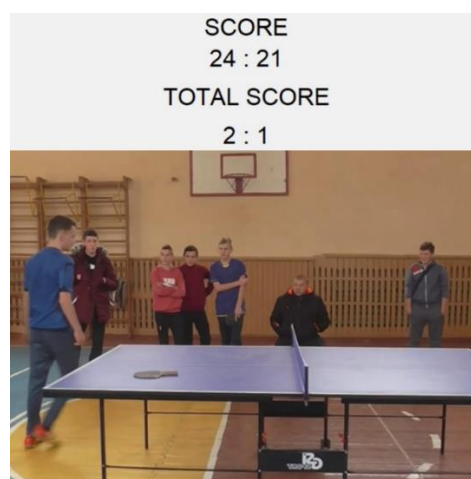
g) Тест 7



h) Тест 8



i) Тест 9



j) 10

Рисунок 3.6 – Знімки з результатом тестування

Найвища точність була зафіксована у п'ятому тесті (96.9%) (див рис.3.6е), демонструючи хорошу відповідність рахунку для лівого гравця та точну для правого. Це вказує на високу адаптивність системи до різних стилів гри та здатність точно відслідковувати швидкі та складні рухи м'яча.

У другому тесті (див рис. 3.6b), де була зафіксована найнижча точність у 89.68%, спостерігалися додаткові ускладнення через варіювання рівнів освітленості. Це було викликано змінними погодними умовами на вулиці, зокрема чергуванням сонячного світла та хмарності, що проникали через вікно в ігровий зал. Такі різкі зміни в освітленні, від дуже темного до пересвіченого середовища, створили значні виклики для алгоритму відстеження м'яча. Ці умови, з одного боку, ускладнювали розпізнавання м'яча через недостатність світла, а з іншого — через засліплення від занадто яскравого світла. Це вказує на необхідність оптимізації алгоритму для кращої адаптації до змінних умов освітлення, що є критично важливим для забезпечення стабільної та надійної роботи системи у різноманітних середовищах та умовах.

Таблиця 1. Результатів тестування

Тест	Реальний рахунок	Рахунок за системою	Точність для лівого гравця (%)	Точність для правого гравця (%)	Середня точність по матчу
1	35:22	32:19	91,14	86,36	88,75
2	34:29	30:21	88,23	72,41	80,32
3	27:38	23:37	85,18	97,36	91,27
4	30:34	29:30	96,67	88,23	92,45
5	29:35	30:34	96,67	97,14	96,9
6	38:30	34:30	89,47	100	94,73
7	30:33	29:32	96,67	96,97	96,82
8	31:29	27:24	87,09	82,75	84,92
9	27:34	25:31	92,59	91,17	91,87

10	26:25	24:21	92,3	84,0	88,15
Середня точність			95,37	94,41	90,6

Отже, отримані результати свідчать про значний потенціал нашої системи в контексті аматорського настільного тенісу, де надійність, точність та швидкість реакції системи є критично важливими для об'єктивного та справедливого ведення рахунку. Така висока ефективність системи відкриває можливості для її використання не тільки в аматорських іграх, але й у тренувальних сесіях, де точний аналіз руху м'яча може сприяти покращенню техніки гравців.

3.5.2 Тестування системи з RGB палітрою

Додатково з ціллю перевірки використання якої кольорової палітри RGB, чи HSV було проведено тестування для програми з використаннями палітри RGB. Результати якого показані на Таблиці та рисунку 3.7.

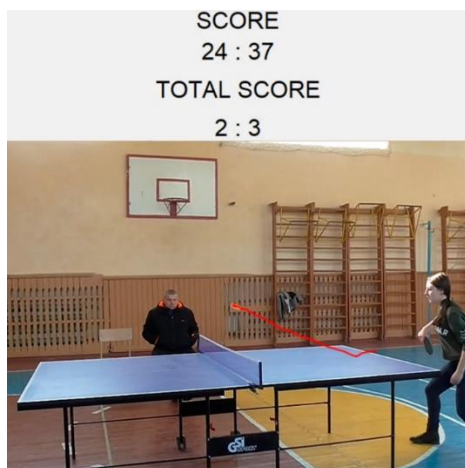
Загальна точність роботи системи трекінгу в процесі виконання даного тесту склала 95,4%.



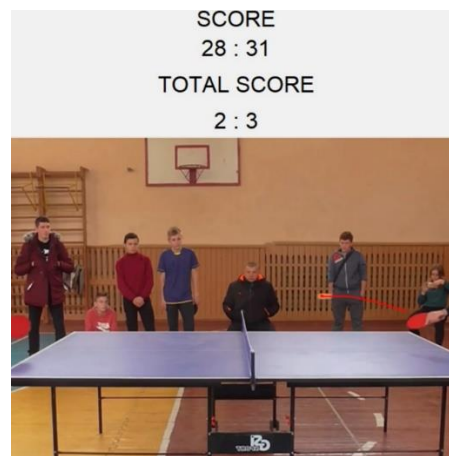
a) Тест 1



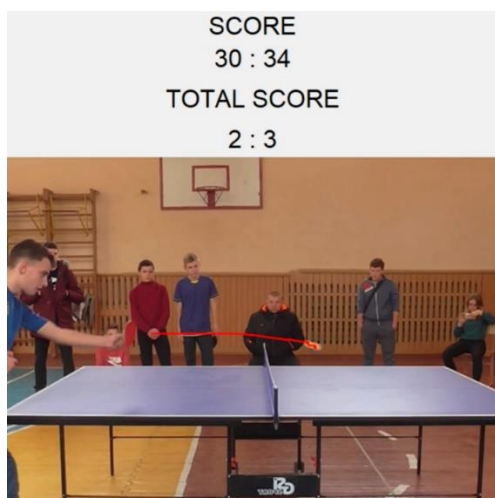
b) Тест 2



c) Tect 3



d) Tect 4



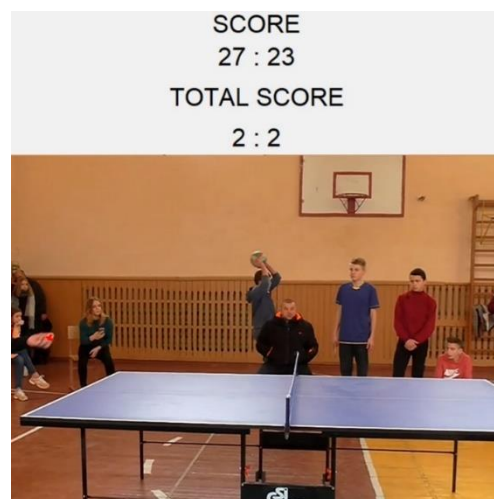
e) Tect 5



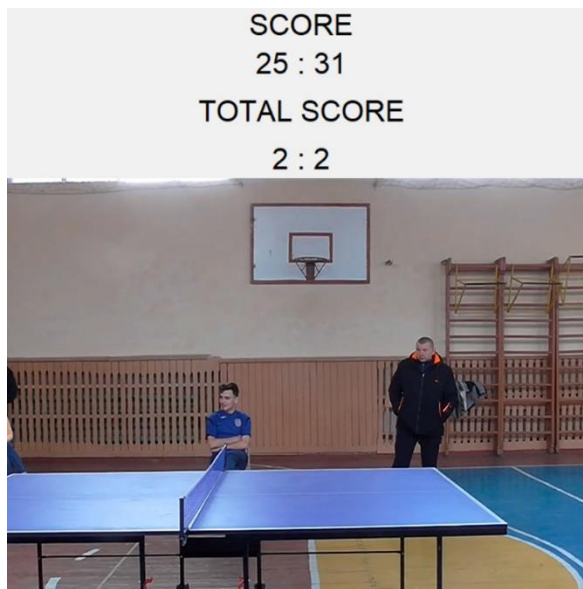
f) Tect 6



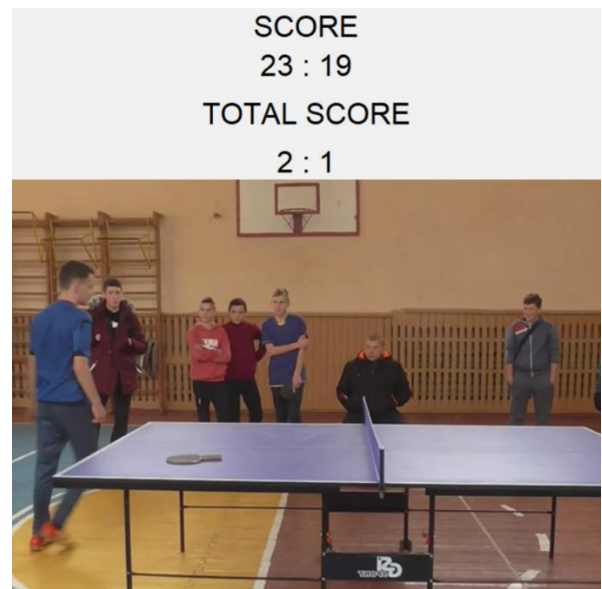
g) Tect 7



h) Tect 8



i) Тест 9



j) Тест 10

Рисунок 3.7 – Вікна результатів тестування з RGB палітрою

В процесі проведення даного тестування найвища точність для лівого гравця склала 96,67%, її було зафіксовано під час п'ятого та сьомого(див Таблиця2, рис.3.7e, 3.7g). Для правого гравця вона склала 100% та була зафіксована під час шостого тесту(див Таблиця2, рис.3.7f). Найвища середня точність становить 96,9%(див Таблиця2, рис.3.7e), її було зафіксовано під час п'ятого тесту.

Найнижча точність для правого гравця спостерігалась в другому тестуванні, вона становила 65.51%(див Таблиця2, рис.3.7b), а для лівого 87.09%, яка спостерігалась в тестуванні №8(див Таблиця2, рис.3.7h). Як ми можемо бачити знову під час другого тесту зафіксовані найнижчі показники точності, саме коли варіювалось освітлення. Це вже вдруге демонструє нам чутливість системи до освітлення та в цілому оточення. Даний недолік є досить суттєвим і потребує подальших доопрацювань.

Середня точність за підсумками всього тестування становить 88,7%(див Таблиця2, що є нижчим ніж результати системи з використанням HSV палітри та підтверджує твердження про те, що вона краще підходить для вирішення подібних завдань.

Таблиця 2. Результатів тестування

Тест	Реальний рахунок	Рахунок за системою	Точність для лівого гравця (%)	Точність для правого гравця (%)	Середня точність по матчу
1	35:22	31:20	88,57	90,09	89,33
2	34:29	31:19	91,17	65,51	78,34
3	27:38	24:37	88,89	97,36	93,125
4	30:34	28:31	93,33	91,17	92,25
5	29:35	30:34	96,67	97,14	96,9
6	38:30	34:30	89,47	100	94,735
7	30:33	29:31	96,67	93,39	95,03
8	31:29	27:23	87,09	79,31	83,195
9	27:34	25:31	92,59	91,17	91,88
10	26:25	23:19	88,46	76	82,23
Середня точність			95,37	94,41	88,7

3.6 Порівняння з аналогами

Серед найближчих досліджень (див.табл 1 та табл 3) у сфері розвитку технологій для настільного тенісу, дослідження, які вивчають використання комп'ютерного зору та інших технологій для відстеження руху м'яча та аналізу гри, є важливими для підвищення точності та об'єктивності арбітражу.

Дослідження Ning, Wang, Fu і Duan [11] використовує алгоритм для виявлення точки приземлення м'яча з точністю 78.5% та виявлення ключових кадрів з точністю 93.3%. Цей підхід залежить від обладнання Jetson Nano, що

коштує близько 290 доларів. Wong, Myint і Hopgood [12] використовують багатокамерну систему з багатоагентною системою, але не вказують точність свого підходу. Їх система вимагає чотирьох камер Casio EX-F1, що значно підвищує вартість до приблизно 1000 доларів. Sanusi, Di Mitri, Limbu і Klemke [20] пропонують систему для класифікації ударів, використовуючи мультимодальні дані та нейронні мережі з точністю 73% і запам'ятовуванням 61%, з використанням смартфона та датчиків руху Microsoft Kinect, загальна вартість яких складає близько 1230 доларів.

Таблиця 3. Порівняння технічних можливостей пропоновано системи з аналогами

Дослідження	Підхід	Точність	Потреба технічного забезпечення	Вартість обладнання
Дане дослідження	Електронна арбітражна система для аматорського настільного тенісу	Середня точність відстеження м'яча 97,8% Середня точність підрахунку балів 90,6%	Raspberry Pi 4, камера, освітлення	~125\$
Tao Ning, Changcheng Wang, Meng Fu, Xiaodong Duan [11]	Алгоритм виявлення точки приземлення	78.5% загальна точність, 93.3% виявлення ключових кадрів	Jetson Nano	~290\$
Patrick Wong, Hnin Myint, Adrian Hopgood [12]	Мульти-камера, багатоагентна система (MAS)	Не зазначено	4 шт Casio EX-F1 512 × 384 pixels	~\$1000 (4x \$250)

Khaleel Asyraaf Mat Sanusi, Daniele Di Mitri, Bibeg Limbu, Roland Klemke [20]	Класифікація ударів настільного тенісу за допомогою мультимодальних даних та нейронних мереж	Точність (73%) і запам'ятовування (61%)	Смартфон з датчиками руху Microsoft Kinect	~\$1 230
---	--	---	--	----------

На відміну від аналогів [11, 12, 20] (див.табл 2), наше дослідження, направлене на розробку електронної арбітражної системи для аматорського настільного тенісу, показує переваги з точки зору вартості, простоти використання та точності. З середньою точністю відстеження м'яча 97,8% та підрахунку балів 90,06% , дана система використовує доступний Raspberry Pi 4 та стандартну камеру з додатковим освітленням, загальна вартість яких складає лише приблизно 125 доларів. Це робить її значно доступнішою та простішою у використанні для аматорських гравців у порівнянні з іншими представленими технологіями, при цьому забезпечуючи високу точність і надійність.

Отже, електронна арбітражна система для аматорського настільного тенісу, розроблена з використанням комп'ютерного зору, продемонструвала високу ефективність, що підтверджується середньою точністю відстеження м'яча в 97,8% та точністю підрахунку балів у 94,89%. Завдяки використанню доступного мікрокомп'ютера Raspberry Pi 4 та стандартної відеокамери, система є економічно вигідною та простою у впровадженні, з загальною вартістю приблизно 125 доларів. Ця система не лише підвищує об'єктивність та точність суддівства у аматорських змаганнях, але й відкриває можливості для її використання у тренувальних сесіях, допомагаючи гравцям покращувати свою техніку гри.

Висновки до розділу 3:

В даному розділі було описано реалізацію та тестування системи «Електронний суддя для пінг-понгу».

Спочатку було здійснено підбір технічних та програмних засобів для виконання даного завдання. До технічних засобів відносяться мікрокомп'ютер Raspberry PI4, камера Sony IMX708 CMOS та лампа для освітлення. До програмних операційна система мікрокомп'ютера Ubuntu MATE, мова програмування Python та наступні пакети: collections, numpy, argparse, imutils, cv2, time, pandas, matplotlib, tkinter та PIL.

Наступним кроком було проектування, та створення інтерфейсу програми. Спочатку було сформовано вимоги до нього, потім побудовано макет, а після за допомогою пакету Tkinter і реалізовано сам інтерфейс в відповідності до вище згаданих вимог та макета.

Систему було реалізовано в двох варіантах, один використовував для пошуку кольорову палітру HSV, а інший RGB. За результатами якого отримно наступні результати:

- Точність системи трекінгу склала 97,8% для HSV палітри та 95,4% для RGB.
- Точність обрахунку склала 90,6% для HSV палітри та 88,6% для RGB.

З чого можна зробити такий висновок, що HSV палітра для виконання даного завдання підходить краще. Також в процесі тестування були виявлені певні недоліки роботи системи такі як фальш-голи та ситуації коли гол не зараховувався, що робить фактичну її точність трохи нижчою, проте дані аспекти ще будуть доробляться.

Потім було виконано порівняння з аналогічними системами, встановлено переваги та недоліки описаної системи.

Дана система є значно більш доступнішою та простою в використанні, що робить її перспективною для використання в аматорському спорті чи тренуваннях

ВИСНОВКИ

Під час виконання даної роботи було розглянуто тему "Створення програмного модулю електронний суддя для пінг-понгу".

Метою роботи було розроблення програмного модулю, який здатен автоматично визначати правильність виконання правил гри в пінг-понгу.

Під час роботи були досліджені основні правила гри в пінг-понгу та виявлені основні вимоги до електронного судді. Було проведено аналіз існуючих програмних рішень, що використовуються в спортивних змаганнях, та визначено їх переваги та недоліки.

У першому розділі було представлено всі необхідні поняття та термінологія, які є важливими для подальшого розуміння дипломної роботи. Викладені відомості охоплюють основні концепції комп'ютерного зору, його застосування в спорті, а також перспективи розвитку цієї галузі. Особлива увага приділялась машинному навчанню та його підрозділу - розпізнаванню образів та алгоритмам, пов'язаним з ними. Також було проведено аналіз відомих рішень по даній темі. Досліджено поточну ситуацію з використанням комп'ютерного зору в спорті в цілому та виконано аналіз досліджень по інтегруванню комп'ютерного зору в конкретні види спорту такі як, безпосередньо, настільний теніс, баскетбол та бейсбол. Також було досліджено комерційні системи на основі технології комп'ютерного зору, а саме «Hawk-Eye», «SportVu», «Sport Vision». Варто виділити дослідження «Robotic Table Tennis with Model-Free Reinforcement Learning» де вдалось відносно стабільно прогнозувати траєкторію м'ячика.

У другому розділі було виконано розробку методу побудови системи з виконання електронного відстеження та ведення суддівських процесів в настільному тенісі на основі методу відстеження об'єкта за допомогою пошуку кольорової плями. Також було здійснено розробку архітектури системи та створено алгоритм її роботи, варто зауважати, що алгоритм суддівських процесів потребує подальшого доопрацювання.

У третьому розділі було проведено підбір технічних та програмних засобів, розглянуто проектування та реалізацію інтерфейсу програми, а потім і саму реалізацію системи трекінгу та ведення рахунку. Після чого було проведено тестування двох варіантів програми з HSV та RGB палітрами для відстеження м'яча. Точність системи трекінгу склала 97,8% та 95,4% для систем з HSV та RGB палітрами, відповідно. Точність обрахунку для першої, очікувано, теж виявилась вищою, вона склала 90,6% проти 88,6%.

Також в процесі тестування було виявлено такі недоліки як недосконалість алгоритму ведення рахунку, саме в плані відстеження головних ситуацій, та чутливість системи трекінгу до зміни освітлення.

Накінець було проведено порівняння даної системи з аналогічними та встановлено такі основні переваги як доступність та простота використання.

Тож за результатами дослідження можна підвести такий підсумок, що дана система має певні перспективи з модернізації та застосування в аматорських турнірах та тренуваннях з метою покращення якості як арбітражних процесів так і оптимізації процесу тренування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. James E. Dobson (2023). The Birth of Computer Vision. University of Minnesota Press. ISBN 978-1-5179-1421-9.
2. Thomas, G., Gade, R., Moeslund, T. B., Carr, P., & Hilton, A. (2017). Computer vision for sports: Current applications and research topics. Computer Vision and Image Understanding.
3. O.Yu. Sergiyenko, V.V. Tyrsa. 3D optical machine vision sensors with intelligent data management for robotic swarm navigation improvement, IEEE Sensors Journal 2021.
4. Nilsson, Nils (1998). Artificial Intelligence: A New Synthesis. Morgan Kaufmann. ISBN 978-1-55860-467-4. Archived from the original on 26 July 2020. Retrieved 18 November 2019.
5. Nils J. Nilsson, Introduction to Machine Learning Archived at the Wayback Machine(2019).
6. David J. C. MacKay. Information Theory, Inference, and Learning Algorithms Archived (2016).
7. Isabelle Guyon Clopinet, André Elisseeff (2017). An Introduction to Variable and Feature Selection.
8. Gerdes, J. Christian; Kegelmann, John C.; Kapania, Nitin R.; Brown, Matthew; Spielberg, Nathan A. (2019-03-27). "Neural network vehicle models for high-performance automated driving".
9. Інтернет ресурс - <https://www.superannotate.com>. (Переглянуто 13 червня, 2023).
10. Zhang, Z.Z.Z., Xu, D.X. De & Tan, M.T.M., 2010. Visual Measurement and Prediction of Ball Trajectory for Table Tennis Robot. Ieee Transactions On Instrumentation And Measurement.
11. Ruiz, N.A., 2010. Tracking of a Basketball Using Multiple Cameras.
12. Chakraborty, B. & Meher, S., 2018. 2D Trajectory-based Position Estimation and Tracking of the Ball in a Basketball Video.

- 13.Theobalt, C., Albrecht, I. & Magnor, M., 2004. Pitching a Baseball — Tracking High-Speed Motion with Multi-Exposure Images.
- 14.Itseez, 2022. OpenCV | OpenCV. Available at: <http://opencv.org/> [Електронний ресурс]. - Режим доступу: [Відвідано 12 червня 2023].
- 15.Hawk-Eye Innovations Ltd. Hawk-Eye// : <http://www.hawkeyeinnovations.co.uk/> (переглянуто 12 червня 2023).
16. SportVU, SportVU. Stats LLC. Available at: <https://www.statsperform.com/> (Перглянуто 13 червня 2023).
17. SportVision Inc.. Available at: <http://www.sportvision.com/> (Переглянуто 14 червня 2023).
- 18.Opta Sports Available at: <https://www.statsperform.com/opta/>
- 19.Zebra MotionWorks, Zebra Technologies/ Available at: <https://www.zebra.com/>(Переглянуто 8 вересня 2023).
- 20.Kinexon. Available at: <https://kinexon.com/> (Переглянуто 9 вересня 2023).
- 21.Ning, T., Wang, C., Fu, M. et al. A study on table tennis landing point detection algorithm based on spatial domain information. Sci Rep 13, 20656 (2023). <https://doi.org/10.1038/s41598-023-42966-6>
- 22.Wong P, Myint H, Dooley L, Hopgood A. A multi-view automatic table tennis umpiring framework. Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology. 2023;0(0). doi:10.1177/17543371231171433
- 23.Rocha, N.M.S., Pinto, M.F., Biundini, I.Z. et al. Analysis of tennis games using TrackNet-based neural network and applying morphological operations to the match videos. SIViP 17, 1133–1141 (2023). <https://doi.org/10.1007/s11760-022-02320-1>
- 24.A. J. Moshayedi, Z. Chen, L. Liao and S. Li, "Kinect Based Virtual Referee For Table Tennis Game: TTV (Table Tennis Var System)," 2019 6th International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 2019, pp. 354-359, doi: 10.1109/ICISCE48695.2019.00077.

- 25.H. Myint, P. Wong, L. Dooley and A. Hopgood, "Tracking a table tennis ball for umpiring purposes," 2015 14th IAPR International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 2015, pp. 170-173, doi: 10.1109/MVA.2015.7153160.
- 26.Kamble, P.R., Keskar, A.G. & Bhurchandi, K.M. Ball tracking in sports: a survey. *Artif Intell Rev* 52, 1655–1705 (2019). <https://doi.org/10.1007/s10462-017-9582-2>
- 27.Hiemann A, Kautz T, Zottmann T, Hlawitschka M. Enhancement of Speed and Accuracy Trade-Off for Sports Ball Detection in Videos—Finding Fast Moving, Small Objects in Real Time. *Sensors*. 2021; 21(9):3214. <https://doi.org/10.3390/s21093214>
- 28.Javadiha, M.; Andujar, C.; Lacasa, E.; Ric, A.; Susin, A. Estimating Player Positions from Padel High-Angle Videos: Accuracy Comparison of Recent Computer Vision Methods. *Sensors* 2021, 21, 3368. <https://doi.org/10.3390/s21103368>
- 29.He, C., Zhang, X., Gui, Y., Liu, Y., & Zhang, W. (2021). Mathematical modeling and simulation of table tennis trajectory based on digital video image processing. *Advances in Mathematical Physics*, 2021, 1-11.
- 30.Mat Sanusi, K.A.; Mitri, D.D.; Limbu, B.; Klemke, R. Table Tennis Tutor: Forehand Strokes Classification Based on Multimodal Data and Neural Networks. *Sensors* 2021, 21, 3121. <https://doi.org/10.3390/s21093121>
- 31.Ambre, S., Masurekar, M., & Gaikwad, S. (2020). Face recognition using raspberry pi. *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI*, 1-11.
- 32.Raspberry Pi // [raspberrypi.com](https://www.raspberrypi.com) [Электронный ресурс]. - Режим доступа: <https://www.raspberrypi.com/>
33. Collections // docs.python.org [Электронный ресурс]. - Режим доступа: <https://docs.python.org/3/library/collections.html>
- 34.Numpy // numpy.org [Электронный ресурс]. - Режим доступа: <https://numpy.org/doc/>

- 35.Argparse // python.org [Електронний ресурс]. - Режим доступу:
<https://docs.python.org/3/library/argparse.html>
- 36.Imutils // pypi.org [Електронний ресурс]. - Режим доступу:
<https://pypi.org/project/imutils/>
- 37.OpenCV // opencv [Електронний ресурс]. - Режим доступу:org
<https://opencv.org/>
- 38.Time // python.org [Електронний ресурс]. - Режим доступу:
<https://docs.python.org/3/library/time.html>
- 39.Pandas // pydata.org [Електронний ресурс]. - Режим доступу:<https://pandas.pydata.org/docs/>
- 40.Matplotlib // matplotlib.org [Електронний ресурс]. - Режим доступу:<https://matplotlib.org/stable/users/index.html>
- 41.Tkinter // python.org [Електронний ресурс]. - Режим доступу:
<https://docs.python.org/3/library/tkinter.html>
- 42.PIL // pypi.org Електронний ресурс]. - Режим доступу:
<https://pypi.org/project/Pillow/>
- 43.Загальні рекомендації з підготовки, оформлення, захисту та оцінювання випускних кваліфікаційних робіт здобувачів вищої освіти першого «бакалаврського» і другого «магістерського» рівнів / За ред. доц. М.І. Шинкарика. Тернопіль: ТНЕУ, 2018. 67 с.
- 44.Комар М.П., Саченко А.О., Васильків Н.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за другим (магістерським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2021. 32 с
- 45.Івасечко А.В., Осолінський О.Р. «Відстеження об'єкта методом пошуку кольорової плями на мові python», Тренди та перспективи розвитку мультидисциплінарних досліджень: матеріали IV Міжнародної студентської наукової конференції, м. Луцьк, 1 грудня, 2023 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2023. - 306-309 сс.

ДОДАТОК А
КОД ПРОГРАМИ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ ТА ВИКОНАННЯ
СУДДІВСЬКИХ ПРОЦЕСІВ В НАСТІЛЬНОМУ ТЕНІСІ

```
from collections import deque
import numpy as np
import argparse
import imutils
import cv2
import time
import pandas as pd
import matplotlib.pyplot as plt
import tkinter as tk
from tkinter.font import Font
from PIL import ImageTk, Image

window = tk.Tk()
window.title("Електронний суддя")

window.geometry("800x800")

monitor = tk.Label(window)

Player1 = 0
Player2 = 0
Player1total = 0
Player2total = 0

def process_video():

    ap = argparse.ArgumentParser()
    ap.add_argument("-v", "--video",
                    help="path to the (optional) video file")
    ap.add_argument("-b", "--buffer", type=int, default=64,
                    help="max buffer size")
    args = vars(ap.parse_args())
    global Player1, Player2, Player1total, Player2total
    hMin = 14
    sMin = 14
    vMin = 100
    hMax = 255
    sMax = 255
    vMax = 255

    #lower = (hMin, sMin, vMin)
    #upper = (hMax, sMax, vMax)
    Lower = (200, 200, 200)
    Upper = (150, 150, 150)
    GLower = (15, 70, 220)
    GUpper = (30, 190, 255)
    pts = deque(maxlen=args["buffer"])
```

```

if not args.get("video", False):
    camera = cv2.VideoCapture(0)
#else:
    # camera = cv2.VideoCapture(args["video"])

Data_Features = ['x', 'y', 'time']
Data_Points = pd.DataFrame(data = None, columns =
Data_Features , dtype = float)

netzone=50
imageHeight = 800
x_netzone = imageHeight * float(netzone)/100

start = time.time()

# Player1score = Player1()
# Player2score = Player2()
goaltime = 0
player2goal = False
GoalScored = False

while True:
    # grab the current frame
    (grabbed, frame) = camera.read()

    current_time = time.time() - start

    # if args.get("video") and not grabbed:
    #     break

    frame = imutils.resize(frame, width=800,)
    frame = cv2.flip(frame,1)
    blurred = cv2.GaussianBlur(frame, (13, 13), 0)
    hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv, GLower, GUpper)
    mask = cv2.erode(mask, None, iterations=2)
    mask = cv2.dilate(mask, None, iterations=2)

    cnts = cv2.findContours(mask.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)
    center = None

    if len(cnts) > 0:

        c = max(cnts, key=cv2.contourArea)
        ((x, y), radius) = cv2.minEnclosingCircle(c)
        M = cv2.moments(c)
        center = (int(M["m10"] / M["m00"]), int(M["m01"] /
M["m00"]))

        if (radius < 10) & (radius > 3 ) :

```

```

        cv2.circle(frame, (int(x), int(y)), int(radius),
            (0, 255, 255), 1)
        cv2.circle(frame, center, 5, (0, 0, 255), -1)

        Data_Points.loc[Data_Points.size/3] = [x , y,
current_time]

        pts.appendleft(center)

        for i in range(1, len(pts)):

            thickness = int(np.sqrt(args["buffer"] / float(i +
1)) * 1) # малює лінію
            cv2.line(frame, pts[i - 1], pts[i], (0, 0, 255),
thickness)

            if pts[i - 1] is None or pts[i] is None:
                continue

        processed_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        processed_image = Image.fromarray(processed_frame)
        processed_image = ImageTk.PhotoImage(processed_image)

        monitor.config(image=processed_image)
        monitor.image = processed_image
        window.update_idletasks()
        window.update()
        for i in range(1, len(pts)):
            if pts[i] is not None and pts[i - 1] is not None:
                if pts[i] > pts[i - 1] and x > 750:
                    Player2 += 1
                    update_label2()
                    x = 0
                    time.sleep(2)
                    pass
                if pts[i] < pts[i - 1] and x < 50 and x != 0:
                    Player1 += 1
                    update_label2()
                    #print("Правий гравець :", Player1, "Лівий
гравець :", Player2)
                    x = 0
                    time.sleep(2)
                    pass
            else:
                continue

        if Player1 == 12 and Player1 - Player2 >= 2 or
Player1 == 7 and Player2 == 0:
            Player1total +=1
            update_label4()

```

```

        if Player2 == 12 and Player2 - Player1 >= 2 or Player2
== 7 and Player1 == 0:
            Player2total +=1
            update_label4()

    key = cv2.waitKey(10) & 0xFF
    camera.release()
    cv2.destroyAllWindows()

text_font = Font(family="Helvetica", size=24)

label1 = tk.Label(window, text=f"SCORE", font= text_font)

label2 = tk.Label(window, text=f"23 : 19", font=text_font)
def update_label2():
    label2.config(text=f"0 : 0", font=text_font)

label3 = tk.Label(window, text=f"TOTAL SCORE", font=text_font)

button = tk.Button(window, text="New Game",font= text_font )

label4 = tk.Label(window, text=f"2 : 1",
font=text_font){Player1} : {Player2}
def update_label4():
    label4.config(text=f"{Player1total} : {Player2total}")

label1.pack(ipady = 0, ipadx=200, anchor="n")
label2.pack(ipady = 0, ipadx=200, anchor="n")
label3.pack(ipady = 10, ipadx=150, anchor="n")
label4.pack(ipady = 0, ipadx=150, anchor="n")
monitor.pack(padx=10, pady = 0, side="top")
button.pack()

process_video()

window.mainloop()

```

ДОДАТОК Б
АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

МАТЕРІАЛИ IV МІЖНАРОДНОЇ
СТУДЕНТСЬКОЇ НАУКОВОЇ
КОНФЕРЕНЦІЇ

ТРЕНДИ ТА ПЕРСПЕКТИВИ
РОЗВИТКУ МУЛЬТИДИСЦИП-
ЛІНАРНИХ ДОСЛІДЖЕНЬ



М. ЛУЦЬК, УКРАЇНА

**1 ГРУДНЯ
2023 РІК**

МАТЕРІАЛИ IV МІЖНАРОДНОЇ
СТУДЕНТСЬКОЇ НАУКОВОЇ
КОНФЕРЕНЦІЇ

.....
**ТРЕНДИ ТА ПЕРСПЕКТИВИ
РОЗВИТКУ МУЛЬТИДИСЦИП-
ЛІНАРНИХ ДОСЛІДЖЕНЬ**
.....

м. Луцьк, Україна
1 грудня 2023 рік

Вінниця, Україна
«UKRLOGOS Group»
2023

СЕКЦІЯ 17.**КОМП'ЮТЕРНА ТА ПРОГРАМНА ІНЖЕНЕРІЯ**

АНАЛІЗ ОСОБЛИВОСТЕЙ ВІЗУАЛІЗАЦІЇ АГРАРНИХ ПОКАЗНИКІВ ЗА ДОПОМОГОЮ ЦИФРОВИХ КАРТ Кедрун В.О., Науковий керівник: Полягушко Л.Г.	288
АВТОМАТИЗАЦІЯ ПЕРЕКЛАДУ ЖЕСТОВОЇ МОВИ Веселовська І.М., Науковий керівник: Казимир І.Я.	292
ГРАФОВІ ПІДХОДИ ДО МОДЕЛЮВАННЯ ТА АНАЛІЗУ СПІЛЬНОТ В ІНТЕРНЕТІ: РОЛЬ ТА ЗАСТОСУВАННЯ СОЦІАЛЬНОЇ МЕРЕЖНОЇ АНАЛІТИКИ (SNA) Адамович О.Ю., Науковий керівник: Веретук С.О.	294
ОСОБЛИВОСТІ АРХІТЕКТУРИ JAVA Шибко Д.О., Науковий керівник: Шибко О.М.	297
ПРАКТИЧНА РЕАЛІЗАЦІЯ СУЧАСНИХ ПІДХОДІВ У СТВОРЕННІ ВІДЕОКОНТЕНТУ Гусаров Є.І., Науковий керівник: Ковальов Ю.Г.	300

СЕКЦІЯ 18.**ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ**

АЛГОРИТМ ВИМІРЮВАННЯ ЧАСУ ОБСЛУГОВУВАННЯ ДЛЯ SSD НАКОПИЧУВАЧІВ Іваночко В.Р., Науковий керівник: Осолінський О.Р.	303
ВІДСТЕЖЕННЯ ОБ'ЄКТА МЕТОДОМ ПОШУКУ КОЛЬОРОВОЇ ПЛЯМИ НА МОВІ PYTHON Івасечко А.В., Науковий керівник: Осолінський О.Р.	306
ДОСЛІДЖЕННЯ ВІДКРИТИХ SMS ДЛЯ СТВОРЕННЯ ОСВІТНИХ ОНЛАЙН-РЕСУРСІВ Волківська В.О., Науковий керівник: Устенко С.В.	310
ДОСЛІДЖЕННЯ МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ Фарафонов Г.Р., Науковий керівник: Євсєєв С.П.	313
ІННОВАЦІЇ ІУС ДЛЯ БАНКІВСЬКОЇ ДІЯЛЬНОСТІ Базюк І.В., Науковий керівник: Устенко С.В.	315
ІНФОРМАЦІЙНА ПІДСИСТЕМА З ПІДТРИМКИ ДІЯЛЬНОСТІ ОХОРОННОЇ ОРГАНІЗАЦІЇ Карпина І.С., Науковий керівник: Устенко С.В.	318
КОМП'ЮТЕРИЗАЦІЯ ПРОЦЕСІВ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ УПРАВЛІННІ ЦІННИМИ ПАПЕРАМИ Селіверстов П.С., Науковий керівник: Устенко С.В.	322
МЕТОД ОЦІНКИ ІНВЕСТИЦІЙНИХ РИЗИКІВ ВІРТУАЛЬНОЇ ІТ-КОМПАНІЇ НА ОСНОВІ МАШИНОГО НАВЧАННЯ Дем'янова В., Науковий керівник: Лип'яніна-Гончаренко Х.В.	325

Івасечко Андрій Вікторович, магістр спеціальності 122 “Комп’ютерні науки”
Західноукраїнський національний університет, Україна

Науковий керівник: Осолінський Олександр Романович, канд. техн. наук,
доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

ВІДСТЕЖЕННЯ ОБ’ЄКТА МЕТОДОМ ПОШУКУ КОЛЬОРОВОЇ ПЛЯМИ НА МОВІ PYTHON

Вступ

Виявлення та відстеження м’ячів або кулястих об’єктів за допомогою обробки зображень для спортивних подій з використанням Python – на сьогоднішній день є популярною задачею для дослідників, які займаються розробкою таких систем на базі недорогих мікрокомп’ютерів [1].

В більшості випадків використовується камера [2] для зйомки відеозаписів м’яча в реальному часі, які потім обробляються за допомогою бібліотеки Python OpenCV [3]. Така популярність пов’язана з універсальністю ПО, яке може легко портуватись на різні комп’ютерні системи. Але перед тим як почати відслідковувати м’яч, потрібно правильно знайти кольорові плями (тенісний м’яч теж відноситься до цієї категорії) певної кольорової гами.

В установці з однією камерою відстеження об’єктів відбувається шляхом віднімання фону з подальшим встановленням відповідності регіону, що враховує різні ознаки, такі як швидкість об’єкта, його розміри та відстань до обмежувальної рамки[4]-[5].

Пропонований метод

Суть пропонованого методу полягає в пошуку кольорової плями заданого кольору, або діапазону кольорів, яким повинен відповідати колір відстежуваного об’єкта, фіксованого, або максимального розміру, конкретної форми, або просто плями.

В даному випадку відстежуваним об’єктом є м’яч для пінг-понгу [6], який представляє собою кулю діаметром 40мм оранжевого кольору, котра рухається на великій швидкості (до 30м/с) та з непередбачуваною траєкторією і великою кількістю відскоків.

В реалізації даного методу можна виділити такі ключові етапи:

- Завантаження відеопотоку
- Перетворення кольорового простору з палітри Read Green Blue(RGB) в Hue Saturation Value(HSV).
- Визначення діапазону кольорів шляхом задавання максимального(upper) та мінімального(lower) значень.

Для виконання даного завдання було обрано діапазон характерний для відтінків оранжевого м’ячика.

– Створення бінарного зображення, яке створюється за допомогою функції «cv2.inRange()», для того щоб відобразити тільки області, що відповідають діапазону кольорів:

- Застосування операцій морфології. Для покращення якості знаходження

контурів та об'єктів можна застосовувати операції морфології, такі, як згладжування, розширення або звуження.

- Знаходження контурів в бінарному зображенні за допомогою функції «cv2.findContours()».

- Виділення об'єкта. В даній реалізації він виділяється колом, а його центр червоною точкою.

А. Завантаження потоку

Однією з ключових функцій завантаження потоку є можливість роботи з зображеннями в різних форматах, таких як JPEG, PNG, BMP тощо. Також, OpenCV дозволяє відкривати та обробляти відеопотоки з вебкамер або відеофайлів, виконувати операції з обробки та аналізу кадрів у реальному часі.

Пакет також надає засоби для виявлення об'єктів, розпізнавання паттернів, класифікації зображень та аналізу відеопотоків.

Процес захоплення відеопотоку починається зі створення об'єкту захоплення за допомогою виклику методу `cv2.VideoCapture()`, де вибирається джерело відеопотоку. Для того щоб захопити відео з вбудованої камери, вказується її номер у функції. Наприклад, `cap = cv2.VideoCapture(0)` захопить відео з першої доступної камери. Потім виконується читання кадрів за допомогою функції «`read()`» об'єкта `VideoCapture`, котра дозволяє отримати новий кадр з відеопотоку. Цей метод повертає два значення: перше - `ret`, яке показує, чи був успішно прочитаний кадр, а друге - `frame`, що містить сам кадр.

Б. Створення бінарного зображення

Створення бінарного зображення полягає у перетворенні кольорового чи чорно-білого зображення на зображення, де пікселі приймають тільки два значення: чорний або білий, з метою виділення або відокремлення певних областей на основі заданих порогових значень яскравості чи насиченості, в нашому випадку даному діапазону повинні відповідати колір та відтінки оранжевого тенісного м'ячика, він має наступні значення: Hue мінімальне = 25, Saturation мінімальне = 25, Value мінімальне = 125, Hue максимальне = 50, Saturation максимальне 50, та Value максимальне 255.

Використовується дане перетворення за допомогою функції пакета Cv2 «`cv2.inRange()`».

В. Застосування операцій морфології

Для покращення якості знаходження контурів та об'єктів можна застосовувати операції морфології, такі як згладжування, та розширювання, які застосовуються для виокремлення областей інтересу, видалення шуму, відновлення об'єктів чи для покращення якості зображень. Їх застосування дозволяє підготувати дані для подальшого використання у програмах комп'ютерного зору, що позитивно впливає на якість обробки.

Згладжування (Erosion) - це операція яка зменшує розмір об'єктів на зображенні шляхом видалення зовнішніх пікселів по контуру об'єкта. Це використовується для видалення шуму, створення границь об'єктів, або зменшення товщини контуру. Виконується за допомогою функції «`cv2.erode()`».

Розширення (Dilation): На відміну від ерозії, дана операція збільшує розмір об'єктів на зображенні шляхом додавання пікселів до контуру об'єкта. Вона корисна для заповнення малих отворів, об'єднання об'єктів та збільшення товщини контуру.

Йому відповідає функція «cv2.dilate()».

Знаходження контурів в бінарному зображенні.

Після попередніх підготувань можна безпосередньо задати параметри для пошуку об'єкта (тенісний м'ячик) та виконати його. Знаходження контурів в бінарному зображенні проводиться за допомогою функції «cv2.findContours()», де задаються параметри форми шуканого об'єкта – коло, це виконується за допомогою функції «cv2.circle()». З метою відсіювання дрібних точок, або об'єктів задаються мінімальний та максимальний радіус об'єкта, цикл повторюється до тієї пори поки його не буде знайдено, або коли роботу програми не перерве користувач.

Г. Виділення об'єкта.

Коли об'єкт знайдено потрібно це візуалізувати для користувача, для цього виконується виділення. Периметр обводиться жовтим колом, а його геометричний центр червоною точкою. Так як тенісний м'яч рухається зі швидкістю до 40 м/с з нелінійною траєкторією то людському оку встежити за ним досить важко, тож, додатково виконується візуалізація траєкторії руху: між кількома останніми точками розташування центру малюється червона лінія, за допомогою функції «cv2.line()». Фінальним етапом є виведення отриманого зображення на екран за допомогою функції «cv2.imshow()», що дозволяє подивитись, як працює програма рисунок 1.

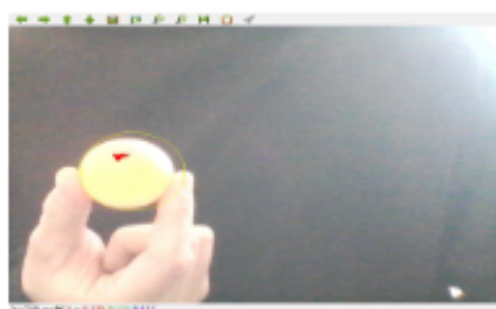


Рис. 1. Отримане зображення

Висновки

Сучасний підхід до відстеження м'ячів або кулястих об'єктів в спортивних заходах, використовуючи обробку зображень та програмування з використанням Python, що представляє значний інтерес для дослідників і практиків. Особливо актуальним стає використання недорогих мікрокомп'ютерів для створення таких систем, що сприяє поширенню та доступності цієї технології.

Однак перед початком відстеження м'яча, важливо правильно визначити кольорові плями або області, які відповідають м'ячу (наприклад, тенісний м'яч), що вимагає певних навичок та підготовки.

Пропонований метод полягає у виявленні заданої кольорової плями або діапазону кольорів, що відповідає м'ячу. В цій реалізації застосовується пошук оранжевого тенісного м'яча діаметром 40мм, який рухається з великою швидкістю та непередбачуваною траєкторією. Основні етапи реалізації включають завантаження відеопотоку, перетворення кольорового простору, визначення діапазону кольорів, створення бінарного зображення, морфологічні операції, знаходження контурів та відображення об'єкта.

Список використаних джерел:

1. KEČA, Dominik, et al. Ball Detection Using Deep Learning Implemented on an Educational Robot Based on Raspberry Pi. *Sensors*, 2023, 23.8: 4071.
2. Raspberry Pi High Quality Camera, <https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/>.
3. ALSHAMMARI, Basim, et al. Eye Movement Tracking Using Opencv Python. *Wasit Journal of Engineering Sciences*, 2023, 11.2: 71-81.
4. Pavithra G., Dr. T.C.Manjunath, Rajanish N., "Design & Development of a novel 4-point minimal pick & place trajectory in robotics", 10th Annual KSTA National Conference, Decennial Celebration of Science & Technology for future of Humanity (Sponsored by IIA, DRSC, ISRO, KSCST, Start up Karnataka, KSTePS) Decennial Celebration of Science & Technology for future of Humanity, Dept. of Science & Technology, Govt. of Karnataka, Reva University, Kattigenahalli, Bangalore-64, Karnataka, ISBN 978-81-936187-4-5, Paper id ES-12, Sl. No. 235, pg. 149, 18-19 Jan. 2018.
5. KAMBLE, Paresh R.; KESKAR, Avinash G.; BHURCHANDI, Kishor M. A deep learning ball tracking system in soccer videos. *Opto-Electronics Review*, 2019, 27.1: 58-69.
6. KAMBLE, Paresh R.; KESKAR, Avinash G.; BHURCHANDI, Kishor M. Ball tracking in sports: a survey. *Artificial Intelligence Review*, 2019, 52: 1655-1705.

Андрій Івасечко
Західноукраїнський національний університет
Олександр Осолінський
Західноукраїнський національний університет
ORCID ID: 0000-0002-0136-395X

Система "Електронний суддя для пінг-понгу" на базі мікрокомп'ютера Raspberry Pi

Abstract. В даній роботі представлено дослідження застосування систем комп'ютерного зору для спортивних подій, а саме «електронних суддів». Було проведено аналіз відомих систем для суддівства. Виходячи з аналізу аналогів запропоновано архітектуру системи «електронний суддя для пінг-понгу», описано апаратну складову, розроблено алгоритм роботи системи та програмне забезпечення та проведено тестування системи.

Ключові слова: комп'ютерний зір, електронний суддя, Raspberry Pi, пінг-понг, теніс, OpenCV

1 ВСТУП

В сучасному світі розвиток програмного забезпечення та технологій штучного інтелекту набуває все більшого значення в різних галузях життя. Спортивні ігри не є винятком, адже вони є важливою частиною нашого дозвілля, фізичного розвитку та здорового способу життя. Не оминуло це і пінг-понгу - популярної спортивної гри, яка вимагає точності, швидкості та реакції. Однак, під час гри виникають ситуації, коли необхідно об'єктивно визначити правильність удару, враховувати правила гри та забезпечити чесність результатів. Традиційно це роблять судді, проте їх наявність не завжди можлива або ефективна. Програмний модуль "Електронний суддя для пінг-понгу" повинен якщо не замінити суддю то допомогти йому, він може використовуватись у спортивних залах, на турнірах та тренуваннях з пінг-понгу для автоматичного ведення підрахунку результатів, що забезпечує більш об'єктивну гру та уникнення конфліктів між гравцями. Крім того, розроблений модуль може бути використаний як основа для подальшого розвитку та впровадження подібних систем у інші спортивні ігри, сприяючи автоматизації та покращенню суддівських процесів[1].

2. Аналіз існуючих систем на базі комп'ютерного зору для спорту

На сьогоднішній день алгоритми комп'ютерного зору (CV) широко використовуються для відстеження гравців, транспортних засобів і/або м'яча в різних

видах спорту. Це зазвичай використовується для підтримки рішень суддів і задоволення запитів команд або гравців щодо перегляду і зміни прийнятих рішень суддів або арбітрів. При огляді літератури щодо резюме та спортивних технологій, необхідно коротко оцінити також системи, призначені для комерційного використання. Хоча деталі використовуваних алгоритмів часто є конфіденційними з питань інтелектуальної власності, наявність таких систем надає деякі вказівки щодо використовуваного обладнання та програмного забезпечення, що можуть бути корисними в цій галузі досліджень. Вимоги до таких систем також вказують на очікування щодо рішень, заснованих на CV, що застосовуються у спорті[2][3].

Більшість комерційних систем CV для спорту зосереджені на підтримці рішень суддів, аналізі ефективності або анотаціях для трансляцій. Це програми не просто виявляють м'яч, але також генерують форму об'єкта, його розмір, розташування та достатню кількість даних для аналізу руху. Це важливі аспекти для застосувань у сфері аналізу ефективності, автоматичного анотування, коучингу та підтримки рішень. Нижче наведено загальний опис основних комерційних рішень щодо резюме, які стосуються даної роботи[4].

2.1 Технологія «Hawk-Eye»

«Hawk-Eye» [5] - це система на основі комп'ютерного зору (CV), яка була розроблена з 1999 року (Hawk-Eye Innovations Ltd, 2013) і пізніше придбана компанією Sony у 2011 році. Спочатку вона була розроблена для крикету як система суддівства, але зараз вона використовується в різних видах спорту, включаючи теніс, футбол, снукер і недавно легку атлетику. У всіх цих видах спорту «Hawk-Eye» вимагає високошвидкісні камери, які будуть правильно налаштовані (наприклад, шість камер у крикеті, сім у футболі), розміщені навколо арени, поля або корта.

У випадку з тенісом, який по свої суті дуже близький до настільного, обробка вхідних даних в системі «Hawk-Eye» складається з чотирьох етапів:

- 1) 2D-інформація з кожної камери використовується для обробки CV з метою виявлення центру м'яча,
- 2) 3D-дані обчислюються за допомогою триангуляції з усіх доступних камер,

3) ці 3D-дані обробляються для отримання точної трекінгової інформації про рух м'яча[5].

2.2 Технологія «SportVU»

Система розроблена компанією Stats LLC [6] у Чикаго і призначена для відстеження гравців та прогнозування 3D-відеоданих. Це рішення для аналізу, відоме як SportVU (2012), зосереджується на відстеженні гравців і створенні статистики гри. Його застосовують для вимірювання відстані, швидкості та відношення (відстань) до м'яча в різних видах спорту, включаючи футбол, баскетбол і американський футбол. Серед його клієнтів є ЗМІ, трансляційні компанії та організації з коучингу.

Існують два варіанти впровадження системи: базовий і розширений. У базовому варіанті встановлюють три невеликі високошвидкісні камери в одному місці, розташовані на декілька сантиметрів одна від одної. Програмне забезпечення ідентифікує об'єкти в полі зору та отримує дані про їх 3D-позиціонування, створюючи потік рухів кожного об'єкта в реальному часі і карту активності матчів. Розширене рішення передбачає використання шести високошвидкісних камер, розташованих у двох місцях у залі, як триплети. Камери можуть бути стаціонарними або портативними, кожна з них фіксує третину поля, і всі шість камер об'єднуються для створення панорамного огляду. Відстеження м'яча здійснюється з частотою 25 кадрів на секунду (FPS), а відстеження гравців - з частотою 15 FPS. Четвертий датчик використовується для виявлення нових об'єктів, таких як заміни гравців. Оператор системи повинен початково ідентифікувати гравців, судді та інших учасників.

Наразі SportVU не застосовується у виді спорту, такому як настільний теніс, оскільки швидкість гри та розміри поля в цих видів спорту значно відрізняються.

2.3 «Sport Vision»

Спортивна компанія SportVision (SportVision Inc., 2012) [7] розробила низку інструментів і методів для відстеження об'єктів у різних видах спорту, включаючи бейсбол, американський футбол, автоспорт NASCAR і конні перегони. Вони надають реально-часові показники та покращені графічні зображення для подій у прямому ефірі.

Спочатку їхніми клієнтами були переважно медіа-компанії, але востаннє споживачі також отримують доступ до цих технологій через портативні пристрої.

Один з продуктів SportVision, відомий як RPTCH/fx, використовується у бейсболі для вимірювання швидкості кидка, обертання м'яча, швидкості та висоти руху м'яча в 60 точках під час його польоту. Для цього використовуються три високошвидкісні камери, розташовані на кожному стадіоні. Ця технологія застосовується для трансляційної накладної графіки та для тренувань, дозволяючи виміряти середню швидкість м'яча з точністю до 2,5 см, що дорівнює приблизно 149.749 км/год. Хоча швидкість та обертання м'яча в бейсболі мають деякі схожі характеристики з настільним тенісом, ця технологія не використовується в настільному тенісі.

2.4 Opta Sports[8] - це одна з провідних компаній у галузі спортивної аналітики та статистики. Вони спеціалізуються на зборі, аналізі та наданні деталізованої статистики для різних видів спорту. Opta надає важливі дані для команд, тренерів, аналітиків, ЗМІ та спортивних фанатів з метою розуміння та аналізу спортивних подій.

Основні характеристики Opta Sports включають:

1. Збір статистики в реальному часі: Opta використовує високоточні технології, такі як оптичне відстеження та системи GPS, щоб відстежувати рух гравців та рух м'яча під час спортивних подій в реальному часі. Це дозволяє надавати докладну статистику та аналіз навіть під час гри.
2. Різні види спорту: Opta працює в різних видів спорту, включаючи футбол (соккер), регбі, крикет, баскетбол, хокей, американський футбол та інші. Вони надають адаптовану статистику для кожного виду спорту.
3. Деталізована статистика: Opta збирає широкий спектр статистичних даних, включаючи інформацію про голи, передачі, штрафи, кидки, різноманітні показники професіоналізму гравців, володіння м'ячем, атаки, оборону та багато іншого.

4. Використання в спортивному аналізі та бізнесі: Інформація, надана Opta, корисна для тренувального процесу команд, стратегічного аналізу, а також для створення спортивних програм, ставок і багатьох інших галузей.
5. Статистика в реальному часі та передача даних: Opta надає послуги статистики в реальному часі для ЗМІ та онлайн-платформ, а також надає API для інтеграції своїх даних в різні додатки та веб-сайти.

Opta Sports є ключовим гравцем у сфері спортивної аналітики і важливим джерелом даних для багатьох фахівців і фанатів спорту, що вивчають і аналізують виступи команд і гравців.

2.5 Zebra MotionWorks[9] - це технологічна платформа для відстеження руху гравців та обладнання в спортивних іграх, зокрема в американському футболі, американському футболі для відкритого повітря, регбі, хокеї на траві, гандболі, лакросі та інших колективних видів спорту. За допомогою системи Zebra MotionWorks можна отримувати високоточну інформацію про рух гравців і м'яча на полі, що дозволяє командам, тренерам і аналітикам отримувати цінні дані для стратегічного аналізу та покращення виступів.

Основні характеристики та можливості Zebra MotionWorks включають:

1. Висока точність вимірювань: Система використовує високоточні сенсори та технології відстеження для вимірювання руху гравців та обладнання з великою точністю. Це дозволяє збирати докладні дані про рух, швидкість, прискорення та інші параметри.
2. Система в реальному часі: Zebra MotionWorks надає дані в режимі реального часу, що дозволяє тренерам та аналітикам слідкувати за подіями на полі під час гри і приймати швидкі стратегічні рішення.
3. Аналіз даних: Платформа включає в себе інструменти для аналізу зібраних даних. Це дозволяє тренерам і аналітикам вивчати рух гравців, визначати стратегії та піддавати оцінці ефективність команди.

4. Інтеграція з іншими системами: Zebra MotionWorks може інтегруватися з іншими системами аналізу та відеоспостереження для отримання комплексної інформації про гру.
5. Використання в тренуваннях і підготовці: Дані, зібрані системою, можуть бути використані для покращення тренувального процесу, збільшення фізичної підготовки гравців та розробки стратегій.
6. Оптимізація стратегій: Збирати дані з реального полі гри дозволяє тренерам розробляти більш ефективні гравчині тактики та стратегії.

Zebra MotionWorks допомагає командам і спортивним організаціям покращити виступи гравців і підвищити рівень конкурентоспроможності в різних видах спорту.

2.6 Kinexon[10] - це компанія, яка розробляє інноваційні системи відстеження руху та аналітики для спортивних команд, тренерів та аналітиків. Вони спеціалізуються на використанні сучасних технологій для збору даних про рух гравців та обладнання на полі, що дозволяє зрозуміти і поліпшити спортивні виступи.

Основні характеристики та можливості Kinexon включають:

1. Висока точність вимірювань: Системи Kinexon використовують високоточні сенсори та технології відстеження, такі як радіочастотна ідентифікація (RFID) та ультразвук, для вимірювання руху гравців, м'ячів та іншого обладнання з великою точністю.
2. Реальний час: Дані, зібрані системою Kinexon, доступні в режимі реального часу, що дозволяє тренерам та аналітикам відстежувати гру в реальному часі та реагувати на події на полі.
3. Аналіз даних: Компанія надає інструменти для аналізу зібраних даних, включаючи інформацію про рух, швидкість, прискорення та інші параметри. Це дозволяє тренерам і аналітикам розробляти стратегії та оптимізовувати тренувальний процес.

4. Інтеграція з іншими системами: Системи Kinecon можуть легко інтегруватися з іншими системами аналізу даних та відеоспостереження для отримання комплексної інформації про гру.
5. Використання в різних видах спорту: Kinecon використовується в різних видів спорту, включаючи футбол, баскетбол, хокей, американський футбол, регбі, бейсбол та інші. Вони працюють з командами і спортивними організаціями по всьому світу.
6. Оптимізація тренувань і підготовки: Дані, зібрані системою Kinecon, можуть бути використані для покращення фізичної підготовки гравців та розробки ефективних тренувальних програм.

Дана система допомагає спортивним командам і організаціям отримати докладні дані про гру та гравців, що дозволяє покращити стратегії, виступи та знизити ризик травм.

3. Пропонована система “електронний суддя для пінг понгу”

На рисунку 3.1 ви можете бачити схему архітектури системи.

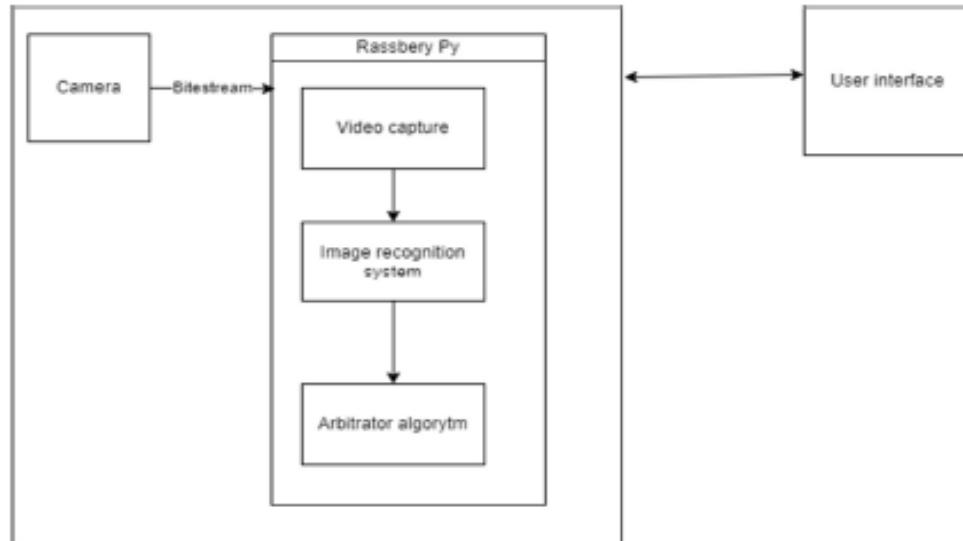


Рисунок 3.1 – Узагальнена архітектура системи

Система реалізовується на базі мікрокомп'ютері Raspberry Pi 4[11] з камерою(рисунок 3.2). Він має наступні характеристики: основа системи на кристалі (SoC) BCM2711 з чотирма ядрами процесора ARM Cortex-A72, тактова частота 1,5 ГГц, і виготовлена за технологічним процесом 28 нм. Додатково, пристрій підтримує стандартний ARM GIC (Generic Interrupt Controller), що сприяє швидкій роботі систем віртуалізації. ОЗУ становить 4 Гб стандарту LPDDR4.

Камера з мікрокомп'ютером закріплені зверху над столом, поряд з лампою, яка повинна нівелювати змінне освітлення, збоку є телевізор на який виводиться отримане зображення.



Рисунок 3.2 – Мікрокомп'ютер Raspberry Pi 4 з камерою

Програмний модуль створюється на мові Python та має наступний вигляд:

Пакет Argparse надає інструменти для обробки командного рядка і аргументів, що передаються програмі під час її виклику. Необхідний для надання програмі доступу до камери.

Для відстеження м'ячика використовується метод пошуку кольорової плями за допомогою OpenCV (cv2) в Python. Він має такі пункти:

- Завантаження відеопотоку

- Перетворення кольорового простору з палітри Read Green Blue(RGB) в Hue Saturation Value(HSV), яку ви можете побачити на рисунку 3.3 для кращого виділення кольорових областей:

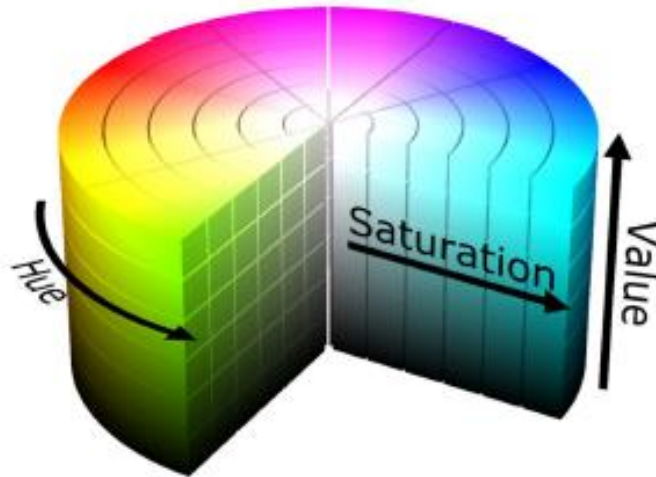


Рисунок 3.3 – Палітра HSV

- Визначення діапазону кольорів шляхом задавання максимального(upper) та мінімального(lower) значень. Для виконання даного завдання було обрано діапазон характерний для віддітків оранжевого м'ячика.
- Створення бінарного зображення. Яке створюється за допомогою функції «cv2.inRange()», для того щоб відобразити тільки області, які відповідають діапазону кольорів:
- Застосування операцій морфології. Для покращення якості знаходження контурів та об'єктів можна застосовувати операції морфології, такі як згладжування, розширення, або звуження.
- Знаходження контурів в бінарному зображенні за допомогою функції «cv2.findContours()».
- Виділення об'єкта. В даному випадку він виділяється колом, а його центр червоною точкою.
- Відображення результату в вигляді виводу на екран отриманого зображення з виділеними об'єктами, приклад якого ви можете побачити на рисунку 3.4.

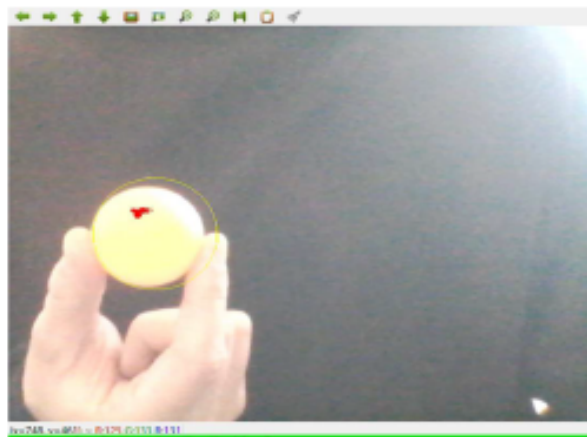


Рисунок 3.4 – Відображення результату

- Фінальним етапом є завершення циклу та звільнення ресурсів.

Алгоритм ведення рахунку ще в процесі реалізації. Але його принцип роботи є таким: якщо м'ячик рухається вздовж столу, тобто по осі x справа на ліво та пересікає певну лінію по осі y то правому гравцю зараховується очко, при русі з ліва на право, тобто коли цей гравець виконує подачу гол йому зараховуватись не буде, аналогічно і правий гравець. При наборі гравця 11 очок, або 7 очок, при умові, що його суперник не набрав жодного йому зараховуватиметься перемога.

Інтерфейс на даний момент має примітивну реалізацію в вигляді табла зробленого за допомогою пакету Turtle, яке виводить кількість очок та екрану з зображенням. В кінцевому результаті він повинен відображатись в одному вікні та мати такі кнопки як "Start game", "Stop" та "Exit".

3.1 Алгоритм роботи системи

Спочатку задаються зінні з рахунком «Player1», «Player2» (див. рис.2 блок 1) для першого та другого гравців, відповідно., далі за допомогою пакету Tkinter створюється вікно програми(див. рис.2, блок 2), до якого додаються віджети для виводу обробленого відеопотоку та рахунку(див. рис.2 блок 3), виводиться рахунок(див рис.2 блок 4), за замовчуванням, «0:0».

Тепер оголошується функція «process_video()»(див. рис2 блок 5), яка відповідає за трекінг та арбітражні пропеси. За допомогою пакету argparse створюється парсер

аргументів(див. рис.2 блок 6), програма яка захоплює відео в якості аргументу для функції. Здійснюється парсинг аргументів(див. рис.2 блок 7) в якості яких буде виступати захоплене відео отримане за допомогою функції "VideoCapture" пакету CV2.

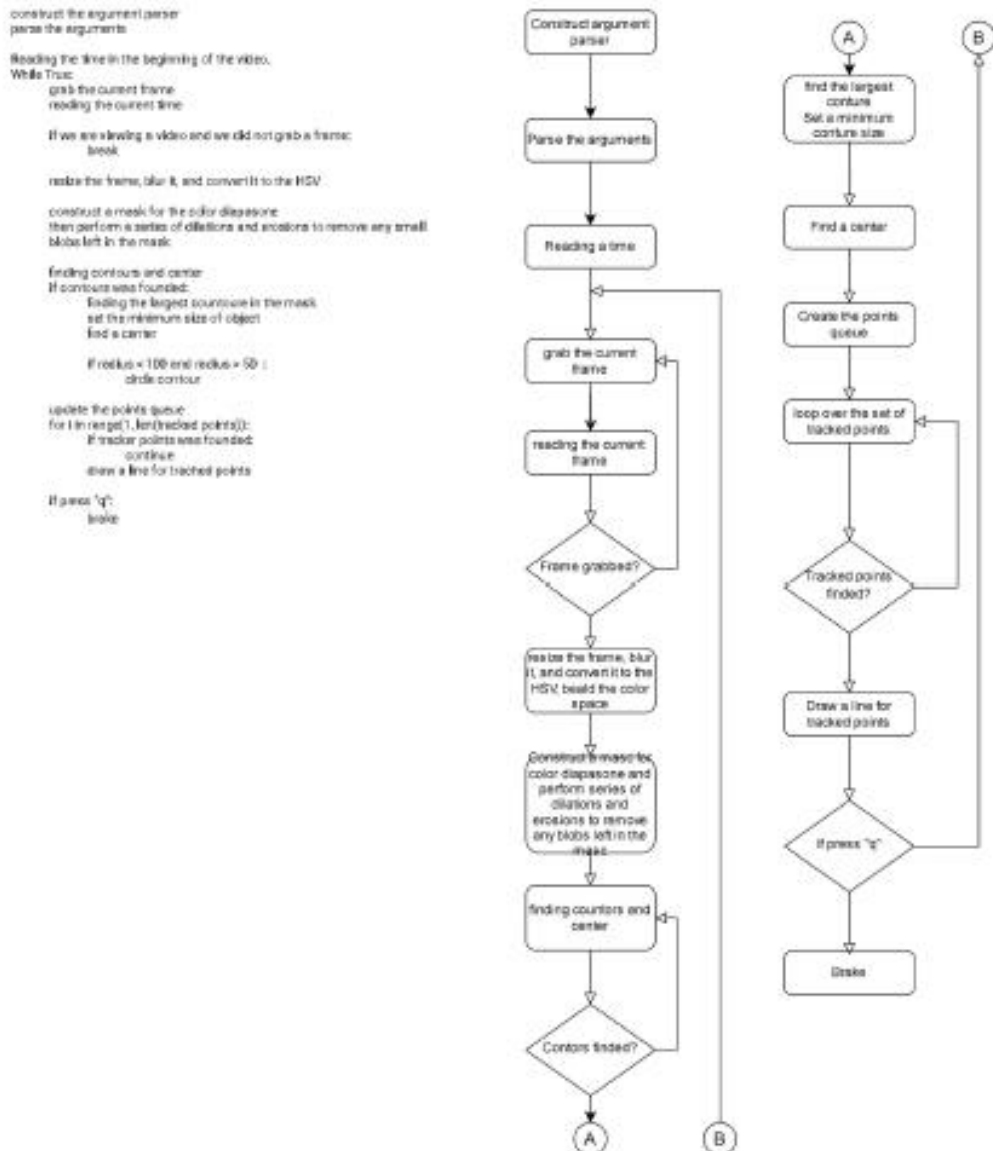
За допомогою функції "time" здійснюється читання коректного часу(див. рис.2 блок 8)

Далі відбувається оголошується цикл While(див. рис.2 блок 9), захоплення фрейму з прочитаного камерою зображення(див. рис.2 блок 10), починається відлік часу(див. рис.2 блок 11). Виконується перевірка, чи вдалось захопити кадр(див. рис.2 блок 12), якщо ні то спроба захопити фрейм повторюються. (див. рис.2 блок 13), якщо все ж зображення захоплюється то здійснюється перетворення його розміру, розмиття та перевід в палітру HSV та створюється простір кольорів(див. рис.2 блок 13) за допомогою функцій. "resize", "GaussianBlur" та "cvtColor", відповідно.

Наступний крок - конструювання маски(див. рис.2 блок 14) для заданого діапазону кольорів за допомогою функції "inRange" з такими значеннями lower = (25,25,125), upper = (50,50,255). Виконується серія ерозій і розширень, за допомогою відповідних функцій "erode" та "dilate" з ціллю прибрати будь які плями, які залишились на масці.

Далі здійснюється пошук кольорової плями(див. рис.2 блок 15) та перевірка(див. рис.2 блок 16), якщо вони не знаходяться то м'ячика немає на столі і функція повторюється доти поки він не повернеться в гру, коли це відбувається здійснюється пошук найбільшої плями в масці(див. рис.2 блок 17), знаходяться контури м'яча, та центр, виконується підсвітка червоним кольором(див. рис.2 блок 18).

Задається функція для набору "tracked points"(див. рис.2 блок 19). Здійснюється відстеження точок знаходження центру м'ячика(див. рис.2 блок 20), тоді відбувається перевірка, якщо вони не знаходяться то м'ячик не на столі і функція повторюється доти поки він не повернеться(див. рис.2 блок 21) в гру, коли це стається то між точкою положення м'яча на даному фреймі та аналогічною точкою попереднього фрейму малюється червона лінія, що дозволяє краще розуміти його траєкторію(див. рис.2 блок 22).



На рисунку 3.5 представлено алгоритм роботи системи.

Рисунок 3.5 - Алгоритм роботи програми

Висновки

В даній роботі було представлено систему "Електронний суддя для пінг-понгу" який дозволяє більш точно та об'єктивно судити спортивні змагання з настільного тенісу, яка б допомагала суддям у винесенні більш точних висновків. В даній роботі було проаналізовано існуючі аналоги "електронних судей" для різних видів спорту, а також засобів та методів обробки зображень, які вони використовують для відслідковування траєкторії руху м'яча. Проведений аналіз показав, що наявні системи використовують складні та дорогі системи, які недосяжні для змагань в любительських лігах, і взагалі дуже мало уваги зосереджено до настільного тенісу. Так як більшість спортивних змагань в Україні по пінг-понгу проводяться в любительському дивізіоні (школи, університети, міжрегіональні змагання) було запропоновано дешеву та продуктивну архітектуру систем ведення рахунку. Удосконалено метод детекції тенісного м'яча та поведінки системи при веденні рахунку по очках та по іграх.

В даний момент система перебуває в режимі тестування, налагодження та оптимізації методів розпізнавання тенісного м'яча під час руху по певній траєкторії. Пілотна версія здатна вести підрахунок голів, які зараховуються при пересіченні м'ячика уявної смуги поперек осі x, за умови, що воно зростає, або спадає. В подальших дослідженнях автори планують розширити список правил "детекції голів", можливості оптимізації методу розпізнавання м'яча з урахуванням апаратної складності системи, підвищення швидкодії та реакції на події при яких зараховуються очки.

Література

1. James E. Dobson (2023). The Birth of Computer Vision. University of Minnesota Press. ISBN 978-1-5179-1421-9.

2. Thomas, G., Gade, R., Moeslund, T. B., Carr, P., & Hilton, A. (2017). Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*.
3. O.Yu. Sergiyenko, V.V. Tyrsa. 3D optical machine vision sensors with intelligent data management for robotic swarm navigation improvement, *IEEE Sensors Journal* 2021.
4. Інтернет ресурс - <https://www.superannotate.com> (Переглянуто 10 липня 2023 року).
5. Hawk-Eye Innovations Ltd. Hawk-Eye. Available at: <http://www.hawkeyeinnovations.co.uk/> (переглянуто 12 липня 2023).
6. SportVU, SportVU. Stats LLC. Available at: <https://www.statsperform.com/> (Перглянуто 13 серпня 2023).
7. SportVision Inc.. Available at: <http://www.sportvision.com/> (Переглянуто 14 серпня 2023).
8. Opta Sports, Opta data from Stats Perform/ Available at: <https://www.statsperform.com/opta/>(Переглянуто 08 вересня 2023).
9. Zebra MotionWorks, Zebra Technologies/ Available at: <https://www.zebra.com/>(Переглянуто 8 вересня 2023).
10. Kinexon. Available at: <https://kinexon.com/> (Переглянуто 9 вересня 2023).
11. Rassberi ry. Available at: <https://www.raspberypuri.com/>. (Переглянуто 10 вересня, 2023).