

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

15 грудня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-наукової програми «Інформатика»
на тему: «Інформаційна система розпізнавання рухів у карате»
здобувача групи ІН.м-25 Чайки Богдана Вікторовича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Богдан ЧАЙКА
(підпис)

Керівник,
ст. викл., к.т.н.

Олег БЕРЕСТ

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук, освітньо-наукової програми «Інформатика»
здобувача групи ІН.м-25 Чайки Богдана Вікторовича

1. Тема роботи: «Інформаційна система розпізнавання рухів у карате»
затверджую наказом по СумДУ від «06» грудня 2023 р. № 1412-VI _____
2. Термін здачі здобувачем кваліфікаційної роботи до 15 грудня 2023 року _____
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їй належить розробити)
1) Огляд технологій, що застосовуються для розпізнавання об'єктів, машинного навчання, комп'ютерного зору; 2) Постановка задачі та формування завдань дослідження; 3) Опис алгоритму розпізнавання рухів у карате; 4) Розробка додатку; 5) Аналіз результатів. _____
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх _____

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «06» листопада 2023 р.

Завдання прийняв до виконання

Керівник

(підпис)

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Перегляд технологій в області ідентифікації об'єктів, методів машинного навчання та комп'ютерного бачення.</i>	01.10-10.10.2023	
2	<i>Постановка задачі та завдання дослідження</i>	11.10-14.10.2023	
3	<i>Опис алгоритму розпізнавання рухів у карате</i>	15.10-31.10.2023	
4	<i>Розробка додатку для розпізнавання рухів у карате</i>	1.11-20.11.2023	
5	<i>Оформлення пояснювальної записки до дипломної роботи</i>	21.11-02.12.2023	

Здобувач вищої освіти

Керівник

(підпис)

(підпис)

АНОТАЦІЯ

Записка: 40 стор., 24 рис., 1 додаток, 20 літературних джерел.

Об'єкт дослідження — Інформаційна система розпізнавання рухів у карате

Мета роботи — створення програми на Python із застосуванням OpenCV для ідентифікації карате рухів.

Результати — здійснено детальний огляд літератури, методологій та інструментарію, що використовуються для ідентифікації та відслідковування певних об'єктів у фото- та відеоматеріалах з використанням технологій комп'ютерного бачення. Після вивчення існуючих методів було створено алгоритм для виявлення рухів у карате, який в подальшому був втілений у вигляді програмного додатку. Розроблений додаток дозволяє розпізнавати карате прийоми для різноманітних цілей, включаючи оцінку точності виконання технік та ідентифікацію конкретних технік. Програма була реалізована з використанням мови програмування Python.

ІНФОРМАЦІЙНА СИСТЕМА РОЗПІЗНАВАННЯ РУХІВ У КАРАТЕ, INFORMATION SYSTEM FOR KARATE MOVES RECOGNITION, MACHINE LEARNING, SPORT ACTIVITIES, ARTIFICIAL INTELLIGENCE

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Дослідження історії карате	8
1.2 Дослідження технічних аспектів карате	9
1.3 Дослідження актуальності проблеми	11
1.4 Дослідження моделей розпізнавання рухів у карате	15
1.5 Постановка задачі	20
2 ОГЛЯД АРХІТЕКТУРИ ТА ВИБІР МЕТОДІВ РЕАЛІЗАЦІЇ	21
2.1 Алгоритми комп'ютерного зору	21
2.2 Аналітичний огляд інструментів	27
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ	29
3.1 Проєктування застосунку	29
3.2 Тестування системи	34
ВИСНОВКИ	38
СПИСОК ЛІТЕРАТУРИ	39
ДОДАТОК	41
Додаток А. Вихідний код продукту	41

ВСТУП

В епоху цифрових інновацій, коли технології проникають у найрізноманітніші сфери нашого життя, з'являється безпрецедентна можливість для трансформації традиційних підходів у спорті. Особливо це стосується таких видів спорту, як карате, де точність, швидкість та техніка виконання рухів мають критичне значення. Важливість інтеграції технологічних новинок у спортивну практику відображається у зростаючому попиті на розробку сучасних тренувальних інструментів, які б підвищували якість та ефективність підготовки спортсменів. Саме тому, тема створення інформаційної системи для розпізнавання рухів у карате є надзвичайно актуальною та перспективною.

Ця дипломна робота має на меті розробку інноваційної системи, що здійснює автоматизоване розпізнавання та аналіз рухів каратистів, використовуючи алгоритми машинного навчання та комп'ютерного зору. Основна ідея полягає в тому, щоб створити інструмент, який би підвищував ефективність тренувань, дозволяючи спортсменам та тренерам отримувати детальний аналіз виконаних технік та вдосконалювати їх. Ця система має на увазі не лише виявлення та аналіз рухів, але й надання об'єктивних відомостей щодо покращення виконання технік, що є ключовим для досягнення високих результатів у спорті.

У першому розділі роботи ми зосередимося на основах карате як виду спорту, його історичному контексті та ключових аспектах техніки. Також буде розглянуто вплив технологій на спортивні дисципліни і зокрема на бойові мистецтва. Далі буде розглянуто сучасний стан розвитку технологій розпізнавання рухів та їх можливе застосування у спортивному контексті. Особлива увага буде приділена ролі комп'ютерного зору та машинного навчання у спортивних аналітичних системах.

Важливим етапом роботи стане практична частина, де ми розробимо та протестуємо алгоритми для ефективного виявлення та аналізу рухів у карате. Буде проведено детальний аналіз та експериментальні дослідження, спрямовані на

визначення найбільш ефективних підходів у розпізнаванні специфічних рухів у карате.

Завершальна частина роботи буде присвячена обговоренню отриманих результатів, їх практичному застосуванню та можливостям подальших досліджень у цій галузі. Цей проект має на меті не тільки створення функціональної та інноваційної системи, але й внесення вагомого вкладу у розвиток технологічних рішень для спорту та підвищення якості спортивної підготовки. Розробка такої системи має потенціал стати важливим інструментом не тільки для спортсменів та тренерів, але й для дослідників у галузі спортивних наук.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження історії карате

Карате, один з найбільш відомих та впливових видів бойових мистецтв у світі, має давню та захоплюючу історію. Цей розділ має на меті дослідити коріння карате, простежуючи його розвиток від ранніх форм до сучасного статусу.

Карате походить з Окінави, невеликого острова, що належить Японії. У 15-му столітті, під час династії Рюкю, Окінава була важливим торговим вузлом, де зливалися культури та бойові техніки з різних частин Азії, особливо з Китаю. Це суміш культур стала основою для формування унікального стилю бойових мистецтв, який згодом став відомий як Окінавське тей-ді (ручне бойове мистецтво).

Карате було значною мірою під впливом китайського кунг-фу, особливо після того, як китайські майстри почали переселятися на Окінаву. Техніки, стилі та філософські підходи кунг-фу зіграли ключову роль у формуванні ранніх стилів карате.

З часом, карате на Окінаві почало приймати більш структуровану форму, адаптуючи і вдосконалюючи техніки для місцевих умов. У 19-му столітті, після злиття Окінавського королівства з Японією, карате почало поширюватися по всій Японії. В цей період відбулися важливі зміни у педагогіці та філософії карате, включаючи акцент на дисципліні, формальних тренуваннях та розвитку духовності.

У 20-му столітті, зокрема після Другої світової війни, карате набуло міжнародного визнання. Різні стилі карате, такі як Шотокан, Вадо-рю, Годзю-рю, та Сіто-рю, з'явилися і почали поширюватися по всьому світу. Це призвело до створення численних федерацій та організацій, а також включення карате в багато міжнародних спортивних заходів.

Карате залишається важливою частиною багатьох культур і продовжує впливати на мільйони людей у всьому світі. Його історичні коріння та розвиток

відображають глибоку зв'язок між фізичним вихованням та духовним розвитком, який є центральним для багатьох східних бойових мистецтв.

1.2 Дослідження технічних аспектів карате

Карате є бойовим мистецтвом, що наголошує на різноманітності технік, які включають удари, блоки, стійки та кати. У цьому розділі ми детально розглянемо технічні аспекти карате, аналізуючи їх важливість для розвитку навичок каратиста[1].

1. Стійки (дачі)

Стійки у карате є основою для усіх рухів та технік. Вони не тільки забезпечують баланс та стабільність, але й визначають ефективність ударів та блоків. Різні стилі карате мають унікальні стійки, але деякі з них, такі як Зенкуцу-дачі (стійка напівлежачи) та Кокуцу-дачі (стійка задня), є загальноприйнятими.

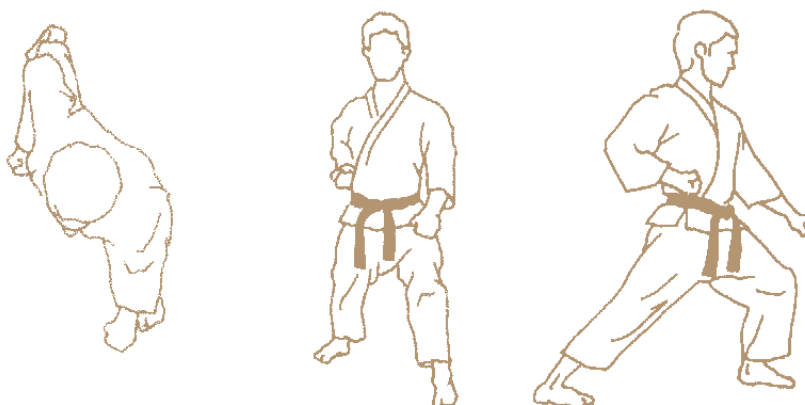


Рисунок 1.1 – Стійка «Зенкуцу-дачі»

2. Удари (цукі та гері)

Удари є ключовою складовою карате. Рукою (Цукі) та ногою (Гері) удари виконуються з використанням комбінації сили, швидкості та точності. Найвідомішими є прямий удар (Ой-цукі), обертовий удар (Маваші-гері) та бічний удар (Яко-цукі).

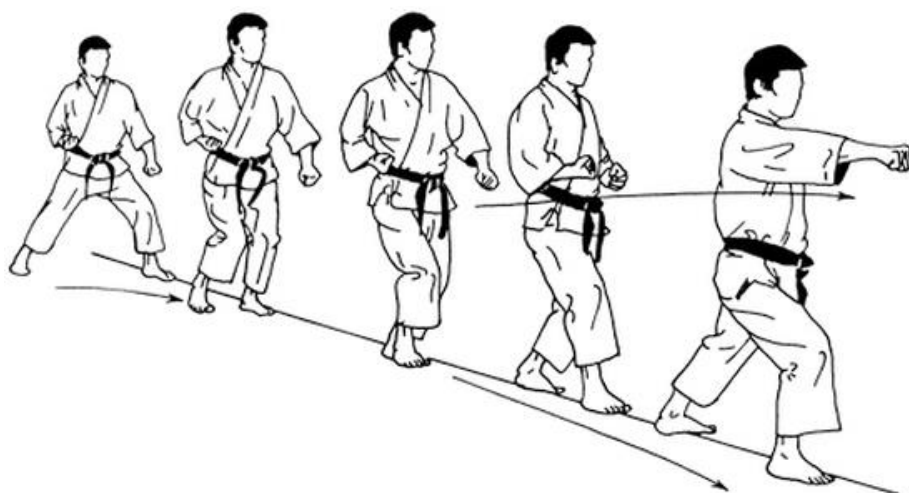


Рисунок 1.2 – Удар «Ой-цукі»

3. Блоки (укі)

Блоки в карате важливі для захисту від атак супротивника. Вони включають в себе не тільки фізичний захист, але й стратегічне позиціонування. Значні блоки включають високий блок (Джодан-укі), середній блок (Чудан-укі) та нижній блок (Гедан-укі)[2].



Рисунок 1.3 – Блок «Аге-уке»

4. Ката

Ката є формальними вправами, які включають у себе послідовності рухів, ударів, блоків та стійок. Кати не тільки демонструють технічну майстерність каратиста, але й розвивають його фізичні та ментальні навички. Вони відіграють ключову роль у передачі традицій та знань у карате.



Рисунок 1.4 – Елемент ката шотокан «Ганкаку»

5. Дихання та концентрація

У карате велика увага приділяється правильному диханню та концентрації. Ці елементи є фундаментальними для контролювання енергії та зосередженості під час виконання технік. Правильне дихання допомагає підтримувати ритм і забезпечує силу ударів.

1.3 Дослідження актуальності проблеми

В сучасному науковому контексті, переважна більшість фахівців підтверджують необхідність використання технологій розпізнавання у різних аспектах, особливо у системах відеоспостереження та моніторингу спортивних подій. Це стосується не тільки забезпечення безпеки, але й підвищення точності та об'єктивності при відстеженні змагань у різних спортивних дисциплінах[3].

Один із яскравих прикладів застосування таких технологій - це система відео асистента судді (VAR) у футболі. VAR дозволяє арбітрам переглядати відеозаписи спірних моментів під час матчу, щоб прийняти більш виважене і точне рішення. Це істотно впливає на результати гри та може визначати хід матчу.

Тим не менш, оскільки цей процес не є повністю автоматизованим та все ще залежить від людського судження, значна частина футбольних уболівальників - близько 90% - сприймають його як суб'єктивний. Це підкреслює потребу у подальшому розвитку та інтеграції авангардних технологій в спортивній індустрії, щоб забезпечити більшу об'єктивність та непристрасність у суддівстві[4].



Рисунок 1.5 – Футбольний рефері під час перегляду VAR

Технології розпізнавання мають потенціал кардинально змінити спосіб функціонування системи відео асистента судді (VAR) у спорті. Перш за все, застосування передових технологій може мінімізувати вплив людського фактору, який іноді може призводити до помилок.

У контексті VAR, суддям доводиться приймати складні рішення, на які часто чиниться тиск з боку обох команд у матчі. Кожна сторона змагання має свої інтереси, наприклад, одна команда може наполягати на тому, щоб гол або штрафний були не зараховані. Відповідно, об'єктивне та правильне рішення мінімізує потребу в подальших скаргах чи апеляціях[5].

По-друге, інтеграція технологій комп'ютерного зору та машинного навчання може значно прискорити процес прийняття рішень. Використання автоматизованих систем дозволяє аналізувати ситуації на полі швидше, ніж це робить людина, зменшуючи час розгляду спірних моментів з 5-10 хвилин до менше ніж однієї хвилини. Це не тільки підвищує ефективність ігрового процесу, але й забезпечує більшу точність у прийнятті рішень, оскільки машини здатні аналізувати відеодані без упереджень та емоційних впливів[6].

У карате також використовується VR (Video Review). Згідно останніх правил WKF (World Karate Federation), відбувається це наступним чином:

- 1) Спортсмен виконує певну техніку, яку потенційно повинні оцінити кутові судді;
- 2) Якщо дана техніка ними не оцінюється, представник спортсмена повинен підняти картку «VR», щоб запросити повторний перегляд рішення кутових;
- 3) Рефері зупиняє поєдинок, показує відповідний жест спортсмену і далі відбувається повторний перегляд певного фрагменту поєдинку кваліфікованими представниками суддівської бригади;
- 4) Даний процес може тривати близько 5 хвилин, судді переглядають покадрово один і той самий момент;
- 5) У випадку позитивного рішення, піднімається картка «YES», спортсмену зараховується відповідна техніка, його представнику повертається картка VR для можливих додаткових оспорювань;
- 6) У випадку негативного рішення, піднімається картка «NO», рахунок залишається незмінним, представник позбавляється картки VR, а відповідно і можливості в подальшому перебігу поєдинку оспорити рішення [7];



Рисунок 1.6 – Перегляд VR у карате

Підсумуємо: витрачається близько 5 хвилин, тренер має шанс бути позбавленим можливості оспорити більш важливі моменти у поєдинку.

За допомогою нашої майбутньої системи є можливість нівелювати подібні фактори, а саме таким чином:

1. Тренер може отримувати сповіщення від системи, що потенційно техніка його спортсмена має бути оцінена. У випадку, якщо від системи такого сповіщення не надходить, тренер задумується над потребою оспорювати рішення кутових суддів і таким чином збереже картку VR для більш важливих моментів у поєдинку[8].
2. Представники суддівської бригади витратять менше часу на перегляд повтору конкретного моменту поєдинку, адже система автоматично надаватиме підказки, чи є ця техніка оцінюваною, чи ні.

Тобто ми зараз розглядаємо нашу систему лише у якості помічника, адже спорт є динамічним та непередбачуваним, що на даний момент робить неможливим повну автоматизацію переглядів рішень.

1.4 Дослідження моделей розпізнавання рухів у карате

При проектуванні ключової частини нашої програми, яка відповідатиме за розпізнавання, необхідно здійснити аналіз доступних моделей згорткових нейронних мереж і вибрати ту, що найкраще відповідає цілям дослідження.

Серед моделей, що забезпечують ефективне виявлення об'єктів, можна виділити такі:

1. Faster R-CNN
2. YOLO (You Only Look Once)
3. SSD (Single-shot Detection)

Давайте детально розглянемо кожен з вищезгаданих моделей, починаючи з Faster R-CNN. Ця модель представляє собою розвинену версію R-CNN, створену Россом Гіршиком у 2014 році, і слугує для покращення детектування об'єктів:

- 1) Вхідне зображення проходить через згорткову нейронну мережу, що створює деталізовану карту ознак, яку далі обробляє шар RPN (Region Proposal Network);
- 2) Через карту ознак переміщується ковзаюче вікно, яке асоціюється з якорними областями, що мають різноманітні пропорції і розміри.
- 3) Шар RPN застосовує три різні пропорції і розміри, і на основі IoU (intersection-over-union) визначає, чи містить регіон об'єкт;
- 4) Після визначення регіонів інтересу (RoI), вони подаються до алгоритму Fast R-CNN, де відбувається додаткова обробка та класифікація з використанням повнозв'язних шарів, а також корегування координат пропозонованих регіонів для точного локалізування об'єктів.

Після розгляду Faster R-CNN, наступною моделлю, яка заслуговує на увагу, є YOLO.

Розроблена Джозефом Редмоном та Алі Фархаді у 2016 році, YOLO представляє собою проривний метод у сфері глибокого навчання. Основна перевага YOLO перед традиційними підходами полягає у швидкості обробки: модель

розпізнає об'єкти на зображенні за один прохід, що значно знижує час обробки і забезпечує можливість використання в системах, які потребують реального часу. Такий підхід різко контрастує з багатоетапними методами інших алгоритмів, які повторюють обробку зображень кілька разів, що веде до збільшення часу на розпізнавання.

На рисунку 1.7 представлено набір даних, на якому вже попередньо навчений YOLO.

person	bird	suitcase	fork	chair	toaster
bicycle	cat	frisbee	knife	sofa	sink
car	dog	skis	spoon	pottedplant	refrigerator
motorbike	horse	snowboard	bowl	bed	book
aeroplane	sheep	sports ball	banana	diningtable	clock
bus	cow	kite	apple	toilet	vase
train	elephant	baseball bat	sandwich	tvmonitor	scissors
truck	bear	baseball glove	orange	laptop	teddy bear
boat	zebra	skateboard	broccoli	mouse	hair drier
traffic light	giraffe	surfboard	carrot	remote	toothbrush
fire hydrant	backpack	tennis racket	hot dog	keyboard	
stop sign	umbrella	bottle	pizza	cell phone	
parking meter	handbag	wine glass	donut	microwave	
bench	tie	cup	cake	oven	

Рисунок 1.7 – Набір даних, на якому вже навчена модель YOLO

YOLO вирізняється своєю універсальністю у виявленні об'єктів, здатною розпізнавати до 80 різних типів об'єктів, які вже включені в модель. Однак, ключовою особливістю YOLO є можливість адаптації та навчання для ідентифікації нових видів об'єктів, що розширює сферу його застосування[10].

Функціонування YOLO базується на поділі вхідного зображення на декілька секцій, умовно створюючи сітку. Розглядаючи приклад сітки розміром 3x3 (рисунку 1.8), можна зрозуміти, як алгоритм аналізує кожен з цих секцій на предмет виявлення та класифікації об'єктів.



Рисунок 1.8 – Сітка 3x3 для моделі YOLO

Кожна секція на вхідному зображенні для YOLO містить специфічні параметри. Допустимо, ми хочемо виявити 3 класи об'єктів (наприклад, людей, автомобілів та літаків), тоді кожна секція буде описуватися 5 загальними параметрами та 3 параметрами специфічними для кожного класу[11].

Ці параметри оцінюються в контексті обмежуючих рамок. Під час підготовки навчального набору даних, об'єкти, які необхідно ідентифікувати, виділяються за допомогою цих рамок. Вони зазвичай являють собою прямокутні або квадратні зони, які відокремлюють об'єкти від решти зображення, як у випадку з автомобілями на рисунку 1.9.



Рисунок 1.9 – Обмеження на об'єктах (автомобілях)

Тепер нашим завданням є визначення п'яти ключових параметрів для кожної з дев'яти секцій сітки розміром 3x3. Для ілюстрації, розгляньмо фотографію однієї секції, де знаходиться автомобіль.



Рисунок 1.10 – Одна з секцій з автомобілем

У центрі кожної секції YOLO-сітки знаходиться червона точка, яка відображає середину обмежувальної рамки. Горизонтальна синя стрілка визначає параметр tx , що є відстанню від центральної точки до лівої сторони секції, а вертикальна синя стрілка представляє ty - відстань від центру до верхньої межі секції. Білі стрілки вказують на ширину та висоту обмежувальної рамки відносно розмірів секції[12].

Додатково в моделі YOLO використовуються параметри, які відображають вірогідність наявності об'єкта в рамці (індекс об'єктності) та ймовірності того, що цей об'єкт є людиною, автомобілем або літаком (p_1 , p_2 , p_3 відповідно). Усі секції сітки 3×3 мають ці параметри, що дозволяє YOLO ефективно ідентифікувати об'єкти.

Алгоритм Single Shot Detector (SSD) - це метод обробки зображень для виявлення об'єктів, який використовує єдиний прохід (single shot) для визначення місцезнаходження та класифікації об'єктів на зображенні. Ось його ключові характеристики:

- 1) **Швидкість:** SSD забезпечує швидку обробку, оскільки він визначає об'єкти на зображенні в один прохід. Це робить його значно швидшим в порівнянні з іншими алгоритмами, такими як Faster R-CNN;
- 2) **Одноетапний процес:** На відміну від алгоритмів, які використовують окремі етапи для виявлення областей інтересу та класифікації, SSD робить це одночасно, що знижує час обробки[13];
- 3) **Багатомасштабність:** SSD аналізує зображення на різних масштабах, що дозволяє точно виявляти об'єкти різного розміру;
- 4) **Використання згорткових нейронних мереж:** SSD використовує базову згорткову мережу (зазвичай VGG-16), після чого додаються додаткові згорткові шари для виявлення об'єктів на різних масштабах;
- 5) **Області виявлення:** SSD визначає фіксовану кількість областей на кожному рівні масштабу, які використовуються для виявлення об'єктів;

Завдяки цим характеристикам, SSD є ефективним рішенням для задач реального часу, де важливі швидкість та точність.

1.5 Постановка задачі

Інформаційна система для розпізнавання рухів у карате може стати ключовим інструментом у забезпеченні справедливості та об'єктивності під час карате змагань. Її розробка вимагає врахування ряду специфічних принципів, щоб забезпечити точність та ефективність системи.

- 1) **Врахування стильових особливостей технік у карате.** Оскільки різні стилі карате, такі як вадо-рю чи сіто-рю, мають унікальні способи виконання однакових технік, система має бути здатна розпізнавати ці відмінності. Наприклад, блок у стилі вадо-рю виконується відрізняється від виконання цього ж блоку в стилі сіто-рю;
- 2) **Точність розпізнавання техніки та її відмежування від інших рухів.** Система повинна вміти точно ідентифікувати техніки карате, незалежно від зовнішніх факторів. До таких факторів можна віднести ситуації, коли спортсмени перебувають у клінчі або коли рухи третіх осіб (наприклад, рефері або глядачів) потрапляють у кадр і можуть ускладнити розпізнавання техніки;
- 3) **Швидкість реагування системи.** Це важливо як для тренерів, яким потрібно швидко реагувати на ситуацію та вирішувати, чи варто оскаржувати рішення суддів, так і для суддівської команди, якій потрібно оперативно оцінювати виконані техніки[14];

Під час розробки такої системи важливо також врахувати можливість існування виключних ситуацій, наприклад, коли виконання техніки відбувається під час незвичайних обставин, які можуть бути нестандартними для програмного розпізнавання. Це дозволить забезпечити більш глибоке та точне розуміння виконаних технік, що в свою чергу сприятиме справедливому оцінюванню виступів спортсменів.

2 ОГЛЯД АРХІТЕКТУРИ ТА ВИБІР МЕТОДІВ РЕАЛІЗАЦІЇ

1. Алгоритми комп'ютерного зору

Перед тим як перейти до вивчення специфічних алгоритмів, які використовуються в комп'ютерному зорі, важливо зрозуміти саму суть та основні концепції цієї області. Комп'ютерний зір (англ. Computer Vision, CV) – це галузь, що лежить на перетині машинного навчання та комп'ютерних наук, і займається наданням машинам здатності "бачити" та інтерпретувати візуальну інформацію, подібно до людського сприйняття. Вона є ключовою складовою штучного інтелекту, спрямованою на створення систем, здатних виконувати завдання, які для людини вимагають візуального розуміння.

Мета комп'ютерного зору полягає у вченні комп'ютерів аналізувати та інтерпретувати оточуючий світ через обробку цифрових зображень та відео. Це досягається шляхом впровадження трьох основних компонентів:

- 1) **Отримання зображень.** Цей етап включає перетворення фізичного світу в цифрові дані за допомогою візуальних пристроїв, таких як веб-камери, цифрові камери, зеркальні фотоапарати та професійне 3D-обладнання. Це дозволяє створити віртуальне відображення реальних об'єктів та сцен, яке може бути подальше аналізоване системою;
- 2) **Обробка інформації.** На цьому етапі відбувається аналіз зібраних зображень для ідентифікації важливих елементів, таких як точки, лінії, контури та інші геометричні форми. Використання складних математичних алгоритмів та технік, таких як виділення границь (edge detection), сегментація, класифікація, та виявлення об'єктів, є критично важливим для точної обробки та розуміння візуальної інформації[15];
- 3) **Аналіз даних.** Цей крок полягає у використанні даних, отриманих після обробки зображень, для прийняття рішень або здійснення конкретних дій. Наприклад, це може включати в себе розпізнавання об'єктів, відстеження їх руху, ідентифікацію осіб чи інтерпретацію поведінкових патернів.

Такий високорівневий аналіз дозволяє машинам не просто сприймати візуальну інформацію, але й робити висновки та приймати самостійні рішення на основі цієї інформації.

Варто поглибити розгляд компонента "Обробка інформації" в контексті комп'ютерного зору, акцентуючи увагу на його ключових та найпопулярніших методах:

1) **Визначення границь (Edge Detection).** Цей метод передбачає аналітичний розбір зображення з метою трансформування його в сукупність кривих ліній та відрізків. Його застосування спрямоване на акцентування ключових елементів зображення, що істотно спрощує подальшу обробку, оскільки відкидаються менш значущі дані. Такий підхід дозволяє виокремити важливі контури та форми, спрощуючи ідентифікацію об'єктів.

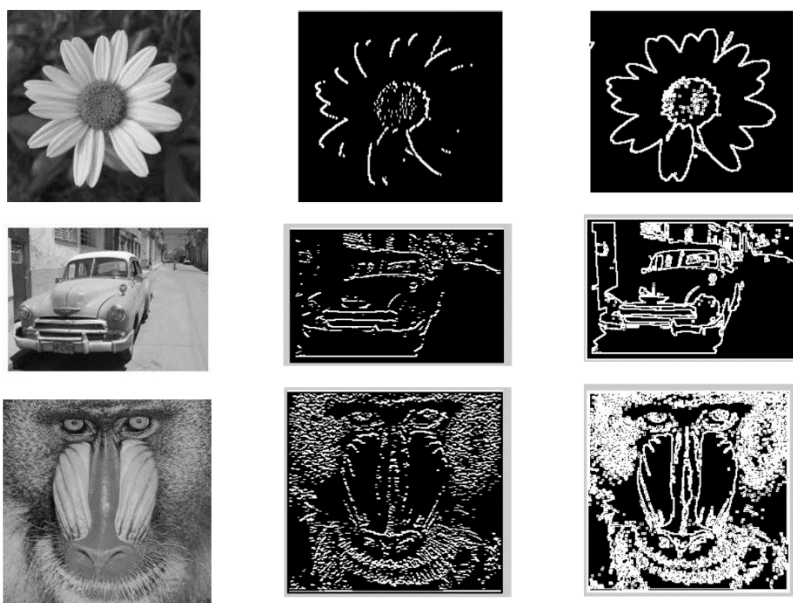


Рисунок 2.1 – Edge detection

2) **Сегментація.** Суть цього методу полягає у розподілі зображення на ділянки або сегменти, кожен з яких має певну мітку. Процес сегментації включає присвоєння унікальної мітки кожному пікселю зображення, що дозволяє системі відрізнити різні об'єкти та їх частини. Це може включати

в себе як виділення цілого об'єкта, так і його окремих компонентів, що допомагає в подальшому аналізі та класифікації.

- 3) **Класифікація.** Цей метод є фундаментальним для розуміння та ідентифікації об'єктів на зображенні. Він базується на віднесенні об'єктів до певних категорій чи класів, наприклад, відрізняючи kota від собаки на одному зображенні. Класифікація в комп'ютерному зорі важлива для розпізнавання та ідентифікації об'єктів, що можуть включати в себе не тільки тварин, але й різноманітні предмети, людей, сцени та інші візуальні елементи[16]. Цей метод тісно пов'язаний з виявленням об'єктів, оскільки дозволяє не тільки знаходити об'єкти на зображенні, але й правильно їх ідентифікувати.



Рисунок 2.2 – Класифікація, локалізація та визначення об'єктів (Object Detection)

У сфері комп'ютерного зору існує цілий арсенал алгоритмів, призначених для специфічних цілей і задач. До них відносяться, зокрема, каскади Хаара, методи блендінгу та лінійна фільтрація зображень.

Каскади Хаара, зокрема, є високоефективними для розпізнавання обличч в реальному часі. Вони використовують набір "слабких" класифікаторів - простих

функцій, які обчислюються на основі властивостей прямокутників, що відображають зміну інтенсивності пікселів. Ці класифікатори організовані в каскадну структуру, де кожен наступний класифікатор уточнює результати попереднього, що дозволяє швидко відкидати неподходящі області зображення і зосереджуватися на найбільш ймовірних кандидатах для детектування облич.

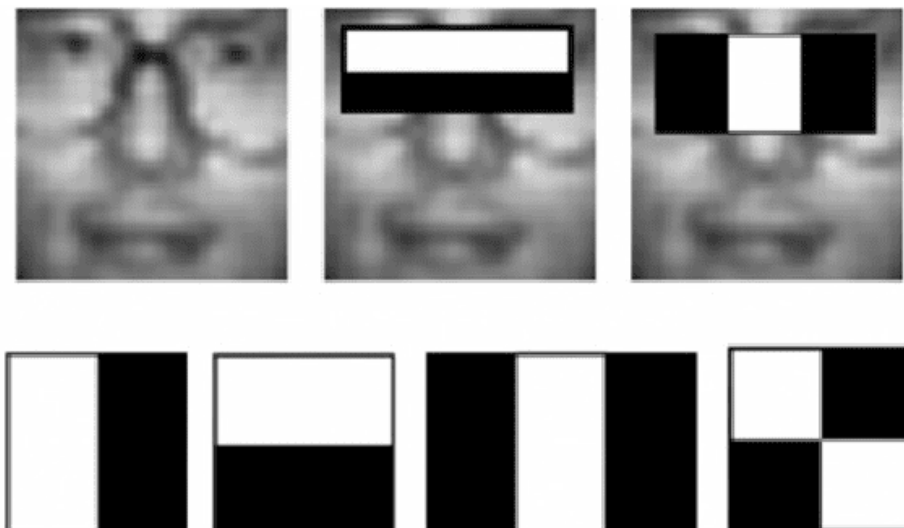


Рисунок 2.3 – Графічне представлення каскадів Хаара

Блендінг у галузі комп'ютерного зору – це процес, який включає зливання двох або більше зображень за допомогою простих арифметичних операцій. Уявімо ситуацію, де потрібно інтегрувати один об'єкт із першого зображення в друге, де вже існує власний фон. В цьому процесі вирізання об'єкта відбувається з урахуванням його точного розташування, замінюючи фонову частину другого зображення. Якщо ж об'єкт із першого зображення накладається на об'єкт другого зображення, то використовується образ другого [17].

У цьому контексті якість обрізання об'єкта має вирішальне значення. Недбале обрізання, що залишає видимі сліди фону навколо об'єкта, може привести до появи очевидних білих країв на фінальному зображенні. Якщо фон нерівномірний і містить, наприклад, білі ділянки, просте видалення білих пікселів не буде ефективним. Тут використовується більш витончена техніка – створення маски, яка регулює прозорість залежно від відстані до білих областей. У місцях

з білими плямами на першому зображенні, пікселі з другого зображення змішуються, зменшуючи видимість недоліків вирізання і створюючи більш плавний перехід між об'єктами.

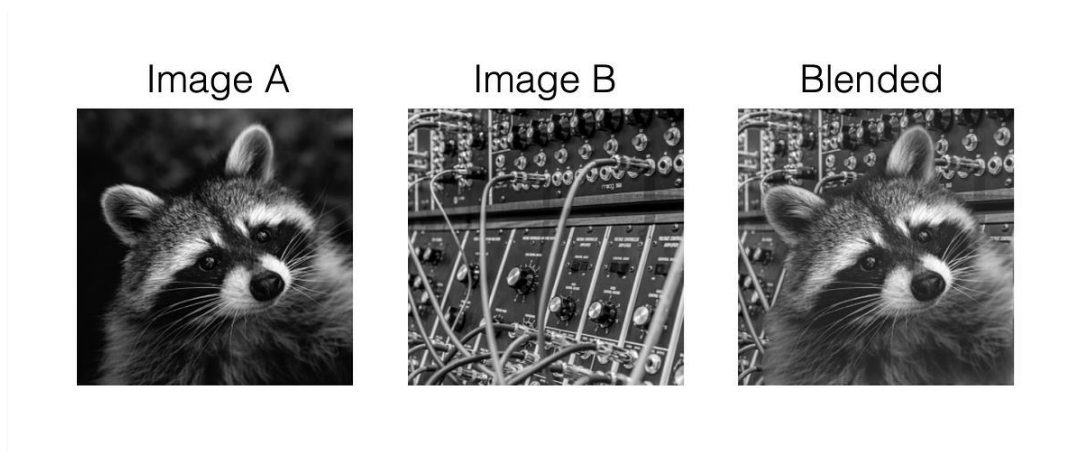


Рисунок 2.4 – Графічне представлення блендінгу

Останньою буде розглянуто лінійну фільтрацію зображень. Лінійна фільтрація зображень - це процес, що використовується для виконання широкого спектру завдань, таких як підсилення або розмиття границь, виявлення кутів та усунення шуму з цифрових зображень. Цей метод можна ілюструвати через операцію обчислення середнього значення пікселів у маленьких областях, званих ядрами або фільтрами[18].

Простий приклад лінійної фільтрації – обчислення ковзного середнього для згладжування зображення. Потрібно розглянути ядро згортання (фільтр) розміром 3×3 , де кожен елемент ядра має однакову вагу $1/9$. Формула згортання в цьому випадку агрегує значення пікселів у вікні 3×3 навколо кожної точки зображення, щоб обчислити середнє значення, яке призначається центральному пікселю вікна.

Цей процес можна описати наступним чином:

$$g[n, m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k, l] = \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 f[n-k, m-l]$$

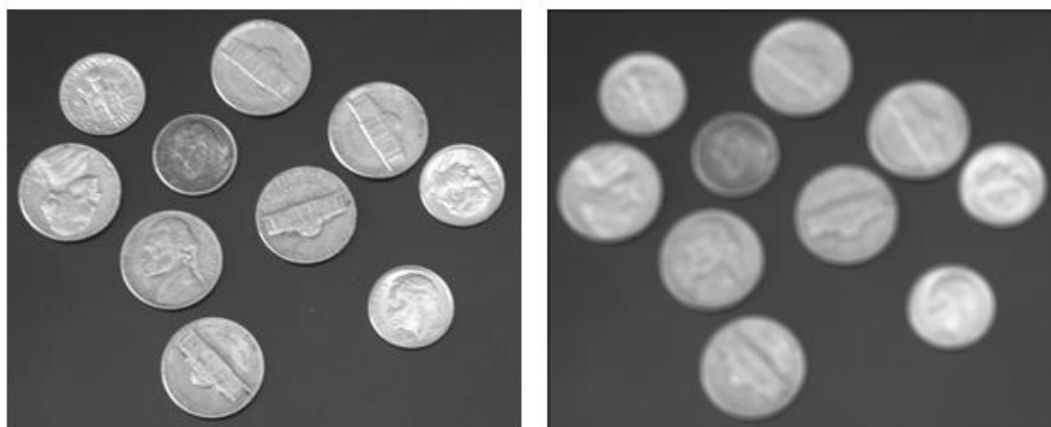
Або ж, можна переписати формулу трохи інакше і отримати у підсумку таке:

$$(f * h)[m, n] = \frac{1}{9} \sum_{k,l} h[m-k, n-l]$$

де f – зображення, g – результат фільтрації, h – ядро згортання, а k та l – індекси, що проходять через всі сусідні пікселі.

Використовуючи бібліотеку OpenCV, можна застосувати таке згортання з допомогою попередньо визначеного ядра:

```
kernel = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 1]], np.float32) / 9
dst = cv2.filter2D(img, -1, kernel)
```



Original Image

Filtered Image

Рисунок 2.5 – Зображення до та після лінійної фільтрації

Таким чином, було проведено ознайомлення з базовими алгоритмами, що використовуються в галузі комп'ютерного зору. Тепер можна приступити до детального розгляду спеціалізованих інструментів, які допоможуть вирішити поставлені задачі.

2.2 Аналітичний огляд інструментів

Для адекватного вирішення завдань, пов'язаних із аналізом та розпізнаванням рухів у карате, необхідно залучити спеціалізовані інструменти комп'ютерного зору. У цій дипломній роботі не варто звертатися до комерційних програмних продуктів, адже це не позначиться на рівні дослідження.

Навпаки, існуючі вільно-розповсюджені рішення, такі як бібліотека OpenCV, володіють потужними алгоритмами для комп'ютерного зору, обробки зображень, що дозволяють з успіхом вирішувати складні задачі, включаючи ідентифікацію, слідування, трансформацію зображень та реалізацію методів машинного навчання. OpenCV надає набір інструментів, який може бути ефективно застосований для розробки систем, здатних аналізувати специфічні та складні рухи у карате[19].

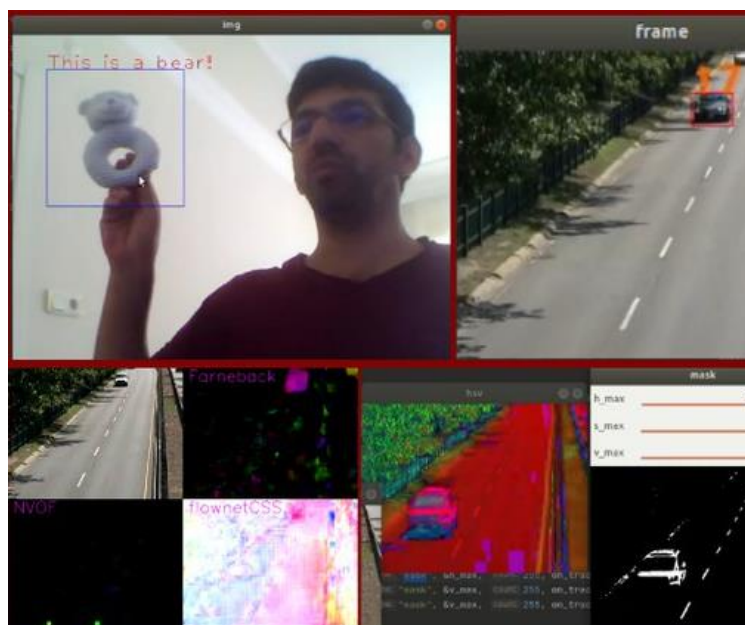


Рисунок 2.6 – Приклад використання OpenCV

OpenCV, ініційована корпорацією Intel та на даний момент підтримувана компаніями Willow Garage та Itseez, є відкритою бібліотекою з багатьма можливостями у сфері комп'ютерного зору. Хоча первинно вона була створена з

використанням C++, її потенціал значно розширено завдяки інтерфейсам для мов програмування, як-от Java, Ruby, Matlab, Lua, Python. Це робить OpenCV доступною для широкого спектру розробників.

При виборі інструментів для розробки системи розпізнавання рухів у карате, критично важливим є вибір мови програмування. Вибір мови визначатиме не тільки інструменти, які ми зможемо використовувати, але й можливості інтеграції з іншими системами та платформами.

У цій роботі планується розробити початкову версію системи, яка надасть можливість ознайомлення з функціоналом розпізнавання рухів у карате. Для цього було обрано Python – мову програмування, яка ідеально відповідає потребам проекту завдяки своїй простоті та гнучкості.

Python, створений Гвідо ван Россумом у 1990 році, є мовою високого рівня з динамічною семантикою, що сприяє чистому та зрозумілому коду. Він включає в себе велику кількість бібліотек та інструментів для розробки різноманітних програм – від веб-додатків до складних алгоритмів машинного навчання, які є основою сучасних систем комп'ютерного зору.

Підтримуючи цю мету, також буде використано фреймворк PyTorch[20], який надає гнучкі можливості для розробки та тренування нейронних мереж, та утиліти imutils, які спрощують обробку зображень. Вони, у поєднанні з бібліотекою OpenCV, дозволяють створювати потужні системи комп'ютерного зору, ефективно розпізнавати рухи у карате.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Проєктування застосунку

Плануючи архітектуру додатку для ідентифікації карате технік, варто врахувати такі ключові аспекти:

- 1) Здатність системи адаптуватися та розширюватися у майбутньому.
- 2) Інтуїтивна зрозумілість інформаційної системи для кінцевого користувача.
- 3) Сумісність із вже існуючими інфраструктурними рішеннями.

Конструкція додатку повинна включати такі важливі елементи:

- 1) Інтерфейс користувача додатку.
- 2) Модуль розпізнавання, який є осердям програми і відповідає за ідентифікацію та слідкування за рухами.
- 3) Навчальний датасет, на базі якого функціонує модуль розпізнавання.
- 4) Налаштування та конфігурації, які керують роботою модулю розпізнавання.

Схематичне представлення застосунку зображено на рисунку 3.1.



Рисунок 3.1 – Представлення застосунку у вигляді схеми

Потрібно зосередитися на понятті датасету. Перед тим, як поглиблюватись у цю тему, важливо зрозуміти, що таке датасет. Це набір даних, який використовується в аналітиці та машинному навчанні.

Щоб краще зрозуміти важливість датасетів, розглядається простий приклад. Перед походом в ліс за грибами, потрібно знати, які з них їстівні, а які небезпечні. Найлегший спосіб відрізнити їх - це за зовнішнім виглядом, але для цього потрібно вміти ідентифікувати різні види грибів. Ми можемо знайти зображення в інтернеті або в книзі, щоб визначити, які гриби безпечні для збору, а які слід уникати.

У сфері машинного навчання датасети використовуються за аналогічним принципом. Ми збираємо обширний набір медіаданих з різними умовами освітлення, ракурсами та зайвими об'єктами, щоб навчити систему розпізнавати на фотографіях чи відео певні рухи у карате, які є припустимими для використання.

Деякі з фрагментів датасету представлено на рисунку 3.2.



Рисунок 3.2 – Фрагменти датасету з рухами у спарингах карате

Потрібно детально розглянути кожен фрагмент датасету, який є набором даних для аналізу в рамках технологій машинного навчання. Кожне ім'я або назва,

представлена на зображеннях у датасеті, містить короткі підказки про те, що відбувається на кожному з фрагментів.

На першому фрагменті спостерігається виконання техніки "ой-цукі" в чіткому та ясному представленні. На другому фрагменті відображено техніку "ой-цукі" знову, але з точки зору спини спортсмена. На третьому фрагменті також демонструється "ой-цукі", але тут зображення ускладнюється через те, що спортсмен частково закритий рефері, що ускладнює визначення техніки.

Четвертий фрагмент представляє собою ідеальне виконання техніки "ура-маваші". На п'ятому фрагменті також можна побачити "ура-маваші", але з виконавцем, частково закритим іншим спортсменом, що робить розпізнавання техніки складнішим. Останній фрагмент показує один з ідеальних прикладів виконання маваші-гері в корпус, де чітко видно ногу спортсмена і момент удару по корпусу опонента.

Ці шість фрагментів, хоч і містять певні спільні елементи, відрізняються своєю унікальністю та ситуаціями, які вони демонструють. Важливо розуміти, що в датасеті може бути значно більше варіацій, ракурсів і ситуацій, ніж просто ці шість фрагментів. У даному випадку датасет включає 54 фотографії та 12 відеозаписів з карате, зроблені з різних ракурсів, у різних світлових умовах, з різними варіаціями рухів спортсменів на татамі, а також включає випадки, де спостерігаються порушення правил спарингу. Щоб розширити та покращити датасет, потрібно більше матеріалу, але це вимагає більш потужного та витривалішого обладнання, що може бути непрактично в рамках цього дослідження. Основною задачею дослідження є створення MVP (мінімально життєздатного продукту), тому важливо враховувати обмеження ресурсів.

Тепер варто звернути увагу на аспекти проектування, особливо на взаємодію між "Користувачем та застосунком".

При розробці інтерфейсу взаємодії між користувачем і застосунком, необхідно враховувати наступні ключові моменти:

- 1) Користувач має можливість передати в систему фотографії або відеоматеріали для подальшого аналізу за допомогою вбудованої системи розпізнавання в застосунку. Це важливо для забезпечення відповідної обробки та аналізу даних.
- 2) Користувачеві надається можливість налаштування конфігурацій застосунку, що дозволяє кастомізувати роботу системи відповідно до його потреб та вимог. Це забезпечує більшу гнучкість та адаптивність застосунку до індивідуальних переваг користувача.
- 3) Важливим є те, що роль користувача може виконувати не тільки людина, але й інший програмний застосунок. Це відкриває можливості для інтеграції нашого продукту в більш широкую інфраструктуру, де він може взаємодіяти з іншими системами та застосунками, розширюючи таким чином його функціональність та застосування.

Таким чином, при проектуванні взаємодії користувача з застосунком, необхідно ретельно продумати ергономіку та інтуїтивність інтерфейсу, а також забезпечити гнучкість налаштувань та інтеграційний потенціал для різних типів користувачів.



Рисунок 3.3 – Взаємодія між користувачем та застосунком

Тепер потрібно розробити базовий функціонал нашого майбутнього застосунку, враховуючи наступні особливості та сценарії:

- 1) Важливо переконатися, що версія OpenCV, яка використовується, є не старішою за 3.2. Якщо ця умова не дотримана, застосунок має ініціювати створення власного трекера для обробки даних. В іншому випадку, коли версія OpenCV відповідає вимогам, активуються всі доступні трекери для підвищення ефективності роботи застосунку.
- 2) Користувач повинен надати медіаконтент для аналізу. У випадку, якщо користувач не надає фото- або відеоматеріал, застосунок автоматично розпочинає пошук доступної камери для трансляції зображень або відео. Після отримання медіаданих від користувача, розпочинається їхній аналіз.
- 3) У випадку з відеоматеріалами, необхідно слідкувати за завершенням їхнього відтворення. Це дозволяє запобігти безцільній роботі застосунку. По завершенню відтворення відео, застосунок має завершити аналіз та підготувати результати для подальшої обробки або відображення користувачу.

Таким чином, ці ключові сценарії допомагають забезпечити гнучку та ефективну роботу застосунку, враховуючи різноманітні умови використання та потреби користувачів.

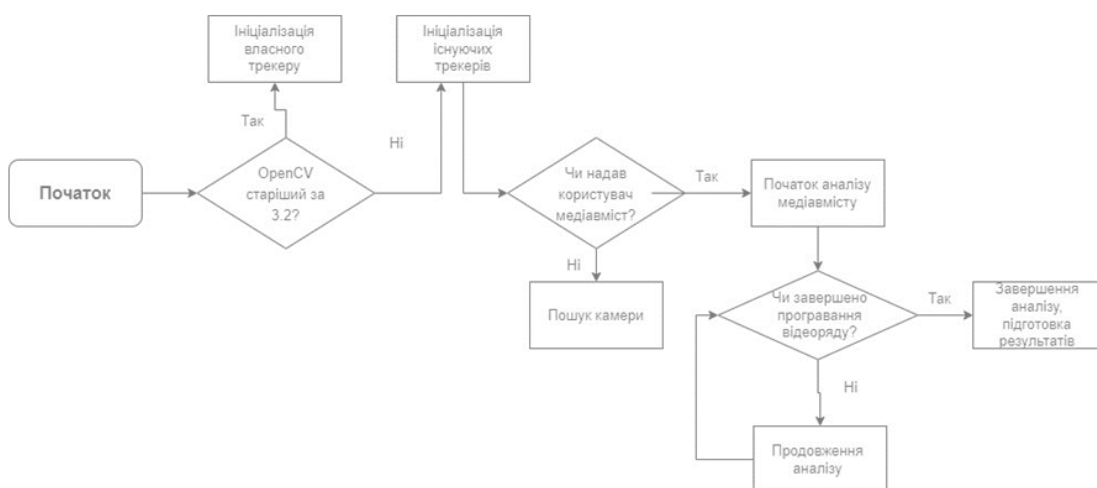


Рисунок 3.4 – Попередній алгоритм роботи застосунку

По завершенню розробки структури та алгоритму додатку, відбувається перехід до етапу програмування та проведення випробувань. Вихідний код можна знайти у Додатку А.

3.2 Тестування системи

Для тестування застосунку було взято відеоряд, який демонструє найкращі моменти зі змагань з карате. На рисунку 3.5 представлено оригінальний кадр з певних змагань.



Рисунок 3.5 – Вхідний кадр зі змагань з карате

На цьому кадрі було перевірено, що саме бачить система, хто є виконавцем техніки, який бал потрібно віддати за виконання даної техніки та відсоток впевненості (точність) системи у тому, що вона бачить. На рисунку 3.6 представлено результат роботи системи.

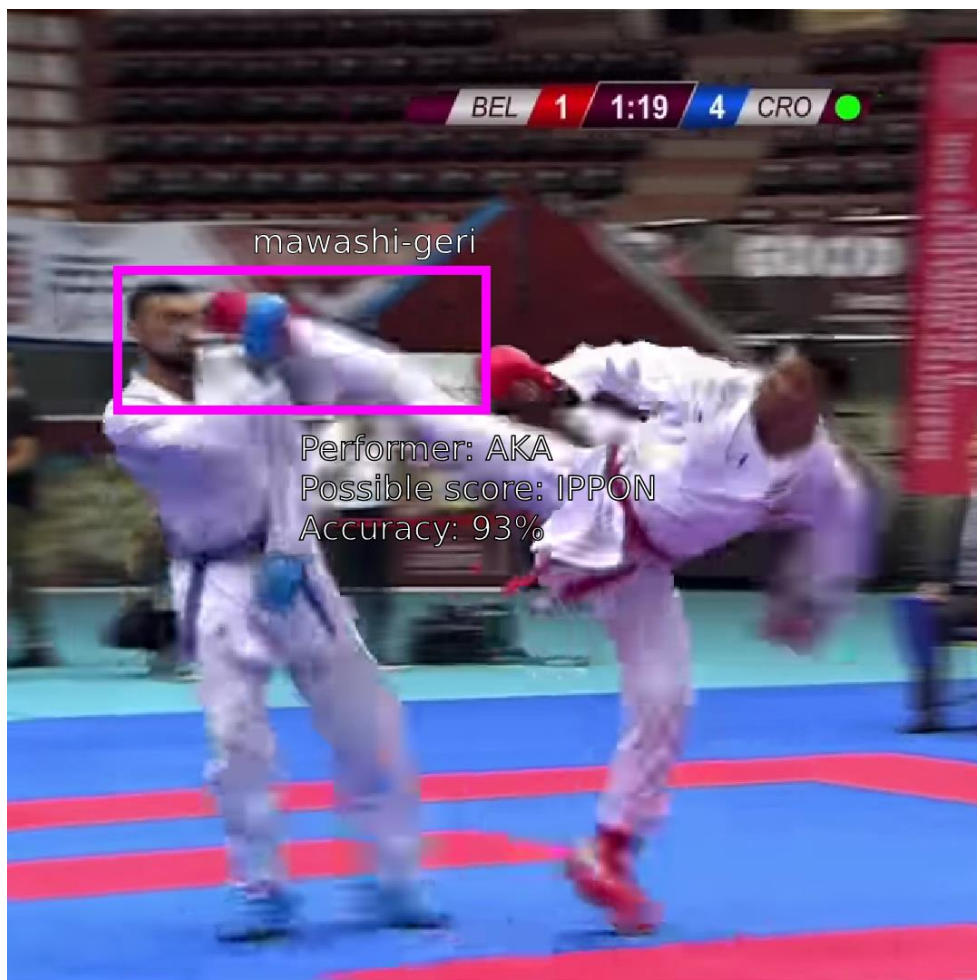


Рисунок 3.6 – Кадр з технікою після роботи системи

На цьому кадрі можна спостерігати, що система коректно розпізнала техніку «маваші-гері», що її виконавцем є спортсмен з червоним екіпіруванням (АКА), оцінка за неї повинна бути 3 бали (ІППОН) та впевненість (точність) системи у цьому – 93%.

Треба провести тестування також на іншому випадку. Спортсмен виконує техніку «ой-цукі» (удар рукою) в голову, проте вона повернута спиною до камери. На рисунку 3.7 представлено оригінальний кадр зі змагань.



Рисунок 3.7 – Інший кадр до обробки системою

На рисунку 3.8 представлено результат роботи системи.



Рисунок 3.8 – Результат після обробки системою

Як можна помітити, система також розпізнає коректно техніку, хоч вона й абсолютно інша. Виконавцем цієї системи також є спортсмен у червоному екіпіруванні, потенційним балом є YUKO, проте в цьому випадку «впевненість» системи у факті залікової техніки – 49.8%.

Це відбувається не просто так і в цьому є один з недоліків системи розпізнавання. Спортсменка, яка виконувала цю техніку, була повернута спиною до камери. Окрім цього, навіть на оригінальному кадрі не до кінця зрозуміло, чи є ця техніка заліковою, адже вона пробивається повз голови. Тобто, ракурс зйомки техніки також багато чого вирішує і тут є два варіанти: або потрібно надати системі більший датасет для навчання, або ж подбати про це питання з матеріально-технічної точки зору, розставивши більше камер з більшої кількості ракурсів.

Проте, результат тестування задовільняє базові потреби: система здатна розпізнавати коректно техніку, показники точності не є низькими, а також вона це робить оперативно.

ВИСНОВКИ

У ході виконання даної дипломної роботи було розроблено інформаційну систему, що дозволяє ефективно розпізнавати рухи у карате. Система базується на сучасних алгоритмах комп'ютерного зору та машинного навчання і демонструє високий рівень точності в ідентифікації технік карате.

Проект показав, що використання інформаційних технологій у спорті може значно підвищити якість тренувань та суддівства. Розроблена система не тільки забезпечує об'єктивне оцінювання виконання технік, але й відкриває нові можливості для аналізу та покращення навичок спортсменів.

Масштабування системи та її інтеграція в спортивну інфраструктуру може розширити її застосування не тільки для тренувань, але й для проведення змагань на вищому рівні. Завдяки гнучкості використання мови програмування Python і відкритих бібліотек, як-от OpenCV, проект має великий потенціал для подальшого розвитку і адаптації під різноманітні потреби в області спортивної аналітики.

СПИСОК ЛІТЕРАТУРИ

1. Overview of cascade classifiers [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/topics/computer-science/classifier-cascade>
2. Image Blending in OpenCV [Електронний ресурс] – Режим доступу: <https://www.educative.io/answers/what-is-image-blending-opencv>
3. OpenCV Documentation [Електронний ресурс] – Режим доступу: <https://docs.opencv.org/4.x/d1/dfb/intro.html>
4. World Karate Federation Kumite Rules [Електронний ресурс] – Режим доступу: <https://karate.ua/media/5834e542-8069-43ee-8467-a36214157802.pdf>
5. World Karate Federation Kata Rules [Електронний ресурс] – Режим доступу: <https://karate.ua/media/ad1eb1ee-344e-444b-bd4c-6340788f18ca.pdf>
6. PyTorch Fundamentals [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/training/paths/pytorch-fundamentals/>
7. Linear Filter [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/topics/engineering/linear-filter>
8. YOLO — You only look once, real time object detection explained [Електронний ресурс] – Режим доступу: <https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006>
9. Mask R-CNN [Електронний ресурс] – Режим доступу: <https://paperswithcode.com/paper/mask-r-cnn>
10. Deep Residual Learning for Image Recognition [Електронний ресурс] – Режим доступу: <https://paperswithcode.com/paper/deep-residual-learning-for-image-recognition>
11. What is Edge Detection – An Introduction [Електронний ресурс] – Режим доступу: <https://www.mygreatlearning.com/blog/introduction-to-edge-detection/>
12. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way [Електронний ресурс] – Режим доступу: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
13. Introduction to CNN [Електронний ресурс] – Режим доступу: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>

14. What is a use case? [Электронный ресурс] – Режим доступа: <https://www.wrike.com/blog/what-is-a-use-case/>
15. Convert OpenCV Image to PyTorch Tensor using Python [Электронный ресурс] – Режим доступа: <https://lindevs.com/convert-opencv-image-to-pytorch-tensor-using-python>
16. What is an ER diagram? [Электронный ресурс] – Режим доступа: <https://www.lucidchart.com/pages/er-diagrams>
17. Object detection using Faster R-CNN [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/en-us/cognitive-toolkit/object-detection-using-faster-r-cnn>
18. Faster R-CNN Explained for Object Detection Tasks [Электронный ресурс] – Режим доступа: <https://blog.paperspace.com/faster-r-cnn-explained-object-detection/>
19. Karate: Origin, Description and History [Электронный ресурс] – Режим доступа: <https://www.britannica.com/sports/karate>
20. How Single-Shot Detector Works? [Электронный ресурс] – Режим доступа: <https://developers.arcgis.com/python/guide/how-ssd-works/>

ДОДАТОК

Додаток А. Вихідний код продукту

```
import cv2
import numpy as np
import imutils
from model import KarateTechniqueClassifier

classifier = KarateTechniqueClassifier.load_model("karate_model")

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()

    if not ret:
        break

    preprocessed_frame = preprocess_frame(frame)

    technique = classifier.predict(preprocessed_frame)

    display_result(frame, technique)

    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()

def preprocess_frame(frame):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    return blurred

def display_result(frame, technique):
    cv2.putText(
        frame,
        f'Recognized Technique: {technique}',
        (10, 30),
```



```
    cv2.FONT_HERSHEY_SIMPLEX,  
    1,  
    (0, 255, 0),  
    2,  
    )  
cv2.imshow("Karate Technique Recognition", frame)
```

```
frame = cv2.imread("technique_1.jpg")  
preprocessed_frame = preprocess_frame(frame)
```

```
recognized_technique = model.predict(preprocessed_frame)
```

```
display_result(frame, recognized_technique)
```

```
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

```
from sklearn.svm import SVC  
import joblib
```

```
class KarateTechniqueClassifier:  
    def __init__(self):  
        self.classifier = SVC(kernel='linear')  
  
    def train(self, X_train, y_train):  
        self.classifier.fit(X_train, y_train)  
  
    def predict(self, features):  
        return self.classifier.predict([features])[0]  
  
    def save_model(self, file_path):  
        joblib.dump(self.classifier, file_path)  
  
    @staticmethod  
    def load_model(file_path):  
        classifier = joblib.load(file_path)  
        return classifier
```