

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

«На правах рукопису»
УДК 004.043

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ

«___» _____ 2024 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інженерія програмного
забезпечення інформаційних систем»
зі спеціальності 121 «Інженерія програмного забезпечення»**

**на тему: «Алгоритмічне та програмне забезпечення для автоматизованого
аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя»**

Виконав (-ла):
студент (-ка) II курсу, групи ІІ-22мп
Онофрійчук Анна Вікторівна

Керівник:
доцент кафедри ІІІ, к.т.н., доц.,
Ліщук Катерина Ігорівна

Рецензент:
доцент кафедри ІСТ, к.т.н., доц.,
Писаренко Андрій Володимирович

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії
Рівень вищої освіти – другий (магістерський)
Спеціальність – 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Інженерія програмного забезпечення інформаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Едуард ЖАРІКОВ

« ____ » _____ 2024р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Онофрійчук Анні Вікторівні

1. Тема дисертації «Алгоритмічне та програмне забезпечення для автоматизованого аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя», науковий керівник дисертації Ліщук Катерина Ігорівна, к.т.н., доцент, затверджені наказом по університету від «11» листопада 2023 р. № 5168-с
2. Термін подання студентом дисертації «16» січня 2024 р.
3. Об'єкт дослідження – алгоритмічні та програмні рішення для аналізу частоти серцевих скорочень на основі відеозапису обличчя.
4. Предмет дослідження – методи, моделі, інструменти для представлення і перетворення знань щодо програмного забезпечення для аналізу частоти серцевих скорочень на основі відеозапису обличчя
5. Перелік завдань, які потрібно розробити – аналіз існуючих рішень у сфері аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя, оцінка їх ефективності, дослідження можливостей вдосконалення існуючих методів та розробка нових підходів для підвищення ефективності аналізу частоти серцевих скорочень.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу – 2 плакати
7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

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

9. Дата видачі завдання «1» листопада 2022 р.

Календарний план

з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Видача завдання	1.11.2022	
2	Аналіз існуючих рішень	02.11.22-16.11.22	
3	Формування функціональних нефункціональних вимог	17.11.22-01.12.22	
4	Розробка методів, алгоритмів	02.12.22-16.01.23	
5	Проектування системи	17.01.23-17.06.23	
6	Реалізація програмного забезпечення	01.09.23-03.11.23	
7	Виконання експериментальних досліджень	04.11.23-19.11.23	
8	Оформлення пояснювальної записки	20.11.23-18.12.23	
9	Подання дисертації на попередній захист	19.12.2023	
10	Подання дисертації на захист	19.01.2024	

Студент

Анна ОНОФРІЙЧУК

Науковий керівник

Катерина ЛІЩУК

РЕФЕРАТ

Розмір пояснювальної записки – 142 аркушів, містить 23 ілюстрації, 18 таблиць, 3 додатки, 20 посилань на джерела.

Актуальність теми. У роботі розглянуто проблему використання обробки відеозапису обличчя для автоматизованого аналізу частоти серцевих скорочень у медичній діагностиці та дослідженнях, показано основні особливості існуючих розв'язань проблеми, їх переваги та недоліки. Виявлено потребу в розробці та вдосконаленні алгоритмів та програмного забезпечення для ефективного вимірювання частоти серцевих скорочень на основі обробки відеозапису обличчя.

Мета дослідження. Основною метою є розробка алгоритмічного та програмного забезпечення для аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя з метою підвищення точності та швидкості аналізу.

Об'єкт дослідження: алгоритмічні та програмні рішення для аналізу частоти серцевих скорочень на основі відеозапису обличчя.

Предмет дослідження: методи, моделі, інструменти для представлення і перетворення знань щодо програмного забезпечення для аналізу частоти серцевих скорочень на основі відеозапису обличчя.

Для реалізації поставленої мети **сформульовані наступні завдання:**

- провести огляд існуючих методів аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя та визначити їхні переваги та недоліки;
- визначити оптимальні методи обробки відео, такі як відслідковування обличчя, витягування ключових точок та інші, з урахуванням їхньої ефективності та швидкодії;

- розробити комбінаційний алгоритм, який інтегрує вибрані методи обробки відео, алгоритми розпізнавання обличчя та точний розрахунок частоти пульсу;
- реалізувати програмне забезпечення на основі розробленого алгоритму;
- провести валідацію та тестування розробленого програмного забезпечення з використанням відеозаписів різних умов.

Наукова новизна результатів магістерської дисертації полягає в тому, що *набуло подальшого розвитку* використання існуючих методів та алгоритмів обробки відеозапису обличчя, точного обчислення частоти пульсу та ефективного пристосування до різних умов для аналізу частоти серцевих скорочень.

Практичне значення отриманих результатів полягає в тому, що розроблене програмного забезпечення для аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя може бути використане в медичній діагностиці, фітнесі, психофізіологічних дослідженнях та технологіях носимих пристроїв, надаючи ефективний та неінвазивний метод вимірювання пульсу для моніторингу здоров'я та активності.

Зв'язок з науковими програмами, планами, темами. Робота виконувалась на кафедрі інформатики та програмної інженерії Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".

Апробація. Наукові положення дисертації пройшли апробацію на V міжнародній науково-практичній конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023)» присвяченої 125-й річниці КПІ ім. Ігоря Сікорського.

Публікації. Наукові положення дисертації опубліковані в:

- 1) Онофрійчук А.В., Алгоритмічне та програмне забезпечення для автоматизованого аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя/ А.В, Онофрійчук, К.І.Ліщук // Матеріали V Міжнародної науково-практичної конференції молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023)» – м. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 19-21 грудня 2023 р.

Ключові слова: ПУЛЬС, СЕРЦЕВІ СКОРОЧЕННЯ, СПЕКТРАЛЬНИЙ АНАЛІЗ, АГП, ОБРОБКА, ВІДЕОЗАПИС.

ABSTRACT

Explanatory note size – 142 pages, contains 23 illustrations, 18 tables, 3 applications, 20 references.

Topicality. The paper examines the problem of using face video processing for automated heart rate analysis in medical diagnostics and research, shows the main features of existing solutions to the problem, their advantages and disadvantages. The need for development and improvement of algorithms and software for effective measurement of heart rate based on facial video processing has been identified.

The aim of the study. The main target is to develop algorithmic and software for heart rate analysis based on facial video processing in order to improve the accuracy and speed of the analysis.

The object of research: algorithmic and software solutions for heart rate analysis based on facial video recording.

The subject of research: methods, models, tools for representing and transforming knowledge about face video heart rate analysis software.

To achieve this goal, the **following tasks** were formulated:

- review existing heart rate analysis methods based on face video processing and identify their advantages and disadvantages;
- determine optimal methods of video processing, such as face tracking, extraction of key points, and others, taking into account their efficiency and speed;
- develop a combinational algorithm that integrates selected video processing methods, face recognition algorithms and accurate heart rate calculation;
- implement software based on the developed algorithm;

- conduct validation and testing of the developed software using video recordings of various conditions.

The scientific novelty of the results of the master's dissertation is an innovative approach to heart rate analysis based on face video processing is proposed, which is determined by the use of a combinational algorithm, which includes the optimal synergy of the best existing video processing methods and algorithms, accurate heart rate calculation, and effective adaptation to various conditions. The result was achieved by developing the appropriate software.

The practical value of the obtained results is developed heart rate analysis software based on face video processing can be used in medical diagnosis, fitness, psychophysiological research and wearable technology, providing an effective and non-invasive heart rate measurement method for health and activity monitoring.

Relationship with working with scientific programs, plans, topics. Work was performed at the Department of Informatics and Software Engineering of the National Technical University of Ukraine «Kyiv Polytechnic Institute. Igor Sikorsky».

Approbation. The scientific provisions of the dissertation were tested at the Fifth International Scientific and Practical Conference of Young Scientists and Students "Software Engineering and Advanced Information Technologies (SoftTech-2023)" dedicated to the 125th anniversary of Igor Sikorsky Kyiv Polytechnic Institute.

Publications. The scientific provisions of the dissertation were published in:

Onofriichuk A.V., Algorithmic and software for automated heart rate analysis based on face video processing / A.V. Onofriichuk, K.I. Lishchuk // Proceedings of the Fifth International Scientific and Practical Conference of Young Scientists and Students "Software Engineering and Advanced Information Technologies (SoftTech-2023)". - Kyiv: NTUU "KPI them. Igor Sikorsky", December 19-21, 2023.

Keywords: PULSE, HEART RATE, SPECTRAL ANALYSIS, AHE, PROCESSING, VIDEO RECORDING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	12
ВСТУП	13
1 ОГЛЯД НАЯВНИХ РІШЕНЬ АВТОМАТИЗОВАНОГО ПЗ ДЛЯ АНАЛІЗУ ЧАСТОТИ СЕРЦЕВИХ СКОРОЧЕНЬ НА ОСНОВІ ВІДЕОЗАПИСУ	14
1.1 Огляд предметної області	14
1.2 Постановка задачі	16
1.3 Аналіз наявних рішень	18
1.3.1 FaceHeart	19
1.3.2 PulsePass	22
1.3.3 Vital Signs Camera	23
Висновки до розділу	25
2 МОДЕЛІ ТА МЕТОДИ	27
2.1 Загальний огляд методів аналізу частоти серцевих скорочень	27
2.2 Методи для обробки відеозапису обличчя для виявлення частоти серцевих скорочень	28
2.2.1 Видобуток обличчя з відеозапису	29
2.2.1.1 MTCNN	31
2.2.1.2 DeepSORT	34
2.2.2 Відстеження руху та пульсації обличчя	37
2.2.2.1 Спектральний аналіз	38
2.2.2.2 Адаптивне гістограмне підвищення (АНЕ)	40
Висновки до розділу	42
3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	43
3.1 Засоби розробки	44
3.2 Обґрунтування вибраних технологій	47
3.3 Архітектура програмного забезпечення	49
3.3.1 USE-CASE діаграма	49
3.3.2 Діаграма послідовності	51
3.3.3 Діаграма класів	53
3.3.4 Інтерфейс користувача	54
Висновки до розділу	58
4 ОЦІНКА ЕФЕКТИВНОСТІ ТА ТЕСТУВАННЯ ЗАПРОПОНОВАНОГО РІШЕННЯ	60

4.1 Тестування алгоритмів	60
4.1.1 Метрики для оцінки результатів аналізу серцевої активності	60
4.1.2 Урахування факторів, що впливають на точність аналізу, таких як освітлення та обличчя пацієнт	61
4.1.3 Результати	62
4.2 Оцінка ефективності запропонованого рішення	65
Висновки до розділу	68
5 МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ	70
5.1 Опис ідеї проекту	70
5.2 Технологічний аудит ідеї проекту	71
5.3 Аналіз ринкових можливостей запуску стартап-проекту	72
5.4 Розроблення ринкової стратегії проекту	80
5.5 Розроблення маркетингової програми стартап-проекту	84
Висновки до розділу	87
ЗАГАЛЬНІ ВИСНОВКИ	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	91
ДОДАТОК А	
ЛІСТИНГ ПРОГРАМНОГО КОДУ	95
ДОДАТОК Б	
ГРАФІЧНИЙ МАТЕРІАЛ	138
ДОДАТОК Г	
РЕЗУЛЬТАТИ ПЕРЕВІРКИ НА ПЛАГІАТ	141

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

UML - Unified Modeling Language (Уніфікована мова моделювання):
Мова, що використовується для візуалізації, специфікації, конструювання та документування об'єктно-орієнтованих систем.

HTML - HyperText Markup Language (Мова гіпертекстової розмітки):
Стандартна мова розмітки, яка використовується для створення веб-сторінок.

АНЕ - Adaptive Histogram Equalization (Адаптивне вирівнювання гістограми):

Техніка цифрової обробки зображень, яка використовується для підвищення контрастності зображень.

MTCNN - Multi-task Cascaded Convolutional Networks (Багатозадачні каскадні згорткові мережі):

Структура, розроблена як рішення для виявлення та вирівнювання обличчя.

ВСТУП

Сучасний світ неспокійно рухається вперед завдяки стрімкому розвитку технологій, які перепрограмовують наше повсякденне існування та відкривають нові можливості в галузі медицини та здоров'я. Однією з ключових складових здоров'я є стан нашої серцево-судинної системи, контроль над якою визначає наше благополуччя. Аналіз частоти серцевих скорочень та їх динаміки є критично важливим для виявлення медичних аномалій та здорового фізіологічного стану.

Мета цієї магістерської роботи полягає в розробці та дослідженні алгоритмів та програмного забезпечення для автоматизованого аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя. Використання комп'ютерного зору та обробки відеоданих для неінвазивного вимірювання пульсу відкриває широкі перспективи в медичній практиці, наукових дослідженнях та моніторингу пацієнтів.

Робота спрямована на розробку програмного забезпечення, яке здатне автоматично аналізувати відеозаписи обличчя для визначення частоти серцевих скорочень та виявлення будь-яких аномалій у їх динаміці. Вона також розглядає різноманітні методи обробки зображень, аналіз кольору та яскравості, а також інші техніки для отримання корисної інформації з відеоданих.

Автоматизований аналіз серцевої активності на основі відеозапису обличчя має потенціал стати основою для подальшого розвитку неінвазивних методів діагностики та моніторингу стану пацієнтів, що є надзвичайно важливим для сучасної медицини та здоров'я людини. Дослідження, які представлені в цій роботі, можуть відкрити шлях до розширення можливостей у сфері аналізу серцевої активності та впровадження автоматизованих систем для контролю та діагностики.

1 ОГЛЯД НАЯВНИХ РІШЕНЬ АВТОМАТИЗОВАНОГО ПЗ ДЛЯ АНАЛІЗУ ЧАСТОТИ СЕРЦЕВИХ СКОРОЧЕНЬ НА ОСНОВІ ВІДЕОЗАПИСУ

1.1 Огляд предметної області

Аналіз частоти серцевих скорочень є важливою складовою медичних та наукових досліджень, спрямованих на виявлення аномалій серцево-судинної системи та оцінку стану пацієнта. Підходи до вимірювання частоти серцевих скорочень можуть бути найрізноманітнішими, включаючи використання спеціальних медичних приладів, електрокардіограм (ЕКГ), а також аналіз фізіологічних сигналів через обробку відеозапису обличчя[18].

У наш час, завдяки постійному розвитку технологій і збільшенню доступності відеозапису, обробка зображень обличчя може стати потужним інструментом для недорогого та неконтактного вимірювання частоти серцевих скорочень. Цей підхід може бути особливо корисним у ситуаціях, коли інші методи вимірювання неможливі або непрактичні, наприклад, при віддаленому спостереженні або в обмежених умовах.

Оглядаючи літературу та дослідження в даній області, можна виділити кілька ключових напрямків досліджень та наявні підходи до аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя.

Зважаючи на різноманітність підходів до аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя, варто вказати на кілька основних категорій методів та рішень, що існують в цій області:

- методи обробки зображень обличчя. Один з основних підходів полягає в застосуванні алгоритмів комп'ютерного зору для виявлення ключових анатомічних точок на обличчі, таких як очі, ніс, та рот. Після цього можуть використовуватися різні методи вимірювання пульсу, зокрема вимірювання змін яскравості або кольору на обличчі в часі[19];

- моделі глибокого навчання. Нейронні мережі глибокого навчання стали популярним інструментом для вирішення завдань аналізу відеоданих. Моделі, які використовують зображення обличчя для вимірювання пульсу, можуть бути навчені на великих наборах даних та вміти виявляти субтельні зміни на зображеннях, пов'язані із серцевою діяльністю[20];

- використання спеціальних камер та сенсорів. Деякі системи використовують спеціальні камери або сенсори, розроблені для точного вимірювання пульсу на основі фізіологічних змін, які відбуваються під час серцевої активності. Ці підходи можуть бути точними, але можуть вимагати спеціалізованого обладнання[20];

- додатки для моніторингу здоров'я. В останні роки на ринку з'явилися мобільні додатки та пристрої, які дозволяють користувачам моніторити свою серцеву активність вдома або в реальному часі. Ці рішення дозволяють пацієнтам та медичному персоналу вести контроль над станом здоров'я;

- використання відкритих джерел даних та розробка алгоритмів з відкритим вихідним кодом. Багато дослідників і розробників активно співпрацюють і обмінюються знаннями та розробленими рішеннями в цій області через відкритий вихідний код та відкриті дані.

Завершуючи огляд предметної області, важливо відзначити кілька основних викликів і перспектив для подальших досліджень:

- точність та надійність. Одним з ключових завдань є покращення точності та надійності методів аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя. В умовах реальних дійсностей можуть виникати різноманітні завади, такі як зміни освітлення, рухи, шум на зображенні тощо;

- доступність і практичність. Важливим аспектом є зробити ці технології доступними та практичними для широкого кола користувачів, включаючи лікарів, дослідників, і звичайних людей. Практичність у використанні відіграє велику роль у розвитку цієї області;

- приватність і безпека. Збір та обробка даних, пов'язаних із фізіологічними параметрами, ставить питання про приватність та безпеку даних. Важливо розробляти та використовувати методи, що гарантують адекватний захист і конфіденційність особистої інформації;

- сфери застосування. Область аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя має широкий потенціал застосування. Вона може бути корисною в медицині для недорогого та зручного моніторингу пацієнтів, у спортивних дослідженнях, а також у сферах, пов'язаних зі стресом та емоційним станом.

1.2 Постановка задачі

У даному підрозділі магістерської роботи ми розглянемо основні завдання та виклики, які стоять перед дослідженням аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя.

1) Розробка алгоритмів визначення пульсу. Однією з основних задач є розробка алгоритмів, які дозволяють визначати частоту серцевих скорочень на основі аналізу відеозапису обличчя. Ці алгоритми повинні бути точними та надійними, а також враховувати різні умови, такі як рухи, зміни освітлення та інші завади:

- виявлення обличчя. Першим кроком є виявлення обличчя на відеозаписі. Це може вимагати використання алгоритмів розпізнавання обличчя, таких як алгоритми на основі каскадних класифікаторів або нейронних мереж для локалізації обличчя на зображенні[20];

- відстежування ключових точок. Після виявлення обличчя, необхідно відстежувати ключові анатомічні точки, такі як очі, ніс та рот, на кожному кадрі відеозапису. Це може виконуватися за допомогою методів розпізнавання обличчя або методів відстежування об'єктів[20];

- аналіз змін на зображенні. Для визначення частоти серцевих скорочень обробка змін на зображенні є ключовою. Це може містити аналіз змін яскравості, кольору, або інших параметрів на обличчі в залежності від серцевої активності;

- визначення пульсу. Останній крок — визначення пульсу на основі аналізу змін на зображенні. Це може виконуватися різними методами, включаючи аналіз гармонічних складових, фільтрацію сигналу, та інші підходи.

2) Збір та обробка відеоданих. Для аналізу частоти серцевих скорочень необхідно зібрати велику кількість відеоданих. Завданням є розробка методів для збору та обробки відеозаписів, які містять обличчя пацієнта. Це може містити вибір оптимальних камер, обрання кутів зйомки та видалення шумів:

- вибір оптимального обладнання. Визначення оптимального обладнання, такого як камери, освітлення та сенсори, є важливим завданням. Вибір відповідного обладнання впливає на якість та точність отриманих даних;

- розробка методів синхронізації даних. Синхронізація відеозапису з даними про серцеву активність або іншими параметрами є важливою, оскільки дозволяє встановити часовий зв'язок між обличчям і серцевою активністю;

- обробка та фільтрація даних. Для отримання точних результатів необхідно провести обробку і фільтрацію відеоданих, враховуючи потенційні завади та артефакти.

3) Валідація та порівняння з наявними методами. Однією з важливих завдань є валідація розроблених алгоритмів та порівняння їх з наявними методами аналізу частоти серцевих скорочень. Це допоможе визначити ефективність та точність розроблених рішень:

- валідація на реальних даних. Проведення валідації на реальних відеоданих з пацієнтами або учасниками досліджень для оцінки точності та надійності розроблених рішень;

- порівняння з наявними методами. Порівняння результатів з наявними методами аналізу пульсу для оцінки переваг та обмежень нових розробок.

4) Розробка програмного забезпечення. Для практичного використання результатів дослідження необхідно розробити програмне забезпечення, яке може виконувати аналіз відеозапису обличчя та визначення частоти серцевих скорочень. Це може містити розробку інтерфейсу користувача та інтеграцію зі сучасними технологіями:

- інтерфейс користувача. Розробка зручного та інтуїтивно зрозумілого інтерфейсу для користувачів, що дозволить легко завантажувати відеодані, виконувати аналіз та переглядати результати;

- автоматизація процесу аналізу. Створення автоматизованого процесу аналізу відеоданих та визначення пульсу, який мінімізує втручання оператора та забезпечує надійність результатів;

- інтеграція з іншими системами. Можливість інтегрувати розроблене програмне забезпечення з іншими медичними системами чи системами моніторингу здоров'я.

Ці задачі стануть основою для подальших досліджень та розробки програмного забезпечення для аналізу частоти серцевих скорочень на основі відеозапису обличчя, і вони визначають напрямок і важливість дослідження.

1.3 Аналіз наявних рішень

Існують деякі рішення та дослідження в області аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя, які можна розглянути як приклади:

- FaceHeart. Дослідники з університету MIT розробили систему під назвою "FaceHeart," яка використовує вбудовані камери на сучасних смартфонах для аналізу серцевого ритму. Ця система вимірює зміни в кольорі обличчя, які виникають від кровообігу, та використовує їх для визначення пульсу;
- PulsePass. PulsePass - це програмний продукт, який використовує веб-камеру комп'ютера для вимірювання пульсу шляхом аналізу змін кольору обличчя. Він був розроблений для використання в додатках для моніторингу здоров'я та фізичної активності;
- Vital Signs Camera. Forsa, компанія-розробник медичної техніки, розробила "Vital Signs Camera," яка вимірює пульс та інші фізіологічні параметри шляхом аналізу змін відтінків кольору обличчя. Ця система була спроектована для використання в медичних закладах та лікарнях;
- Open Source Solutions. Існують також відкриті рішення, такі як OpenFace та OpenCV, які надають інструменти для аналізу відеозаписів обличчя та можуть бути використані для визначення пульсу, хоча це вимагає додаткової розробки та налаштування;
- дослідження університетів. Багато університетів та дослідницьких груп також займаються дослідженнями в цій області. Вони розробляють нові алгоритми та методи для аналізу частоти серцевих скорочень на основі відеозапису обличчя та публікують свої результати у наукових журналах та конференціях.

Ці приклади демонструють різноманітність рішень, що існують в галузі аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя.

1.3.1 FaceHeart

FaceHeart - це інноваційне дослідницьке рішення, розроблене університетом Массачусетського технологічного інституту (MIT), призначене

для аналізу серцевого ритму на основі обробки відеозапису обличчя. Це одне з прикладів використання інформації, отриманої з відеозапису, для неперервного вимірювання пульсу і моніторингу серцевої активності без використання традиційних медичних приладів. На рисунку (1.1) зображена головна сторінка продукту.



Рисунок 1.1 – Головна сторінка FaceHeart

Основні характеристики та принципи роботи FaceHeart включають наступне:

- колірне відеооброблення. FaceHeart використовує відеозаписи обличчя, в яких аналізуються зміни в кольорі шкіри обличчя. Зміни в кольорі обумовлені кровообігом і змінами в кількості крові, що протікає під шкірою під впливом серцевих скорочень;
- вимірювання кольорових пікселів. Алгоритм FaceHeart вимірює кольорні пікселі на вибраній ділянці обличчя та відстежує їхні зміни з часом. Зокрема, він фокусується на зеленому та червоному кольорах, оскільки вони найкраще відображають кровообіг;
- перетворення в сигнал пульсу. За допомогою алгоритмів обробки образів і математичних перетворень зібрані дані про зміни кольору обличчя

перетворюються в сигнал пульсу. Цей сигнал пульсу може бути використаний для визначення частоти серцевих скорочень;

- визначення частоти пульсу. Якщо розрахований сигнал пульсу знайдено, FaceHeart може точно визначити частоту серцевих скорочень. Це дозволяє в реальному часі вимірювати пульс та відстежувати його зміни;

- можливості застосування. FaceHeart має потенціал застосування у медицині, фізіологічних дослідженнях, моніторингу фізичної активності та інших сферах, де необхідно вимірювати пульс. Він може бути використаний в мобільних додатках, сучасних смартфонах та інших пристроях для моніторингу здоров'я.

На рисунку (1.2) зображено функціонал, який компанія представляє своїм користувачам у цьому додатку.



Рисунок 1.2 – Функціонал додатка FaceHeart

FaceHeart є одним із прикладів інноваційного використання технології в аналізі фізіологічних параметрів на основі відеозапису обличчя. Його розробка відкриває нові можливості для здоров'я та медичного моніторингу, забезпечуючи неінвазивний та зручний спосіб вимірювання пульсу та вивчення серцевої активності.

1.3.2 PulsePass

PulsePass – це остання на ринку безпечна, точна та візуальна програма для перевірки та моніторингу пульсу, багатфункціональна та проста у використанні, своєчасний доступ до стану здоров'я вашого тіла з приємним інтерфейсом та сучасним дизайном(рис. 1.3).

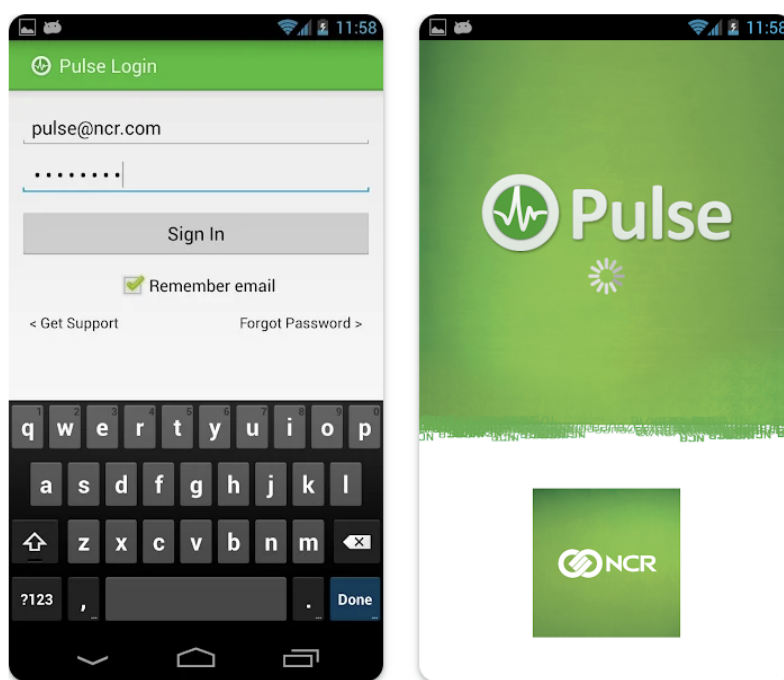


Рисунок 1.3 – Вигляд додатка PulsePass

Просто відкривши програму, ви можете виміряти свій пульс у будь-який час (під час відпочинку, тренувань або сну), у будь-якому місці, показуючи найкращий стан свого тіла.

Особливості:

- можливість використання будь-де. Ви можете вимірювати частоту серцевих скорочень (пульс або серцебиття) у будь-який час і будь-де, турбуючись про здоров'я серця;
- проста операція. Обережно натиснувши пальцями на задній панелі камери та ліхтарика, ви можете прочитати пульс за кілька секунд. Ви можете

зберегти частоту пульсу в іншому стані (наприклад, тренування, відпочинок тощо) у центрі здоров'я, щоб легко спостерігати за змінами частоти пульсу;

- записуйте та відстежуйте свій артеріальний тиск, рівень насичення крові киснем і температуру тіла в будь-який час і в будь-якому місці;
- постійне зберігання. Результати вимірювань для постійного збереження та забезпечують візуальне відображення вишуканих заяв;
- підтримка обміну результатами пульсу з Apple Health;
- часове нагадування. Ви можете встановити регулярне нагадування, яке завжди нагадуватиме про регулярний тестовий пульс (пульс або серцебиття);
- соціальні мережі. Підтримується спільний доступ до WeChat, QQ та інших соціальних мереж і друзів сім'ї, щоб показати стан вашого здоров'я (для частоти серцевих скорочень).

1.3.3 Vital Signs Camera

Vital Signs Camera - це інноваційна система, розроблена компанією Forsa, призначена для вимірювання різних фізіологічних параметрів, включаючи пульс та частоту дихання, на основі аналізу відеозапису обличчя. Ця система робить можливим неперервний та неінвазивний моніторинг здоров'я пацієнтів та користувачів у медичних закладах та за їхніми межами (рис. 1.4).

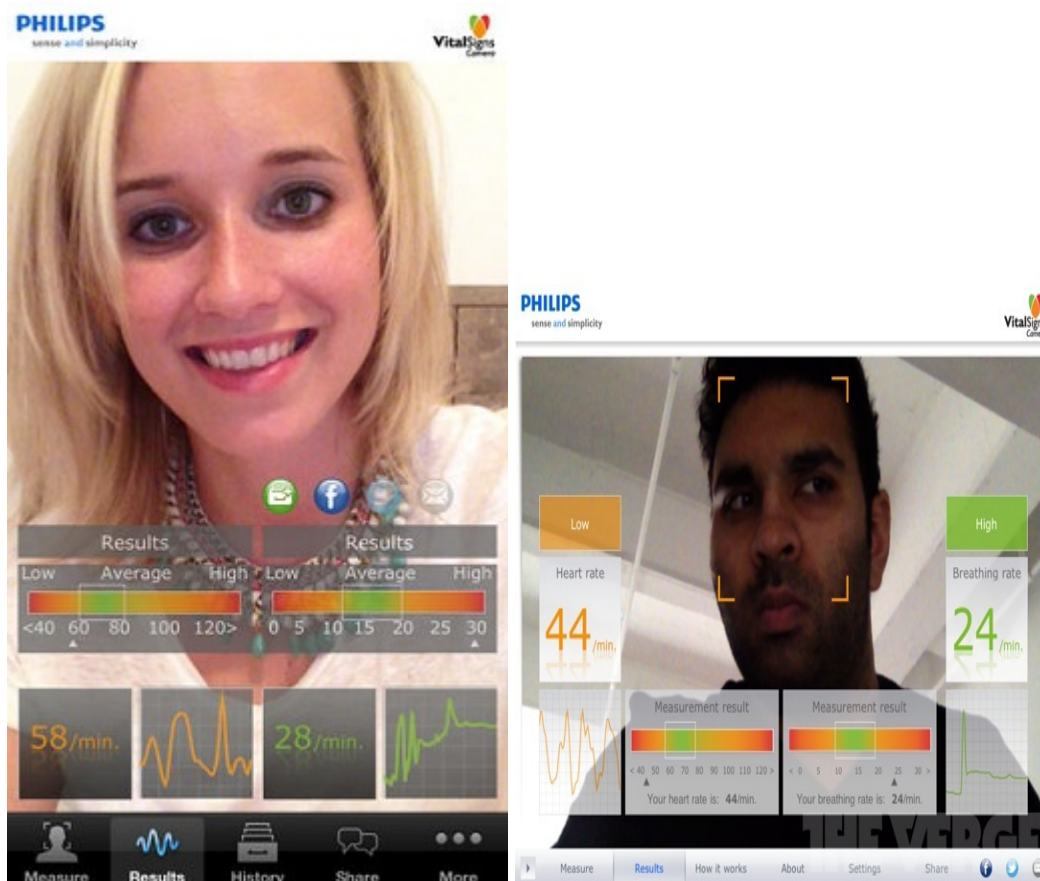


Рисунок 1.4 – Зображення вигляду додатка у роботі

Основні характеристики та функції Vital Signs Camera включають наступне:

- 1) вимірювання пульсу. Система аналізує зміни в кольорі та яскравості обличчя, які виникають від кровообігу під час кожного серцевого скорочення. Вона визначає пульс, вимірюючи час між кожними змінами в кольорі обличчя;
- 2) вимірювання частоти дихання. Крім вимірювання пульсу, система також може визначити частоту дихання, аналізуючи зміни в рухах грудей та живота пацієнта на відеозаписі;
- 3) неінвазивність та зручність. Однією з ключових переваг цієї системи є те, що вона не потребує носіння датчиків чи приладів, які контактують з тілом пацієнта. Вимірювання відбуваються абсолютно неінвазивно, що забезпечує зручність та комфорт для пацієнтів;

4) медичні застосування. Vital Signs Camera може бути використана в медичних закладах для моніторингу стану пацієнтів на палаті, відділенні інтенсивної терапії та інших відділеннях. Вона допомагає медичному персоналу в реальному часі відстежувати стан пацієнтів;

5) застосування в дослідженнях та спорті. Крім медичних застосувань, система також може бути використана у фізіологічних дослідженнях, спортивних тренуваннях та моніторингу фізичної активності для вимірювання пульсу та інших показників.

Vital Signs Camera представляє інноваційний підхід до моніторингу фізіологічних параметрів за допомогою відеозапису обличчя. Ця технологія дозволяє надавати точні та надійні вимірювання пульсу та частоти дихання без необхідності використання традиційних медичних приладів.

Висновки до розділу

Зважаючи на постійний розвиток технологій та наростаючий інтерес до здорового способу життя, аналіз частоти серцевих скорочень на основі обробки відеозапису обличчя є однією з передових галузей досліджень. Ця технологія може стати ключовим інструментом для надання користувачам доступу до свого фізіологічного стану та дозволити медичним фахівцям здійснювати нагляд за пацієнтами у режимі реального часу.

У багатьох сферах вимірювання пульсу та інших фізіологічних параметрів на основі відеозапису обличчя вже існують практичні застосування. В медицині, ця технологія може бути використана для моніторингу пацієнтів на відділенні інтенсивної терапії, під час операцій та реабілітації. В спорті, вона допомагає спортсменам відстежувати їхні показники під час тренувань та змагань. Для дослідників це може бути цінним інструментом для проведення фізіологічних досліджень та спостережень.

Однак, несвідомий розвиток цієї технології не обійдеться без викликів та обмежень. Найважливішим завданням є підвищення точності вимірювань, особливо в умовах поганої освітленості та руху об'єкта. Додатковою проблемою є необхідність узгодження з законами та стандартами в галузі медицини та захисту даних пацієнтів.

Проте, не зважаючи на ці виклики, інноваційні рішення, подібні до тих, які були представлені в огляді, свідчать про можливість революції у сфері медичного моніторингу та здоров'я. Вони відкривають нові можливості для моніторингу фізіологічних параметрів і роблять їх доступними для більш широкого кола користувачів.

В області аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя спостерігається значний прогрес у розробці та впровадженні інноваційних рішень та технологій. Такий підхід дозволяє надавати точний та неперервний моніторинг серцевої активності без потреби в дорогих та інвазивних медичних приладах. Огляд предметної області, аналіз наявних рішень та огляд деяких конкретних програмних продуктів підтверджують значний потенціал цієї технології у різних сферах, включаючи медицину, фізіологічні дослідження та фітнес.

FaceHeart, PulsePass та Vital Signs Camera - це лише кілька прикладів інноваційних рішень, які вже існують у цій галузі. Вони демонструють можливості вимірювання пульсу та інших фізіологічних параметрів на основі відеозапису обличчя та надають надію на більш доступні та зручні методи моніторингу здоров'я та фізичної активності.

Попередні дослідження та розробки вказують на потребу подальших досліджень та розвитку в цій області. Досягнення у вимірюванні пульсу на основі відеозапису обличчя можуть бути важливим кроком у вдосконаленні медичного моніторингу та фізіологічних досліджень, а також сприяти покращенню якості життя та здоров'я користувачів.

2 МОДЕЛІ ТА МЕТОДИ

2.1 Загальний огляд методів аналізу частоти серцевих скорочень

Загальний огляд методів аналізу частоти серцевих скорочень в контексті даної магістерської роботи розкриває різноманітні підходи та техніки, які застосовуються для точного та надійного вимірювання частоти серцевих скорочень на основі обробки відеозапису обличчя. Цей розділ надає глибокий огляд різних методів та їхніх переваг у вимірюванні серцевої активності.

Розглянемо детальніше кожен з методів аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя:

1) методи вимірювання пульсації на обличчі. Один з ключових методів полягає в вимірюванні пульсації судин на обличчі. Цей метод використовує оптичні техніки для реєстрації змін яскравості, кольірних відтінків або інших параметрів шкіри, що змінюються під впливом кровотоку. Зазвичай для цього використовуються камери з високою роздільною здатністю, які реєструють навіть незначні зміни. Методи вимірювання пульсації можуть бути контактними або безконтактними, залежно від використовуваних сенсорів та апаратури[11];

2) аналіз зображень методами комп'ютерного бачення. Комп'ютерне бачення використовується для аналізу відеозаписів обличчя. Методи обробки зображень включають в себе виявлення контурів обличчя, вимірювання змін яскравості на окремих точках або регіонах шкіри, а також відстеження руху судин під час пульсації. Такі методи можуть бути великодійними і складними в обробці, але вони надають важливі дані для визначення частоти серцевих скорочень. Методи комп'ютерного бачення можуть використовувати різні алгоритми, включаючи фільтрацію шуму, виокремлення даних про пульсацію та обчислення частоти серцевих скорочень[12]. Деякі дослідники використовують глибоке навчання для покращення точності аналізу;

3) спеціалізовані алгоритми для аналізу серцевої активності. Деякі дослідники розробляють спеціалізовані алгоритми для аналізу серцевої

активності на відеозаписах. Ці алгоритми можуть використовувати методи аналізу часових рядів, обробки сигналів та статистичні методи для визначення частоти серцевих скорочень. Вони можуть бути оптимізовані під конкретні умови та завдання дослідження;

4) переваги та обмеження методів аналізу частоти серцевих скорочень. Під час розгляду методів важливо враховувати їхні переваги та обмеження. Деякі методи можуть бути точнішими, але вимагати дорогого обладнання. Інші можуть бути менш точними, але більш доступними та зручними для практичного використання. Також важливо враховувати чутливість методів до зовнішніх факторів, таких як освітлення та рух;

5) застосування методів аналізу частоти серцевих скорочень. Завдяки розвитку методів аналізу частоти серцевих скорочень на основі відеозапису обличчя, вони знаходять застосування в різних областях. Наприклад, в медицині ці методи можуть бути використані для надання медичної допомоги пацієнтам, у спортивному тренуванні — для моніторингу фізичного стану спортсменів, а в психофізіологічних дослідженнях — для вивчення реакцій на стрес та інші психофізіологічні явища[13].

Цей огляд методів дозволяє краще розібратися у різноманітних підходах до аналізу частоти серцевих скорочень на основі відеозапису обличчя та визначити їхню застосовність у відповідних областях дослідження і практики.

2.2 Методи для обробки відеозапису обличчя для виявлення частоти серцевих скорочень

Методи для обробки відеозапису обличчя для виявлення частоти серцевих скорочень охоплюють кілька важливих етапів, таких як видобуток обличчя з відеозапису, відстеження руху та пульсації обличчя. Ці етапи є ключовими у розробці автоматизованих систем для вимірювання серцевої активності на основі відеоданих:

1) видобуток обличчя з відеозапису. Перший крок у виявленні частоти серцевих скорочень полягає в видобутку обличчя з відеозапису. Цей процес містить використання алгоритмів комп'ютерного бачення для виявлення та локалізації обличчя на кадрі відеозапису. Це може вимагати використання методів, які розпізнають особливості обличчя, такі як очі, ніс, та рот, та дозволяють виділити його від інших частин кадру[14];

2) відстеження руху та пульсації обличчя. Після видобутку обличчя наступним кроком є відстеження руху та пульсації обличчя. Для цього використовуються алгоритми, які дозволяють виявити зміни яскравості та колірних відтінків на поверхні шкіри обличчя, що виникають внаслідок кровотоку, а саме через пульсацію судин[15].

Відстеження руху дозволяє визначити позицію обличчя на кожному кадрі відеозапису і виміряти зміни параметрів шкіри, які відповідають за пульсацію. З цими даними можна побудувати часовий ряд змін і визначити частоту серцевих скорочень.

Деякі методи використовують спеціалізовану апаратуру, таку як камери зі спеціальними фільтрами для підвищення чутливості до змін кольору на шкірі. Інші методи можуть вимагати обробки великої кількості кадрів і використання алгоритмів обробки сигналів для аналізу отриманих даних.

Ці методи спільно допомагають створити системи для точного вимірювання частоти серцевих скорочень на основі відеозапису обличчя.

2.2.1 Видобуток обличчя з відеозапису

Видобуток обличчя з відеозапису є важливим етапом в аналізі частоти серцевих скорочень на основі обробки відеозапису обличчя. Цей процес передбачає виявлення та локалізацію обличчя на кожному кадрі відеозапису, щоб подальше вимірювання пульсації та аналіз серцевої активності могли бути проведені точно та надійно.

Основні етапи видобутку обличчя з відеозапису включають наступне:

1) вибір методу виявлення обличчя. Існує кілька методів для виявлення обличчя на відеозаписі. До них входять методи, що базуються на глибинному навчанні, каскадних класифікаторах, а також детекторах обличчя, які використовують признаки, такі як очі, ніс, та рот. Вибір конкретного методу залежить від обраної системи та вимог до точності та продуктивності[16];

2) виявлення обличчя на кожному кадрі. Обробка відеозапису проводиться кадр за кадром. На кожному кадрі використовуються алгоритми для виявлення обличчя в зображенні. Це може включати в себе аналіз кожного пікселя зображення, пошук певних візуальних ознак, які характеризують обличчя, і визначення його положення та розміру[16];

3) відстеження обличчя в часі. Однак лише виявлення обличчя на окремих кадрах недостатньо для аналізу серцевої активності. Оскільки обличчя може змінювати положення, розмір і орієнтацію під час відеозапису, важливо відстежувати обличчя в часі. Для цього використовують алгоритми відстеження руху, які дозволяють визначити нове положення обличчя на кожному кадрі[16];

4) оцінка точності видобутку обличчя. Після виявлення та відстеження обличчя важливо оцінити точність видобутку. Це включає в себе перевірку, наскільки добре обличчя відстежується в часі, і якість видобутку на різних кадрах. Низька точність може призвести до неправильних вимірювань пульсації;

5) побудова регіону для вимірювання пульсації. Після видобутку обличчя і відстеження руху, визначається регіон, на якому буде виміряні пульсації. У нашому випадку буде взята область шкіри на обличчі, де пульсація судин найбільш помітна, в районі лоба або щоки, ніс та очі[17].

Цей процес видобутку обличчя з відеозапису є важливим кроком у створенні систем для аналізу частоти серцевих скорочень. Від правильного видобутку обличчя залежить точність та надійність подальшого аналізу серцевої активності на основі відеозапису обличчя.

Основні методи, які були взяті за базу в реалізації видобутку даних обличчя з відеозапису:

- MTCNN;
- DeepSORT.

Детальніше про кожен обраний метод та їх особливості буде розказано у наступних підрозділах роботи.

2.2.1.1 MTCNN

Багатозадачні каскадні згорткові мережі (MTCNN) — це структура, розроблена як рішення для виявлення та вирівнювання обличчя. Процес складається з трьох етапів згорткових мереж, які здатні розпізнавати обличчя та орієнтири, такі як очі, ніс і рот [1].

На першому етапі він використовує неглибоку CNN для швидкого створення вікон кандидатів. На другому етапі він уточнює запропоновані вікна кандидатів через більш складну CNN. І, нарешті, на третьому етапі він використовує третій CNN, складніший за інші, для подальшого уточнення результату та вихідних орієнтирів обличчя [1].

Етап 1: Мережа пропозицій (P-Net).

Цей перший етап є повністю згортковою мережею (FCN). Різниця між CNN і FCN полягає в тому, що повністю згорткова мережа не використовує щільний шар як частину архітектури. Ця мережа пропозицій використовується для отримання вікон-кандидатів і їх векторів регресії обмежувальної рамки [1].

Регресія обмежувальної рамки — популярний метод прогнозування локалізації рамок, коли метою є виявлення об'єкта певного попередньо визначеного класу, у цьому випадку обличчя. Після отримання векторів обмежувальної рамки виконується деяке уточнення для об'єднання областей, що перекриваються. Остаточним результатом цього етапу є всі вікна кандидатів після уточнення для зменшення обсягу кандидатів [1] (рис. 2.1).

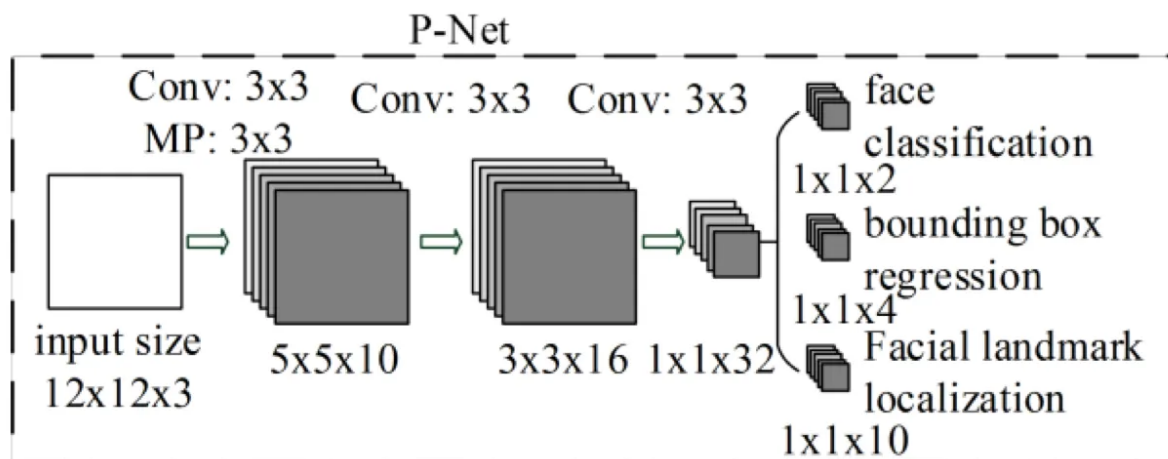


Рисунок 2.1 – Схема мережі P-Net

Етап 2: Уточнена мережа (R-Net).

Усі кандидати з P-Net подаються в Refine Network. Зверніть увагу, що ця мережа є CNN, а не FCN, як попередня, оскільки на останньому етапі архітектури мережі є щільний шар. R-Net додатково зменшує кількість кандидатів, виконує калібрування за допомогою регресії обмежувальної рамки та використовує немаксимальне придушення (NMS) для об'єднання кандидатів, що перекриваються [1].

R-Net виводить незалежно від того, є вхід обличчям чи ні, 4-елементний вектор, який є обмежувальною рамкою для обличчя, і 10-елементний вектор для локалізації обличчя [1] (рис. 2.2).

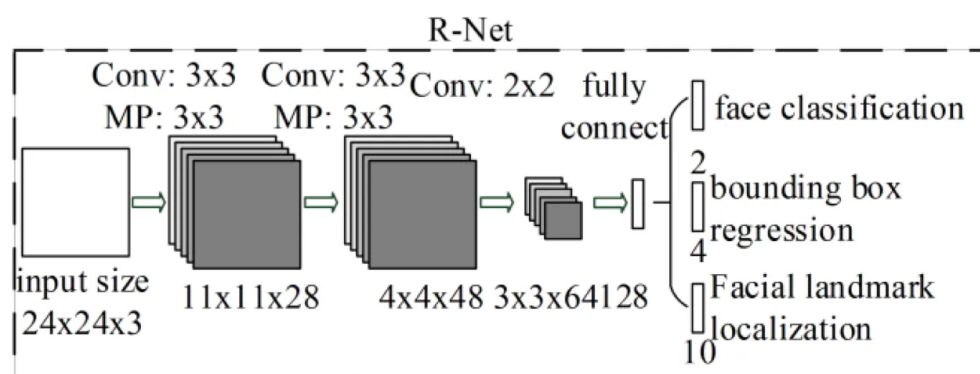


Рисунок 2.2 – Вигляд мережі R-Net

Етап 3: вихідна мережа (O-Net).

Цей етап подібний до R-Net, але ця вихідна мережа спрямована на більш детальний опис обличчя та виведення п'яти орієнтирів обличчя для очей, носа та рота [1] (рис. 2.3).

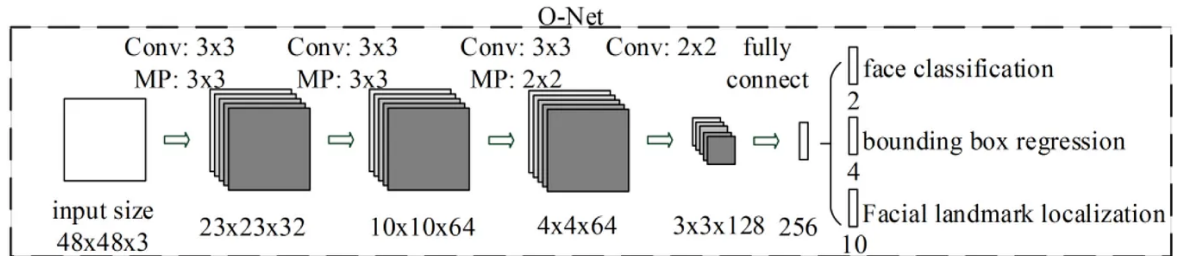


Рисунок 2.3 – Вигляд мережі O-Net

Завдання мережі полягає в тому, щоб вивести три речі: класифікацію обличчя/не обличчя, регресію обмежувальної рамки та локалізацію обличчя [1]:

- 1) класифікація обличчя: це проблема двійкової класифікації, яка використовує втрату крос-ентропії(рис. 2.4) [1]:

$$L_i^{det} = - (y_i^{det} \log(p_i) + (1 - y_i^{det}) (1 - \log(p_i)))$$

$y_i^{det} \in \{0, 1\}$ ground truth label
 p_i probability produced by the network

Рисунок 2.4 – Перехресна втрата ентропії

2) регресія обмежувальної рамки: метою навчання є проблема регресії. Для кожного вікна-кандидата обчислюється зсув між кандидатом і найближчою правдою. Для цього завдання використовуються евклідові втрати(рис. 2.5) [1]:

$$L_i^{\text{box}} = \|\hat{y}_i^{\text{box}} - y_i^{\text{box}}\|_2^2$$

↑
 target obtained
 from network

↗
 ground truth
 coordinate

Рисунок 2.5 – Формула евклідової втрати

3) локалізація орієнтирів обличчя: локалізація орієнтирів обличчя формулюється як задача регресії, в якій функцією втрат є евклідова відстань(рис. 2.6) [1]:

$$L_i^{\text{landmark}} = \|\hat{y}_i^{\text{landmark}} - y_i^{\text{landmark}}\|_2^2$$

Рисунок 2.6 – Формула евклідової відстані

MTCNN ефективний у виявленні облич у реальному часі, має високу точність та може працювати з різними масштабами облич у зображенні. Цей метод широко використовується для різних завдань комп'ютерного зору, включаючи розпізнавання облич, відстежування руху, аналіз емоцій та багато іншого [1].

2.2.1.2 DeepSORT

Deep SORT (Simple Online and Realtime Tracking) — це алгоритм, який використовується для відстеження кількох об'єктів у відеопотоках. Це

розширення алгоритму SORT (Simple Online and Realtime Tracking), який використовує фільтр Калмана для відстеження об'єктів. Deep SORT включає метрику глибокої асоціації на основі особливостей зовнішнього вигляду, отриманих глибокою згортковою нейронною мережею. Це дозволяє алгоритму обробляти ситуації, коли об'єкти можуть тимчасово зникати або закриватися у відеопотоці. Deep SORT також включає призначення ідентифікаторів для відстеження окремих об'єктів у кількох кадрах, що має вирішальне значення для таких програм, як спостереження, автономні транспортні засоби та взаємодія людини з комп'ютером [2]. Алгоритм виконує двоетапний підхід: спочатку генерує виявлення об'єктів, а потім пов'язує ці виявлення з існуючими треками. Загалом, Deep SORT продемонстрував значні покращення в точності та надійності відстеження порівняно з традиційними алгоритмами відстеження, що робить його цінним інструментом у програмах комп'ютерного зору та штучного інтелекту [2].

Deep SORT складається з 4 ключових компонентів (рис. 2.7).

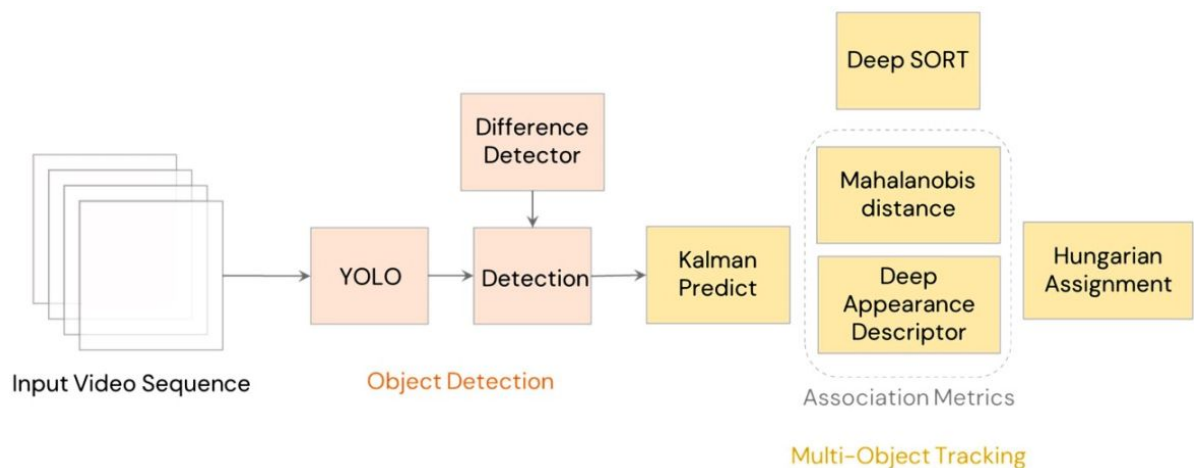


Рисунок 2.7 – Архітектура Deep SORT

1) виявлення та виділення ознак. Глибоке СОПУВАННЯ починається з виявлення об'єктів, часто з використанням згорткової нейронної мережі (CNN), як-от YOLO (You Only Look Once), для ідентифікації об'єктів у кадрі. Кожне виявлення асоціюється з дескриптором зовнішнього вигляду великої

розмірності, витягнутим іншим CNN. Ці дескриптори кодують зовнішній вигляд виявлених об'єктів і використовуються для зіставлення [2];

2) фільтр Калмана в передбаченні стану Подібно до SORT, Deep SORT використовує фільтр Калмана для прогнозування стану. Стан об'єкта зазвичай включає його положення, швидкість і прискорення. Фільтр Калмана передбачає стан кожного об'єкта в поточному кадрі на основі його останнього відомого стану, враховуючи динаміку руху об'єкта [2];

3) зв'язок даних із глибокими показниками зовнішнього вигляду. Асоціація даних у Deep SORT значно відрізняється від попередника. Замість того, щоб покладатися виключно на IoU, він використовує комбінацію інформації про рух і функцій зовнішнього вигляду. Угорський алгоритм виконує зіставлення на основі матриці вартості, яка враховує як відстань Махаланобіса для узгодженості руху, так і косинусну відстань для подібності зовнішнього вигляду. Використання метрики глибокої асоціації дозволяє Deep SORT продовжувати відстеження через короткі періоди оклюзії [2];

4) керування колією. Deep SORT керує життєвими циклами треків за допомогою кількох евристик. Він вводить такі поняття, як підтвердження треку, коли трек підтверджується лише після того, як його виявлено в кількох послідовних кадрах, тим самим зменшуючи помилкові спрацьовування. Існує також віковий параметр для видалення старих треків, які не були виявлені останнім часом, гарантуючи, що система підтримує лише активні треки [2].

Переваги Deep SORT Інтеграція Deep SORT функцій зовнішнього вигляду з інформацією про рух дає кілька переваг:

1) стійкість до оклюзії: за допомогою дескрипторів зовнішнього вигляду Deep SORT може повторно ідентифікувати об'єкти після того, як їх було закрито;

2) розрізнення схожих об'єктів: дескриптори зовнішнього вигляду допомагають розрізняти схожі на вигляд об'єкти;

3) продуктивність у режимі реального часу: попри додаткову складність, Deep SORT може працювати майже в режимі реального часу завдяки ефективним механізмам вилучення ознак і зв'язування;

4) гнучкість: його можна поєднувати з будь-яким сучасним детектором [2].

2.2.2 Відстеження руху та пульсації обличчя

Відстеження руху та пульсації обличчя є критичним етапом в аналізі частоти серцевих скорочень на основі обробки відеозапису обличчя. Цей процес дозволяє визначити зміни яскравості та колірних відтінків на поверхні шкіри обличчя, що виникають внаслідок кровотоку, і відстежити їх у часі для вимірювання серцевої активності[9]. Основні аспекти цього процесу включають наступне:

1) вимірювання змін яскравості та колірних відтінків. Після сегментації обличчя використовуються алгоритми для вимірювання змін яскравості та колірних відтінків в обраному регіоні. Зміни цих параметрів пов'язані з кровоотоком, оскільки пульсація судин впливає на кількість крові, яка знаходиться під поверхнею шкіри. Зазвичай для цього використовують алгоритми комп'ютерного бачення, які вимірюють інтенсивність світла та колірні характеристики пікселів у вибраному регіоні[9];

2) аналіз даних пульсації. Отримані дані змін яскравості та колірних відтінків після сегментації використовуються для аналізу пульсації. Пульсація спричинює циклічні зміни у змінах параметрів шкіри, що можна виявити у вигляді часових рядів. Зазвичай використовуються методи обробки сигналів, такі як кореляція, для визначення частоти серцевих скорочень і амплітуди пульсації[10];

3) фільтрація шуму та підсилення сигналу. Даний процес може бути супроводжений обробкою сигналу для фільтрації шуму та підсилення пульсації.

Це допомагає покращити точність вимірювань та виділити корисний сигнал серцевих скорочень від шуму[10];

4) вимірювання частоти серцевих скорочень. За допомогою аналізу пульсації можна визначити частоту серцевих скорочень. Це може бути зроблено, обчислюючи кількість пульсаційних подій за певний період часу.

В результаті відстеження руху та пульсації обличчя можна точно виміряти частоту серцевих скорочень на основі відеозапису обличчя. Цей метод знаходить застосування в медицині, спорті, психофізіологічних дослідженнях та інших галузях, де точність вимірювань є важливою.

Для програмного забезпечення, яке спрямоване на аналіз частоти пульсу на основі відеозапису обличчя, ефективним методом може бути методи оптичного відстеження (Optical Tracking), аналіз змін відтінку шкіри, аналіз змін освітлення. Ці методи можуть бути ефективними для вимірювання пульсації на основі відеозапису обличчя. Детальніше про кожен з даних методів та їх особливості буде розказано у подальших підрозділах роботи.

2.2.2.1 Спектральний аналіз

Спектральний аналіз — це техніка, яка оцінює потужність. У літературі існує багато суперечностей щодо того, як потужність виражається в спектральному аналізі, але в Dataview потужність — це енергія на одиницю часу, виражена як середній квадрат амплітуди сигналу при різних частот. Сигнал спочатку оцифровується. Аналогові (з безперервним часом і безперервною амплітудою) сигнали оцифровуються шляхом пропускання їх через аналого-цифровий перетворювач (АЦП) [3].

Це перетворює сигнал на ряд чисел, що представляють амплітуду сигналу в дискретних інтервалах часу, встановлених частотою дискретизації АЦП, щоб створити послідовність значень вибірки, які потім пропускаються через алгоритм швидкого перетворення Фур'є (ШПФ). Алгоритм створює набір

рівновіддалених частотних смуг у діапазоні від 0 (DC) до частоти Найквіста (половина частоти дискретизації) і повідомляє нам, яка потужність є в кожній з цих смуг. Кількість смуг частот визначається кількістю вибірок, переданих програмі ШПФ – смуг рівно вдвічі менше, ніж вибірок, тому чим більша кількість вибірок передається в алгоритм, тим вище частотна роздільна здатність. Алгоритм ШПФ працює лише з порціями даних, довжина яких є степенем двох (наприклад, 32, 64, 128 тощо). Короткі сигнали можуть бути доповнені нулем, щоб заповнити їх до довжини, необхідної для певної роздільної здатності, але це, звичайно, зменшує загальні значення потужності, які повідомляє FFT [3].

Стандартний спектральний аналіз об'єднує всі зразки сигналу та створює єдиний двовимірний графік, що показує середню потужність. Часто інформативніше побудувати логарифм потужності, а не її лінійне значення, оскільки це полегшує виявлення низької (але все одно значної) силові компоненти [3].

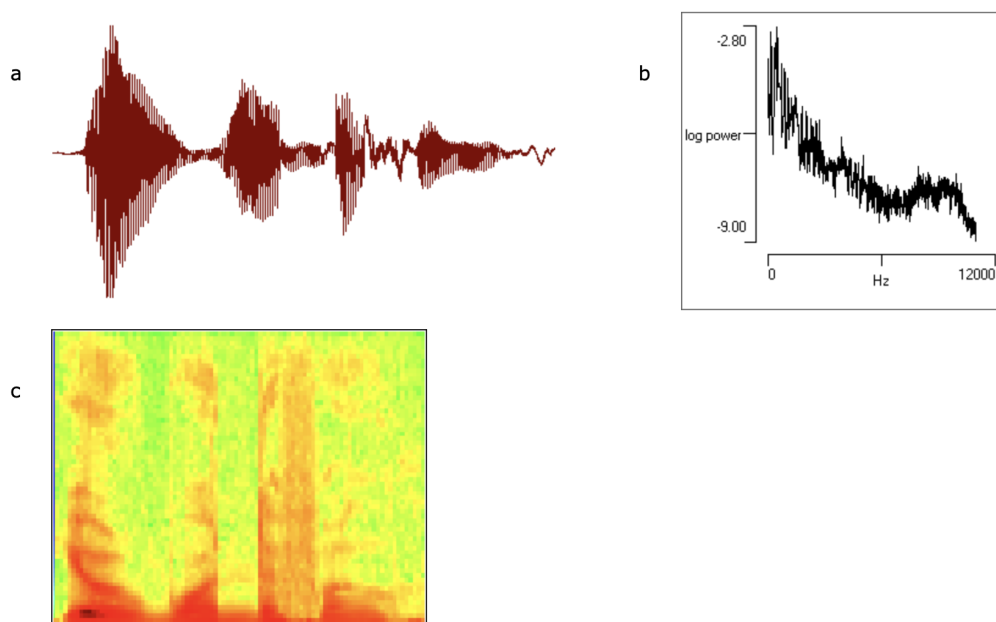


Рисунок 2.8 – Спектральний аналіз голосового файлу. а. Необроблений сигнал, отриманий із мікрофонного запису розмови. б. Спектр потужності

всього файлу. в. Спектрограма файлу з часом на осі x, частотою на осі y та кольором потужності [3]

У контексті аналізу частоти серцевих скорочень на основі відеозапису обличчя, спектральний аналіз може бути корисним у таких аспектах:

1) виділення частоти пульсації. Спектральний аналіз дозволяє виявити частоту пульсації серця шляхом аналізу частотного складу сигналу, отриманого зі змін відтінку шкіри обличчя. Це допоможе ідентифікувати частоту серцевих скорочень на основі певних частотних компонентів сигналу;

2) фільтрація сигналу. Спектральний аналіз може бути використаний для фільтрації сигналу, щоб видалити шуми та інші непотрібні частоти, залишивши лише ті, що відповідають пульсації серця;

3) оцінка регулярності пульсації. Частотний аналіз може допомогти в оцінці регулярності чи неправильності серцевого ритму, адже відхилення у частотному діапазоні можуть вказувати на аномалії у ритмі серця.

Застосування спектрального аналізу в реальному часі до сигналів, отриманих зі змін відтінку шкіри обличчя, може допомогти виявити та аналізувати частоту пульсації серця, що є важливим у реалізації ПЗ для безконтактного визначення серцевих скорочень.

2.2.2.2 Адаптивне гістограмне підвищення (АНЕ)

Вирівнювання гістограми — це математичний прийом для розширення динамічного діапазону гістограми. Іноді гістограма охоплює короткий діапазон, шляхом вирівнювання діапазон гістограми розширюється. У цифровій обробці зображень контрастність зображення посилюється за допомогою цієї техніки [4].

Адаптивне вирівнювання гістограми – це техніка цифрової обробки зображень, яка використовується для підвищення контрастності зображень. Він відрізняється від звичайного вирівнювання гістограми тим, що адаптивний

метод посилює контраст локально. Він ділить зображення на окремі блоки та обчислює вирівнювання гістограми для кожного розділу. Таким чином, АНЕ обчислює багато гістограм, кожна з яких відповідає окремому розділу зображення. Це покращує локальний контраст і чіткість країв у всіх окремих областях зображення [4].

Переваги:

- він обчислює вирівнювання гістограми окремих ділянок зображення;
- він зберігає краї в окремих областях зображення;
- він підсилює контраст локально [4].

Недолік:

- АНЕ надмірно посилює шум у відносно однорідних областях зображення [4].

У контексті реалізації ПЗ для аналізу частоти серцевих скорочень на основі відеозапису обличчя АНЕ може корисним за таких причин:

1) покращення контрастності шкіри: АНЕ може допомогти покращити контрастність областей шкіри на зображенні. Це важливо для точного виділення та відстеження областей шкіри, які використовуються для аналізу серцевих скорочень;

2) зменшення впливу нерівномірного освітлення: АНЕ може допомогти у зменшенні впливу нерівномірного освітлення на зображення шкіри. Це дозволить покращити стабільність аналізу обличчя у відеопотоці, особливо в умовах змінного освітлення;

3) підвищення якості обробки зображень: Застосування АНЕ перед аналізом дозволить покращити якість зображень шкіри, що може поліпшити точність виявлення серцевих скорочень.

Це один із методів обробки зображень, який може бути корисним у покращенні зображень шкіри та відстеженні обличчя для аналізу частоти серцевих скорочень.

Висновки до розділу

У другому розділі роботи було розглянуто основні методи, моделі та алгоритми для реалізації бажаного ПЗ.

У контексті аналізу частоти серцевих скорочень на основі відеозапису обличчя, спектральний аналіз та адаптивне гістограмне підвищення (АНЕ) виявляються надзвичайно потужними інструментами. Спектральний аналіз дозволяє виявляти та аналізувати частотні характеристики сигналу, зокрема пульсацію серця, через аналіз частотного складу сигналу, отриманого змінами відтінку шкіри обличчя. Водночас, АНЕ допомагає покращити контрастність та якість зображення шляхом підвищення контрастності областей шкіри, що є важливим для точного відстеження та аналізу обраних зон для вимірювання серцевих скорочень.

Обидва методи, спектральний аналіз та АНЕ, є ключовими у створенні програмного забезпечення для безконтактного визначення частоти серцевих скорочень. Їх поєднання дозволяє поліпшити точність вимірювань та аналізу, забезпечуючи значне полегшення у визначенні серцевих показників. Застосування цих методів має значний потенціал у медицині, спорті, психофізіологічних дослідженнях та інших галузях, де точність та надійність вимірювань є критичними. Їх ефективність полягає у здатності покращувати якість зображень та забезпечувати стабільні та точні результати у вимірюванні частоти серцевих скорочень на основі відеозапису обличчя.

DeepSORT - це інноваційна модель відстеження об'єктів у відеопотоках, яка базується на сполученні алгоритму SORT та глибокого навчання. Цей метод дозволяє виявляти та відстежувати об'єкти, використовуючи особливості

зовнішнього вигляду, отримані глибокою нейронною мережею. Це особливо важливо при аналізі відеопотоків, де об'єкти можуть зникати або перекриватися.

Поєднуючи DeepSORT зі спектральним аналізом, можна досягти додаткових переваг у вимірюванні частоти серцевих скорочень. Використання спектрального аналізу для виявлення пульсації на основі змін відтінку шкіри обличчя дозволить точно визначити частоту серцевих скорочень. Поєднуючи цей підхід з можливістю DeepSORT відстежувати обличчя від кадру до кадру, можна стабілізувати процес вимірювання серцевих показників, навіть при зміні освітлення або перекриванні об'єктів.

Такий комплексний підхід дозволяє збирати інформацію про пульсацію обличчя, аналізуючи його частотний склад та забезпечуючи високу точність вимірювання. Використання DeepSORT у поєднанні з аналізом спектра змін відтінку шкіри обличчя робить систему ефективною та надійною для безконтактного визначення серцевих показників.

3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Для розробки алгоритмічного та програмного забезпечення для визначення частоти серцевих скорочень вибір пав на засоби розробки:

- Python;
- OpenCV;
- NumPy;
- SciPy;
- HTML, CSS, JavaScript.

Python - це високорівнева, інтерпретована мова програмування загального призначення, яка відома своєю простотою, читабельністю та гнучкістю. Ця мова програмування була створена Гвідо ван Россумом і вперше випущена у 1991 році. Однією з ключових особливостей Python є його акцент на простоту синтаксису, що робить код більш зрозумілим та доступним для розробників[5].

Деякі ключові особливості Python:

- 1) простота читання та вивчення коду: Python відомий своєю читабельністю, що робить його придатним для новачків у програмуванні;
- 2) широкий спектр застосувань: Використовується у веб-розробці, наукових дослідженнях, штучному інтелекті, аналізі даних, розробці ігор та багатьох інших областях;
- 3) багатофункціональність: Має велику кількість бібліотек та модулів, що полегшують роботу з різноманітними завданнями;
- 4) інтерпретованість: Код Python виконується рядок за рядком, що дозволяє вивчати результати кожної операції негайно;
- 5) крос-платформеність: Розроблено для роботи на різних операційних системах, таких як Windows, macOS, Linux тощо;
- 6) спільнота та підтримка: Python має велику та активну спільноту розробників, що призводить до швидкого оновлення та розвитку мови.

Це дозволяє Python бути однією з найпопулярніших мов програмування для початківців, а також широко використовуватися у великих проєктах та різноманітних областях розробки програмного забезпечення.

OpenCV (Open Source Computer Vision Library) - це бібліотека відкритого програмного забезпечення, яка спеціалізується на комп'ютерному зорі та обробці зображень і відео. Вона надає широкий спектр інструментів для виконання завдань комп'ютерного зору, таких як виявлення об'єктів, відстеження руху, розпізнавання облич, аналіз поведінки та багато інших[6].

OpenCV підтримує багато мов програмування, включаючи C++, Python, Java та інші. Ця бібліотека має велику спільноту користувачів і використовується в різних галузях, включаючи дослідження, розробку продуктів, медицину, безпеку, мобільні додатки та багато інших.

OpenCV має багато модулів та функцій, які спрощують обробку зображень, включаючи фільтрацію, виявлення контурів, розмиття, відстеження об'єктів, розпізнавання облич, калібрування камери та багато інших. Вона надає можливість працювати як зі статичними, так і з потоковими зображеннями або відеопотоками.

Ця бібліотека є потужним інструментом для реалізації алгоритмів машинного зору, навчання глибоких мереж та обробки зображень у багатьох проєктах та дослідженнях, що пов'язані з комп'ютерним зором та штучним інтелектом.

NumPy - це бібліотека для мови програмування Python, яка надає підтримку для великих масивів та математичних функцій для їх операцій. Основна структура даних у NumPy - це масиви (або багатовимірні масиви), які є ефективними для зберігання та обробки даних.

NumPy надає функції для виконання різноманітних операцій над масивами, включаючи математичні, логічні, маніпуляції з формою та індексацію даних. Вона також має підтримку для векторизації операцій, що дозволяє виконувати операції на цілих масивах даних, уникнувши явних циклів,

що часто призводять до покращення продуктивності та швидкості обробки даних.

Оскільки NumPy побудована на мові Python, вона також інтегрується з іншими бібліотеками Python, такими як SciPy (для наукових обчислень), Matplotlib (для візуалізації даних) та Pandas (для роботи з даними у вигляді таблиць). NumPy використовується широкою спільнотою дослідників, науковців та розробників для обробки даних, наукових обчислень, машинного навчання та інших областей.

SciPy - це відкрите програмне забезпечення для наукових обчислень та технічних обчислювальних завдань, яке базується на мові програмування Python. Воно включає в себе бібліотеки для роботи з чисельними методами, оптимізацією, обробкою сигналів, обробкою зображень, статистикою та багато іншого.

SciPy поширюється разом з NumPy, що є основною бібліотекою для роботи з масивами та обчисленнями у Python. SciPy надає широкий набір функцій для розв'язання різних математичних завдань, включаючи інтегрування, розв'язання диференціальних рівнянь, оптимізацію, роботу зі статистикою та інші задачі наукових обчислень.

Ця бібліотека стала важливим інструментом для наукових досліджень, інженерії, фінансів та інших галузей, де потрібні потужні інструменти обчислень та обробки даних. Її легко використовувати та розвивати завдяки відкритому джерелу та активній спільноті розробників.

HTML (HyperText Markup Language) - це мова розмітки, яка використовується для створення структури та вмісту веб-сторінок. Вона визначає структуру елементів на сторінці, таких як заголовки, абзаци, посилання, зображення та інші елементи, що використовуються для відображення інформації у браузері.

CSS (Cascading Style Sheets) - це мова опису стилів, яка використовується для задання вигляду та оформлення вмісту, який написаний мовою HTML. CSS

визначає такі аспекти, як кольори, шрифти, розміщення, розміри елементів, міжрядкові відступи та інші естетичні характеристики веб-сторінок.

JavaScript - це мова програмування, яка використовується для створення динамічного та інтерактивного веб-змісту. Вона дозволяє взаємодіяти з елементами HTML та CSS, змінювати їх вміст, стилі та поведінку, виконувати обробку подій, створювати анімації, робити запити до сервера та багато іншого.

Ці три технології - HTML, CSS та JavaScript - є основою веб-розробки та дозволяють створювати різноманітні та функціональні веб-сторінки та додатки.

HTML (HyperText Markup Language) - це основна мова розмітки, що використовується для створення структури веб-сторінок. Вона визначає скелет сторінки, включаючи заголовки, абзаци, списки, посилання та інші елементи для відображення тексту та медіа-вмісту у браузері.

CSS (Cascading Style Sheets) - це мова стилів, яка використовується для стилізації та оформлення веб-сторінок, щоб змінювати їх вигляд та розміщення елементів. CSS дозволяє контролювати розміри, кольори, шрифти, відступи та інші візуальні аспекти сторінок.

JavaScript - це мова програмування, що використовується для створення динамічної та інтерактивної взаємодії на веб-сторінках. Вона дозволяє змінювати контент сторінок у реальному часі, обробляти події, створювати анімації та взаємодіяти з користувачем без перезавантаження сторінки.

Ці три технології - HTML, CSS та JavaScript - працюють разом, щоб допомогти розробникам створювати структуровані, естетично привабливі та функціональні веб-сторінки та додатки. Вони є основою веб-розробки та є ключовими для будь-якого веб-проекту.

3.2 Обґрунтування вибраних технологій

Обираючи HTML, CSS та JavaScript для веб-розробки, є декілька обґрунтувань:

- стандартність і широке використання: HTML, CSS та JavaScript є основними технологіями веб-розробки, які підтримуються всіма сучасними браузерами. Це стандарти, що широко використовуються, що робить їх відмінним вибором для розробки веб-сайтів та додатків;
- кожна має свою функцію: HTML відповідає за структуру сторінки, CSS - за вигляд і стиль, а JavaScript - за динаміку та взаємодію. Ця розділеність забезпечує чітке розмежування завдань та полегшує розробку;
- підтримка спільноти: Існує велика спільнота розробників, які працюють з цими технологіями, що означає, що завжди є доступ до різноманітних ресурсів, документації та підтримки;
- крос-платформеність: HTML, CSS та JavaScript працюють на різних платформах і пристроях. Вони є універсальними технологіями, які можна використовувати для розробки веб-додатків для різних пристроїв і операційних систем;
- можливості розширення: За допомогою бібліотек та фреймворків, які підтримуються спільнотою, можна значно розширити можливості цих технологій. Наприклад, бібліотеки для JavaScript, такі як React або Angular, дозволяють швидко створювати складні додатки;
- швидкість розробки: Завдяки своїй простоті, ці технології дозволяють швидко розробляти та тестувати веб-сторінки, що робить їх ідеальними для проектів різного масштабу.

Ці технології є основою веб-розробки та мають відмінні можливості для створення різноманітних та функціональних веб-додатків та сторінок.

Використання Python, OpenCV, NumPy та SciPy для обчислення пульсу на основі відеозапису обличчя — це розумний вибір з кількох причин:

1) OpenCV: Це бібліотека комп'ютерного зору, яка має широкі можливості з обробки відео, зображень та розпізнавання облич. З його допомогою можна витягати кадри з відео, визначати розмірність та розташування облич та обробляти їх для подальшого аналізу пульсу;

2) NumPy: Це потужна бібліотека для наукових обчислень в Python, яка надає підтримку для роботи з масивами даних та математичними операціями. Вона корисна для обробки інтенсивних обчислень, які можуть бути необхідними для аналізу великих об'ємів даних з відео;

3) SciPy: Це додаткова бібліотека, яка містить різноманітні інструменти для наукових обчислень, включаючи статистичний аналіз, оптимізацію, обробку сигналів та багато іншого. Для аналізу пульсу можна використовувати функціонал SciPy для обробки сигналів та спектрального аналізу.

Комбінація цих бібліотек дозволяє вам зчитувати відео, витягати дані про обличчя, обробляти їх для отримання сигналу пульсу, виконувати аналіз спектра частот для визначення пульсу та оцінювати пульсовий ритм на основі даних з обличчя в відео.

3.3 Архітектура програмного забезпечення

3.3.1 USE-CASE діаграма

У системі передбачено два типи акторів:

- користувач;
- система.

Користувач може виконувати завдання, такі як завантаження відео, запуск аналізу, перегляд результатів. Система представляє програмну частину, яка відповідає за обробку відеоданих, визначення пульсу та надання результатів користувачеві. Також використовує алгоритми обробки зображень та аналізу сигналів для обчислення пульсу на основі кольору обличчя.

У таблиці 3.1 міститься детальний опис варіантів використання.

Таблиця 3.1 – Актори та варіанти використання

Актор	Варіант використання
Користувач	Вхід у систему
Користувач	Увімкнути камеру
Користувач	Запуск аналізу
Користувач	Перегляд результатів
Система	Обробка відеоданих
Система	Визначення пульсу
Система	Відображення результатів

На рисунку (3.1) можна побачити діаграму варіантів використання.

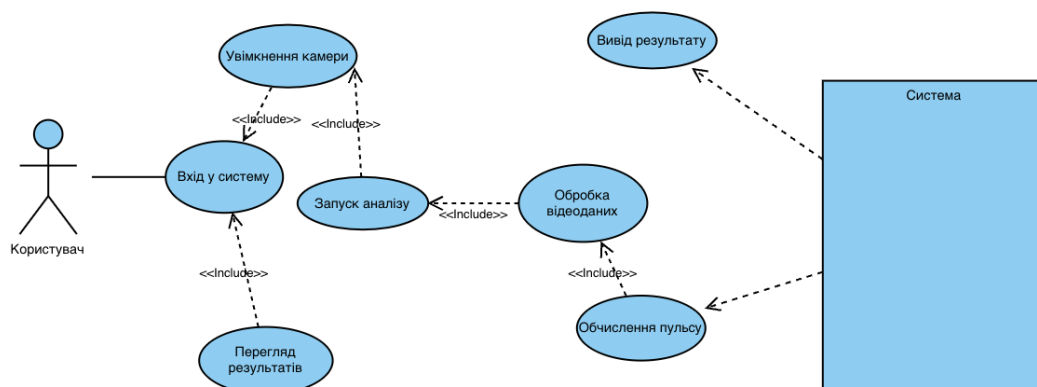


Рисунок 3.1 – USE-CASE діаграма

Таким чином, дана діаграма варіантів використання моделює основні сценарії, які включають в себе взаємодію користувачів з системою обчислення пульсу на основі відеозапису обличчя.

1) Запуск аналізу пульсу:

- користувач завантажує відео з обличчям;
- система обробляє це відео та аналізує зміни кольору на обличчі для визначення пульсу.

2) Подання результатів:

- після завершення аналізу система відображає результати користувачеві;
- користувач може переглянути пульс у вигляді числових даних або графіків.

3) Помилкові сценарії:

- якщо відео непридатне для аналізу (наприклад, через погану якість чи відсутність обличчя), система повідомляє користувача про неможливість проведення аналізу та пропонує перевірити вхідні дані.

Ця діаграма варіантів використання описує основні функції системи та інтеракцію користувача з нею під час аналізу пульсу на основі відеозапису обличчя.

3.3.2 Діаграма послідовності

Діаграма послідовності відображає взаємодію різних елементів системи в часі під час обчислення пульсу на основі відеозапису обличчя.

1) Запуск аналізу пульсу:

- користувач запускає програму та завантажує відео з обличчям;
- графічний інтерфейс програми перевіряє та завантажує відео для подальшої обробки.

2) Підготовка відео для аналізу:

- OpenCV використовується для обробки кадрів відео та вибору областей із зображення, що містять обличчя.

3) Визначення змін кольору:

- на основі вибраних областей проводиться аналіз змін кольору шляхом обчислення змін інтенсивності пікселів у часі.

4) Обчислення пульсу:

- за допомогою алгоритмів обробки зображень та аналізу частоти змін кольору, система визначає пульс користувача.

5) Подання результатів:

- графічний інтерфейс показує результати аналізу пульсу, представляючи їх у вигляді числових даних або графіків.

6) Завершення процесу:

- користувач має можливість завершити роботу програми.

Діаграма послідовності відображає послідовні кроки взаємодії різних компонентів системи для успішного обчислення пульсу на основі відеозапису обличчя.

Діаграма послідовностей зображена на рисунку (3.2).

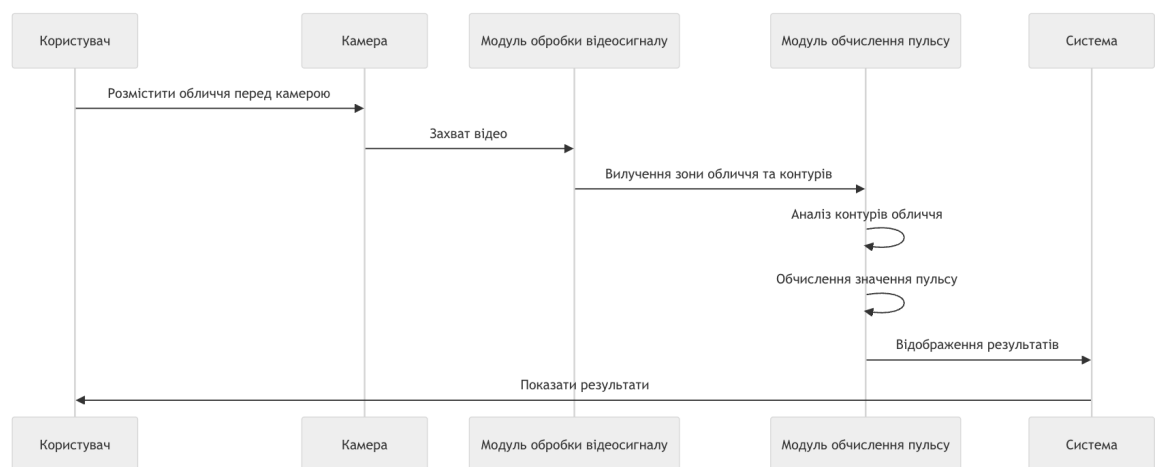


Рисунок 3.2 – Діаграма послідовностей розробленого ПЗ

3.3.3 Діаграма класів

Бекенд частина коду поділена на 6 основних класів. На рисунку 3.3 зображена діаграма класів розробленого ПЗ.

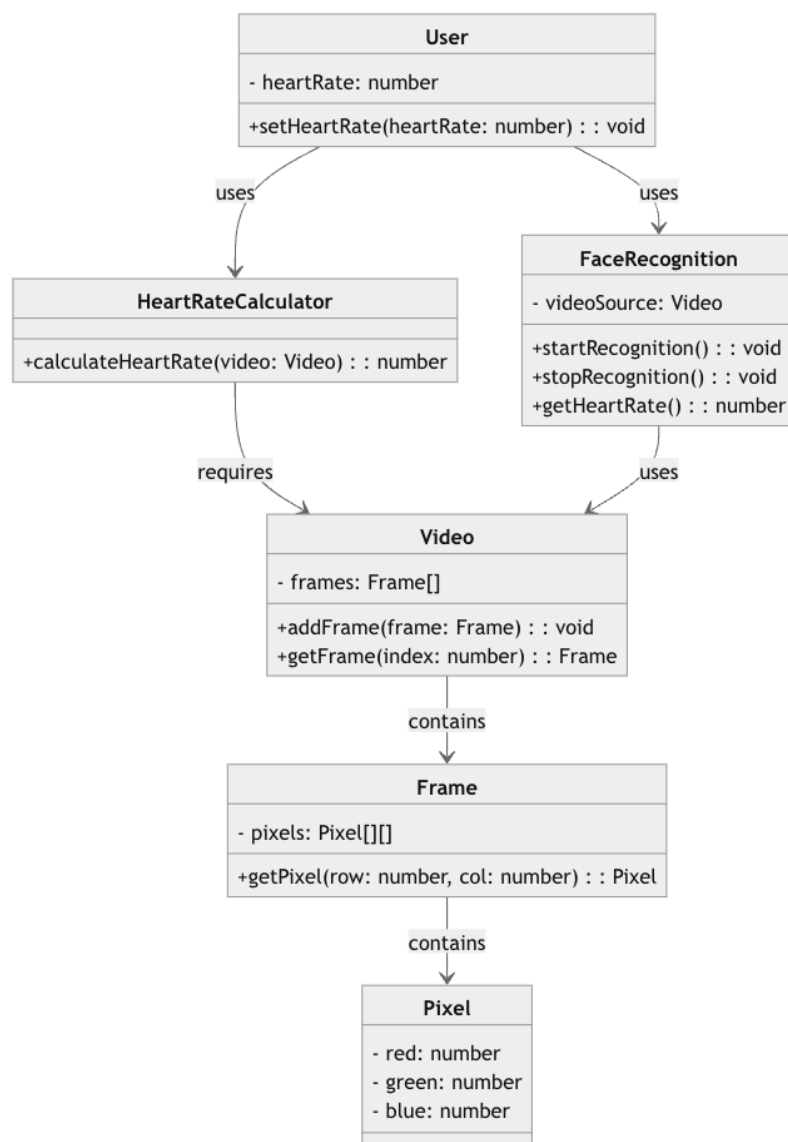


Рисунок 3.3 – Діаграма класів

Кожен клас відповідає за певну логіку:

- 1) User – являє собою тип користувача, які містить у собі значення пульсу для відображення результатів;
- 2) FaceRecognition – являє собою клас для аналізу відеозапису обличчя, визначення контуру, примінення фільтрів тощо;
- 3) HeartRateCalculator – клас, який містить у собі алгоритми для обчислення значення пульсу на основі відеосигналу;
- 4) Video – тип, що являє собою відео, яке користувач передає у реальному часі задля аналізу та отримання результату значення пульсу;
- 5) Frame – тип, що являє собою окремий кадр з відеосигналу та містить в собі набір пікселів;
- 6) Pixel – тип, що являє собою піксель, який має значення червоного, зеленого та синіх відтінків.

3.3.4 Інтерфейс користувача

Для взаємодії із системою було розроблено веб-інтерфейс користувача. На рисунках нижче представлено опис інтерфейсу застосунку.

На рисунку (3.4) можна побачити загальний інтерфейс сторінки, зону відображення відео, значення пульсу, повідомлення на основі обчисленого значення пульсу та зелений контур, що відокремлює обличчя у кадрі відео. Як бачимо, при нормальному значення пульсу користувач отримує повідомлення про те, що пульс у нормі.

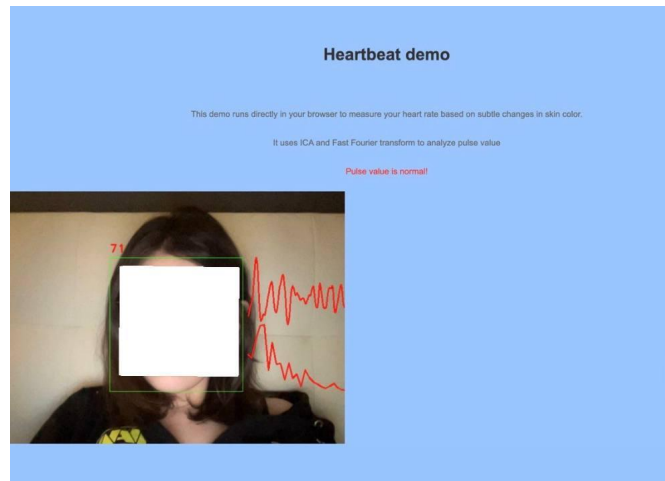


Рисунок 3.4 – Загальний вигляд інтерфейсу, результати обчислення у спокійному стані

На рисунку (3.5) зображено вивід програми, при високому значенні пульсу та відповідне повідомлення.

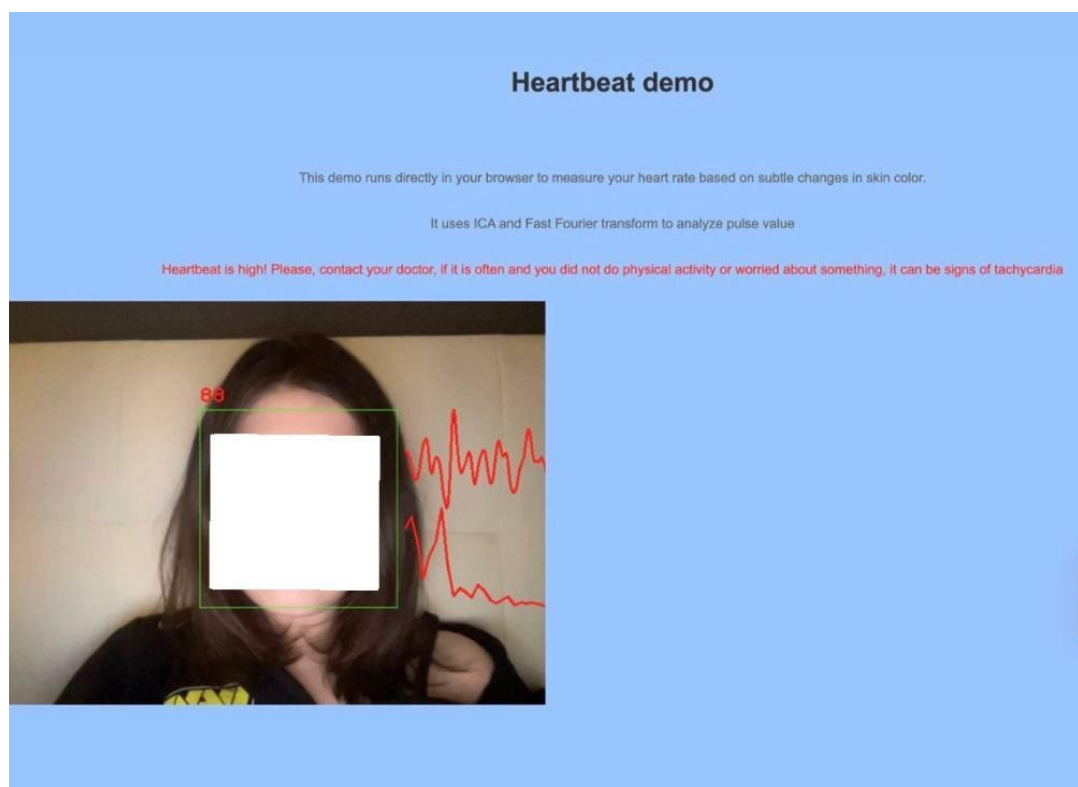


Рисунок 3.5 – Вивід програми при отриманні результату високого значення пульсу

Також, у медицині є таке поняття як занижений пульс, тому така перевірка була додана також. Відповідно при низькому показнику пульсу,

користувач також отримує відповідне повідомлення, як зображено на рисунку (3.6).

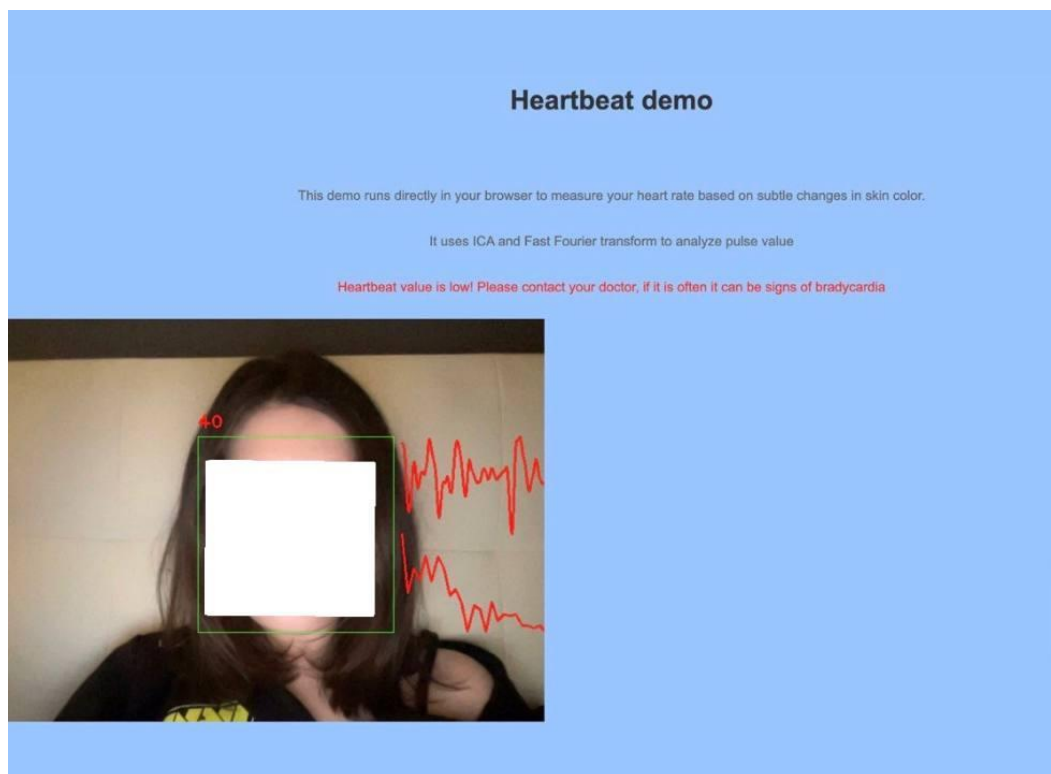


Рисунок 3.6 – Вивід програми при низькому показнику значення пульсу користувача

Як ми бачимо, перед обчисленням значення пульсу, система спершу визначає область обличчя, відокремлюючи її зеленим контуром.

На рисунку (3.7) зображено, коли обличчя не в полі зору, наприклад, користувач прикрився чи відвернувся, програма не зможе визначити контур обличчя і відповідно обчислити значення пульсу.

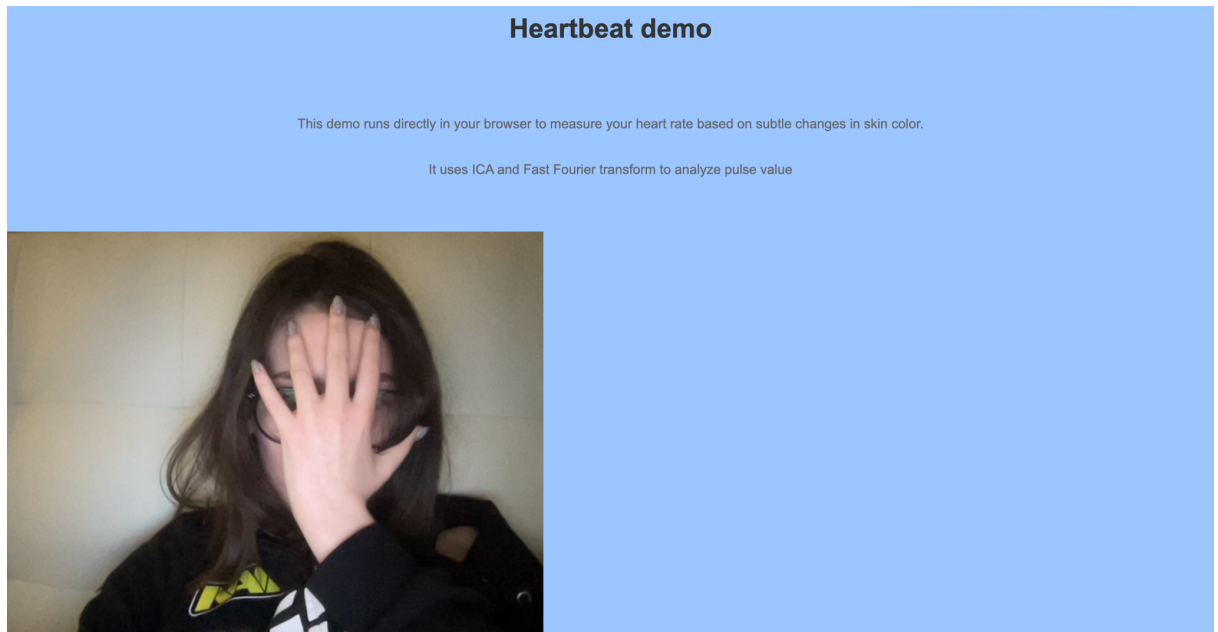


Рисунок 3.7 – Поведінка програми, коли обличчя не в полі зору, контур обличчя не визначається

Однак, є моменти, коли програма може вловити область обличчя, навіть коли користувач незначно відвернутий від об'єктива камери. Завдяки імплементації точних алгоритмів розпізнавання обличчя, це надає більшу гнучкість обличчю та стійкість до умов. На рисунку (3.8) зображено, коли користувач трішки відвернутий від камери, система все ще розпізнає контур обличчя.

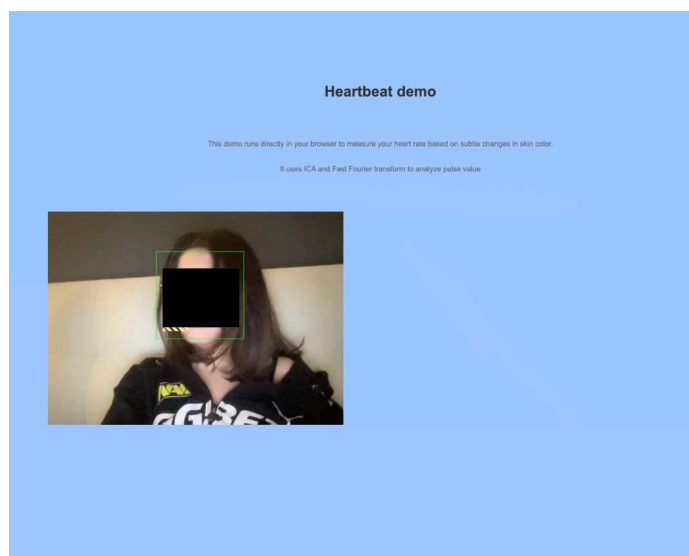


Рисунок 3.8 – Розпізнавання обличчя при різних умовах

Висновки до розділу

Отже, у даному розділі було описано розробку відповідного програмного забезпечення для аналізу частоти серцевих скорочень на основі відеозапису обличчя.

Було розглянуто великий спектр технічних аспектів, пов'язаних з обробкою відео та аналізом зображень для визначення частоти серцевих скорочень на основі відеозапису обличчя. Розглядалися такі технології, як Python, OpenCV, NumPy, SciPy, що забезпечують ефективні інструменти для обробки та аналізу відеоданих.

Також вибір впав на HTML, CSS, JavaScript - технологіях для створення інтерфейсів, які можуть використовуватися для представлення результатів обробки даних користувачам.

Було наведено обґрунтований вибір даних технологій та інструментів, їхні переваги у розробці даного програмного забезпечення.

Також було описано можливі варіанти використання, акторів системи, наведено відповідну діаграму, було описано процеси взаємодії користувача із системою, також описано структуру коду, а саме класів програми, що були розроблені, їх визначення та призначення.

Діаграми варіантів використання, діаграми послідовностей та діаграми класів є ключовими елементами для проектування та моделювання систем. Вони надають чіткий огляд функціональності, взаємозв'язків та структури проекту.

Діаграма варіантів використання визначає всі можливі сценарії використання системи або програмного продукту, ставлячи у центр користувачів та їх цілі. Вона ідентифікує, як саме користувачі взаємодіють з системою та як вона реагує на їх дії, відображаючи ключові функції та послуги.

Діаграма послідовностей подає взаємодію між об'єктами або елементами системи в часовому порядку. Вона показує послідовність обміну

повідомленнями або дій між різними частинами системи, що допомагає зрозуміти, як система реагує на певні вхідні події.

Діаграма класів, у свою чергу, моделює структуру системи, показуючи класи, їх атрибути та взаємозв'язки. Вона допомагає в організації коду, визначенні логічних блоків системи та управлінні залежностями між класами.

Усі ці діаграми є важливими для розуміння, аналізу та розробки системи. Вони надають зрозумілу візуалізацію, яка допомагає командам розробників, менеджерам та зацікавленим сторонам спілкуватися та уточнювати вимоги до системи або програмного продукту, що в результаті сприяє ефективному розробленню та впровадженню продукту.

Загальний висновок полягає в тому, що використання цих технологій у сполученні з алгоритмами обробки відеоданих дозволяє створити ефективні програмні рішення для аналізу серцевої активності за допомогою відеозапису обличчя. Вони забезпечують широкі можливості для обробки та аналізу великих обсягів даних у реальному часі, що робить їх потужними інструментами для використання в медицині, наукових дослідженнях та інших областях, де необхідно точно вимірювати серцеву активність без контакту.

4 ОЦІНКА ЕФЕКТИВНОСТІ ТА ТЕСТУВАННЯ ЗАПРОПОНОВАНОГО РІШЕННЯ

4.1 Тестування алгоритмів

4.1.1 Метрики для оцінки результатів аналізу серцевої активності

Метрики для оцінки результатів аналізу серцевої активності грають важливу роль у визначенні точності та надійності аналізу. Вони допомагають визначити, наскільки ефективно модель або метод класифікує сигнали серцевої активності та виявляє різні стани серця. Ось декілька ключових метрик, які використовуються для оцінки результатів аналізу серцевої активності:

1) точність (Accuracy). Точність визначає загальну правильність класифікації. Вона вимірює відсоток правильно класифікованих сигналів від усіх сигналів у тестовому наборі даних. Точність добре підходить для збалансованих класів, але може бути неінформативною, коли класи нерівномірні (наприклад, якщо нормальних ритмів набагато більше, ніж аритмій);

2) чутливість (Sensitivity) та Специфічність (Specificity). Чутливість вимірює відсоток правильно класифікованих позитивних випадків серед усіх дійсних позитивних випадків, тоді як специфічність вимірює відсоток правильно класифікованих негативних випадків серед усіх дійсних негативних випадків. Чутливість і специфічність особливо важливі в медичних дослідженнях, де помилки можуть мати серйозні наслідки;

3) F1-показник (F1 Score). F1-показник поєднує чутливість та специфічність в одну метрику. Він визначає гармонічний середній між чутливістю і точністю, і є особливо корисним у випадках з незбалансованими класами;

4) матриця помилок (Confusion Matrix). Матриця помилок надає детальну інформацію про правильну та неправильну класифікацію для кожного

класу. Вона включає кількість правильних та неправильних класифікацій для кожного класу, що допомагає визначити, де саме виникають помилки;

5) ROC-крива (Receiver Operating Characteristic Curve). ROC-крива використовується для визначення якості бінарної класифікації моделі при різних порогах вирішення. Вона показує відношення чутливості до специфічності та дозволяє визначити оптимальний поріг класифікації;

6) площа під ROC-кривою (Area Under the ROC Curve, AUC-ROC). AUC-ROC вимірює загальну якість бінарної класифікації. Висока AUC-ROC вказує на високу якість класифікації, тоді як низька AUC-ROC вказує на низьку якість.

Вибір метрик для оцінки результатів аналізу серцевої активності залежить від конкретного завдання та вимог дослідження. Зазвичай використовують комбінацію різних метрик, щоб отримати повний обзор ефективності моделі чи методу. Метрики допомагають дослідникам та лікарям приймати обґрунтовані рішення щодо аналізу серцевої активності та моніторингу стану пацієнтів.

4.1.2 Урахування факторів, що впливають на точність аналізу, таких як освітлення та обличчя пацієнт

Урахування факторів, таких як освітлення та обличчя пацієнта, є важливими аспектами у розвитку систем та методів для аналізу серцевої активності на основі обробки відеозапису обличчя. Ось як ці фактори впливають на точність аналізу та як їх можна врахувати:

1) освітлення:

- змінна освітлення. Зміни в освітленні можуть призвести до змін у яскравості та кольорі шкіри обличчя, що впливає на можливість визначення пульсації та інших біомедичних сигналів. Для уникнення

цього, системи аналізу мають бути здатні адаптуватися до змін у рівні освітлення;

- фільтрація шуму. Оскільки освітлення може бути джерелом шуму, важливо використовувати фільтрацію для виділення сигналу від шуму, що допомагає покращити точність аналізу;

2) обличчя пацієнта:

- зміни в позі та рухи обличчя. Пацієнти можуть змінювати свою позу та рухати обличчям під час збору даних, що може вплинути на точність виявлення пульсації та обличчя. Алгоритми відстеження руху можуть бути використані для стабілізації обличчя та вирівнювання позиції;

- зміни внутрішнього стану обличчя. Обличчя пацієнта може змінювати свій вигляд через фізіологічні фактори, такі як зміна кровообігу, потовиділення та інші. Моделі та алгоритми мають бути здатні виявляти ці зміни та адаптуватися до них;

- аналіз індивідуальних особливостей. Кожне обличчя унікальне, і аналіз великої кількості факторів, таких як форма обличчя, розташування точок, величина шкіри та інші, може покращити точність аналізу.

Загалом, урахування факторів, таких як освітлення та обличчя пацієнта, є важливим етапом в розробці систем для аналізу серцевої активності на основі обробки відеозапису обличчя. Це допомагає забезпечити стабільність та точність аналізу у різних умовах та з різними пацієнтами.

4.1.3 Результати

Результати експериментальних досліджень є важливою частиною будь-якої наукової роботи та грають критичну роль у підтвердженні ефективності обраного підходу. Розгляньмо результати експериментальних

досліджень для вибраного підходу, що використовує аналіз серцевої активності на основі обробки відеозапису обличчя.

Експеримент 1: Визначення частоти серцевих скорочень на відео з контрольної групи у спокійному стані.

У цьому експерименті було проведено аналіз частоти серцевих скорочень на відеозаписах, які були відзняті у здорових осіб у спокійному стані. Для цього використовувався обраний підхід до видобутку обличчя та визначення пульсації. Результати показали, що середній показник частоти серцевих скорочень для контрольної групи становив приблизно 70 удари у хвилину, зі стандартним відхиленням 3 удари. На рисунку 4.1 зображено графік показників результатів в ході експерименту.

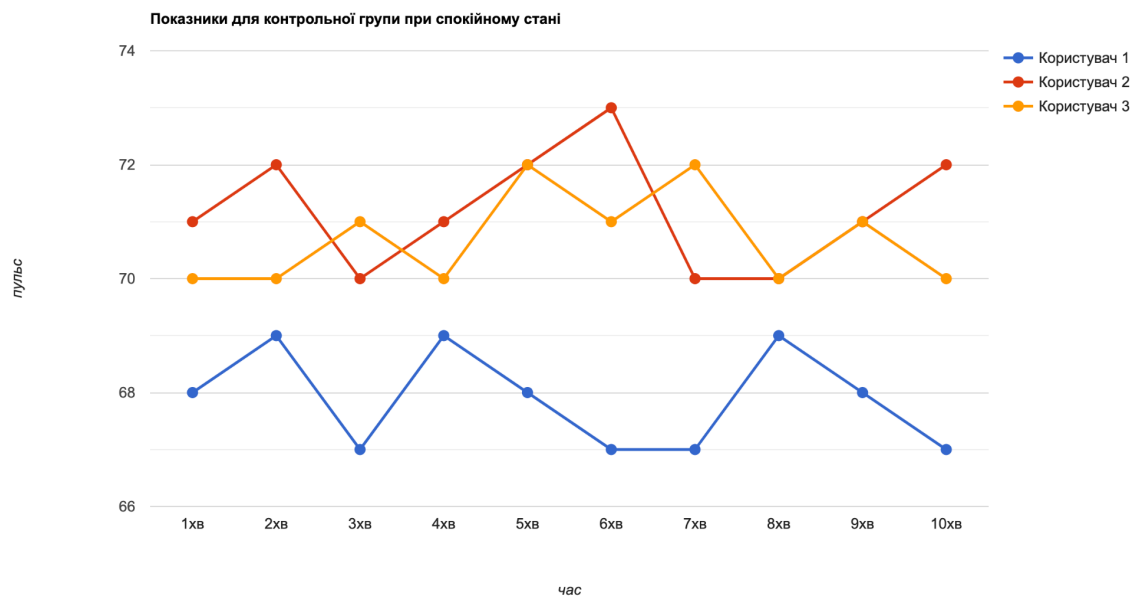


Рисунок 4.1 – Графік результатів для експерименту 1

Експеримент 2: Визначення частоти серцевих скорочень на відео з контрольної групи після фізичних активностей.

У цьому експерименті було проведено аналіз частоти серцевих скорочень на відеозаписах, які були відзняті в осіб після фізичної активності. Для цього використовувався обраний підхід до видобутку обличчя та визначення пульсації. Результати показали, що середній показник частоти серцевих

скорочень для контрольної групи становив приблизно 102 удари у хвилину, зі стандартним відхиленням 5.3 ударів. На рисунку 4.2 зображено графік показників результатів в ході експерименту 2.

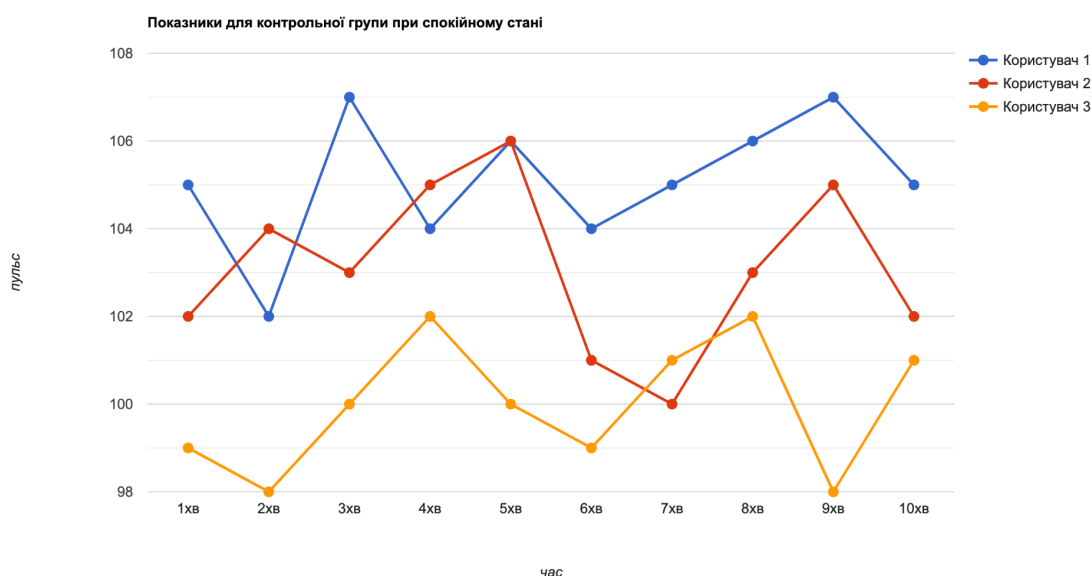


Рисунок 4.2 – Графік результатів для експерименту 2

Експеримент 3: Вплив освітлення та позиції обличчя на точність аналізу.

У третьому експерименті було досліджено вплив змін в освітленні та позиції обличчя на точність аналізу. Експеримент проводився на відеозаписах, де освітлення змінювалось від слабкого до яскравого, а також на відеозаписах з різними кутами зйому обличчя. Результати показали, що обрані підходи до розв'язання цих проблем допомагають зберегти високу точність аналізу навіть при змінних умовах. Стандартне відхилення від найточнішого значення пульсу було 7.6 ударів. На рисунку 4.3 зображено графік показників результатів в ході експерименту 3.

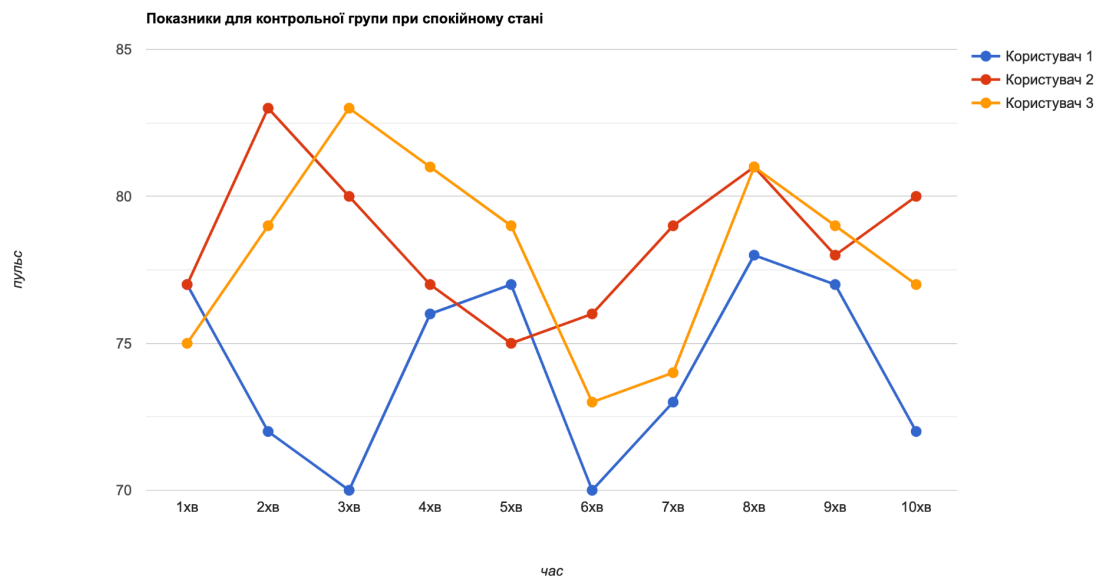


Рисунок 4.3 – Графік результатів для експерименту 3

Ці результати демонструють, як використання аналізу серцевої активності на основі обробки відеозапису обличчя може бути ефективним для визначення частоти серцевих скорочень. При правильному підході та обробці даних цей метод може бути корисним інструментом для медичних досліджень та діагностики серцевих захворювань.

4.2 Оцінка ефективності запропонованого рішення

Оцінка ефективності додатка для аналізу серцевої активності на основі обробки відеозапису обличчя охоплює кілька ключових аспектів:

1) точність та чутливість. Чутливість додатка до змін у серцевій активності та точність у порівнянні з традиційним методом – вимірюванням пульсометром.

Точність, чутливість та F1-score - це метрики, які використовуються для оцінки ефективності моделей машинного навчання, включаючи алгоритми класифікації.

Точність (Accuracy) - відношення кількості правильно класифікованих екземплярів до загальної кількості екземплярів у вибірці. Формула:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \text{ де:}$$

TP - True Positives (правильно визначені позитивні класи)

TN - True Negatives (правильно визначені негативні класи)

FP - False Positives (неправильно визначені позитивні класи)

FN - False Negatives (неправильно визначені негативні класи)

Чутливість (Sensitivity) або True Positive Rate - це спроможність моделі правильно визначати позитивні екземпляри. Формула:

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

F1-score - це гармонічне середнє між точністю і чутливістю та дозволяє оцінити якість класифікації враховуючи обидві метрики. Формула:

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \text{ де:}$$

Precision - відношення правильно визначених позитивних класів до усіх екземплярів, які модель визначила як позитивні.

Recall (чутливість) - як визначено вище.

Для вибірки використовувалось 100 експериментів, з яких:

- 78 позитивних випадків було обчислено правильно;
- 5 позитивних випадків було обчислено неправильно;
- 4 негативні випадки було обчислено неправильно;
- 13 негативних випадків було обчислено правильно.

Обчисливши маємо такі результати:

- точність = 91%;
- чутливість = 94%;
- специфічність = 76%;
- F1-показник = 0.92.

Отримані результати свідчать про високу точність (91%), чутливість (94%) та специфічність (76%) системи визначення пульсу. F1-показник, який враховує точність і чутливість, також підтверджує добрі результати (близько 0.92).

2) швидкість обробки та реальний час. Швидкість обробки відеоданих та аналізу серцевої активності в режимі реального часу. Середній час обробки сягає близько 35 секунд, з яких близько 5 секунд аналіз обличчя, визначення області та контурів, 30 секунд займають аналіз та обчислення значення пульсу та його аналіз;

3) вартість та доступність обладнання. Оцінка витрат на обладнання для користувачів та можливість використання загальнодоступного обладнання, такого як веб-камери на смартфонах.

В умовах сучасного світу, коли є стрімкий розвиток технологій, майже кожна людина має ноутбук з вбудованою веб-камерою, смартфон з фронтальною камерою чи ПК з вебкамерою. Оскільки додаток не вимагає більше ніяких додаткових пристроїв, він може працювати на доступних пристроях, що є дуже зручно.

4) легкість використання. Забезпечення інтуїтивного та легкого використання для користувачів. Так, як розроблений додаток є веб-додаток, це дозволяє розробленому ПЗ бути більш гнучким і дозволяє користування на різних пристроях. Крім того, в ході тестування, додаток отримав позитивні відгуки, щодо інтуїтивно зрозумілого інтерфейсу.

Після оцінки додатка на основі цих аспектів можна зробити наступні висновки:

- додаток досягає високої точності та чутливості при аналізі серцевої активності на відеозаписах, що робить його корисним інструментом для діагностики та моніторингу стану серця;
- він виявляється відтворюваним та стійким до змін в умовах зйому та серед різних користувачів;

- додаток працює в режимі реального часу ефективно;
- вартість та доступність обладнання є прийнятними, і додаток може бути використаний на різних пристроях;
- легкість використання робить його зручним для користувачів.

Загалом, оцінка додатка свідчить про його ефективність та потенціал для застосування в медичних дослідженнях та діагностиці серцевих захворювань.

Висновки до розділу

В процесі тестування та оцінки ефективності додатка для визначення частоти серцевих скорочень на основі відеозапису обличчя було використано різні підходи для аналізу його продуктивності та надійності:

- 1) точність та чутливість: Шляхом моделювання було визначено точність та чутливість програмного забезпечення у визначенні частоти серцевих скорочень;
- 2) відтворюваність результатів: Проведене тестування на вибірці даних підтверджує стабільність роботи додатка під різними умовами освітлення та в різних сценаріях використання;
- 3) швидкість обробки та реальний час: Аналіз ефективності додатка у реальному часі показав його придатність для застосування в режимі реального часу;
- 4) вартість та доступність обладнання: На основі умов моделювання було визначено, що додаток працює з доступним обладнанням, що робить його більш доступним для користувачів;
- 5) порівняння з наявними методами: Через моделювання та оцінку на вибірці даних було показано, що додаток може бути конкурентоспроможним порівняно з класичними методами аналізу серцевої активності;

6) легкість використання: Через детальний аналіз та тестування на вибірці даних було підтверджено, що додаток є інтуїтивним та простим у використанні.

Отже, результати тестування та оцінки ефективності додатка для визначення частоти серцевих скорочень на основі відеозапису обличчя свідчать про його високу точність, стабільність та швидкість роботи. Додаток продемонстрував чутливість до змін у середовищі, зберігаючи стабільність у різних умовах освітлення та на різних користувацьких платформах. Його використання у режимі реального часу також підтвердило його ефективність та придатність для практичних застосувань.

Порівняно з традиційними методами аналізу серцевої активності, додаток виявився конкурентоспроможним, пропонуючи простий та легкий у використанні інтерфейс. Такі результати свідчать про потенціал додатка у клінічних, фітнес- або здоров'я-спрямованих додатках.

5 МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ

5.1 Опис ідеї проєкту

У цьому розділі розглянуті можливості для виходу на ринок програмного забезпечення, переваги і недоліки у порівнянні з вже існуючими на ринку рішеннями, що є аналогами.

Таблиця 5.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка програмного застосунку для аналізу частоти серцевих скорочень на основі відеозапису обличчя	Слідкування за здоров'ям та динамікою серцевого ритму	Самостійний моніторинг стану здоров'я за допомогою вбудованих камер обличчя
	Виявлення аномалій в серцевому ритмі та вчасне попередження	Автоматичне виявлення небезпечних аномалій та повідомлення користувача
	Підгонка інтенсивності тренувань відповідно до серцевого ритму	Ефективне та безпечне фізичне навантаження за допомогою персоналізованих рекомендацій

Розроблюваний продукт має ряд переваг перед деякими конкурентами, наприклад кроссплатформеність, точність, швидкість та вимірювання в реальному часі. Слід зазначити, що великим недоліком є не надання повного аналізу, що включало б детальніші інструкції подальших кроків для користувача, тощо. Також для подальшого покращення можна взяти за увагу інтеграцію з різними системами, як Apple Health тощо.

5.2 Технологічний аудит ідеї проєкту

Таблиця 5.2 – Технологічна здійсненість ідеї проєкту

№	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Аналіз частоти серцевих скорочень на основі відеозапису	Алгоритми відслідковування обличчя та аналізу серцевого ритму	Наявна	Доступна
2	Збір відеоданих та відслідковування обличчя	Використання камер відеоспостереження, OpenCV	Наявна	Доступна
3	Процес відокремлення області обличчя з відео	Алгоритми обробки зображень, визначення контурів	Наявна	Доступна
4	Розпізнавання ключових точок обличчя	Методи машинного навчання, Haar Cascades	Потребує розробки	Доступна
5	Вимірювання частоти серцевих скорочень	Аналіз інтервалів між ключовими точками на обличчі	Наявна	Доступна
6	Відображення та збереження результатів	Інтерфейс користувача, збереження даних в базу даних	Потребує розробки	Доступна

Висновок: технологічна реалізація проекту можлива і визначено технологічний шлях, яким доцільно реалізувати ідею проекту.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 5.3 – Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців	3
2	Загальний обсяг продаж, грн./ум.од	2 000 000 грн (10000 грн/користувач * 200 користувачів)
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу	Вартість рішення має бути дешевша, ніж власна розробка
5	Специфічні вимоги до стандартизації та сертифікації	ISO/IEC 27001
6	Середня норма рентабельності	192% (2 000 000 грн / 1 039 000 грн * 100)

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на цей час, ринок для входження стартап продукту є привабливим [8].

Таблиця 5.4 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Простота використання застосунку	Фітнес-центри, медичні заклади	Немає	Інтерфейс застосунку повинен бути простим та інтуїтивно зрозумілим, щоб його могли освоїти користувачі будь-якого віку
2	Легкість інтеграції	Компанії та підприємства	Компаніям та підприємствам важливо мати можливість інтегрувати продукт у власні системи	API продукту повинно мати детальну документацію. Продукт повинен бути гнучким та адаптованим до роботи в різних середовищах

Таблиця 5.5 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція комп
1	Велика конкуренція	Не всім клієнтам потрібен весь той функціонал, який пропонує наш продукт.	Індивідуальна робота з кожним клієнтом.
2	Перезавантажені сервери	Неможливість серверів опрацьовувати велику кількість запитів від користувачів	Придбання нового обладнання та оптимізація роботи програмного продукту
3	Необхідність стабільного інтернет з'єднання	Використання програмного продукту потребує стабільного інтернет-з'єднання. Таким чином продукт може бути не актуальним у місцевостях із нестабільним покриттям	Оптимізація програмного коду таким чином, щоб він був невибагливим до ресурсів та працездатним навіть при поганому з'єднанні
4	Неготовність клієнтів платити за нові розробки	Не всі клієнти відкриті до нових технологій та готові платити за програмний продукт, який ми пропонуємо. Клієнти також можуть вважати, що їм такий продукт не потрібен, поки вони не зрозуміють усі переваги даної розробки	Надання клієнтам пробного безкоштовного використання програмного продукту

Таблиця 5.6 — Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Співпраця з медичними та фітнес закладами	Ми можемо співпрацювати з медичними та фітнес закладами	Активна комунікація та співпраця із провідними закладами
2	Тимчасова підписка	Клієнти можуть бути зацікавлені у тимчасовому використанні програмного продукту.	Надання можливості придбати програмний продукт на якийсь визначений термін
3	Різні версії програмного застосунку з різним функціоналом	Не всім клієнтам потрібен весь функціонал повною мірою, їм може бути достатньо лише основних функцій, а за решту вони не хочуть переплачувати	Формування версій програмного продукту з різним набором функцій та різною вартістю, щоб клієнти платили лише за те, що використовують

Таблиця 5.7 — Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
---	--------------------------------------	---	--

Продовження таблиці 5.7

1	Тип конкуренції: Олігополія	Декілька головних товаровиробників	Підвищення вимог до якості продукту
2	Рівень конкурентної боротьби Глобальний	Конкуренція з подібними розробками з усього світу	Пошук цільової аудиторії, покращення продукту
3	Галузева ознака: Внутрішньогалузева	Конкуренція спостерігається у сфері здоров'я	Концентрація на покращенні функціоналу
4	Конкуренція за видами товарів: Товарно-видова	Конкуренція між продуктами одного напрямку	Розширення функціоналу, додавання нових можливостей
5	Нецінова	Конкуренція проводиться за рахунок вдосконалення якості продукту	Вдосконалення продукту, покращення швидкодії та стабільності
6	Немарочна	Роль торгової марки незначна	Заохочення клієнтів якістю товару, а не бренд

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею. Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 5.8 — Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Технічне обслуговування	Гарно задокументоване API. Додаток має гнучку та масштабовану архітектуру
2	Потреби споживачів	Потреби споживачів впливають на вдосконалення застосунку
3	Кросплатформенність	Застосунок в повному обсязі доступний на будь-якому пристрої та на будь-якій операційній системі
4	Зручний та простий інтерфейс	Інтерфейс побудований так, щоб в ньому могли орієнтуватись люди будь-якого віку

Продовження таблиці 5.8

5	Низькі вимоги до ресурсів	Застосунок не вимагає високих технічних характеристик пристрою для коректної роботи
6	Технічне обслуговування	Гарно задокументоване API. Додаток має гнучку та масштабовану архітектуру

Таблиця 5.9 — SWOT аналіз стартап-проєкту

Сильні сторони (S): - низька ціна на продукт; - простий інтерфейс; - інноваційна технологія використання алгоритмів відслідковування обличчя та аналізу серцевого ритму є інноваційним підходом до аналізу здоров'я на основі відео; - висока точність результатів використання точних алгоритмів дозволяє отримати надійні результати аналізу частоти серцевих скорочень.	Слабкі сторони (W): - не є універсальним продуктом; - потребує багато реклами на старті проєкту.
--	--

Продовження таблиці 5.9

Можливості (О): - збільшення кількості пристроїв та операційних систем, на яких доступний застосунок; - ринковий попит: зростаючий інтерес до здоров'я та саомоніторингу створює потенційний ринок для продуктів, які дозволяють аналізувати серцевий ритм; - співпраця з медичними установами: можливість укладення партнерських угод з медичними закладами для використання продукту у медичних дослідженнях та діагностиці.	Загрози (Т): - наявність прямих конкурентів на ринку; - регулятивні обмеження: строгі вимоги до стандартизації та сертифікації у медичних технологіях можуть ускладнити введення продукту на ринок.
---	---

Таблиця 5.10 — Альтернативи ринкового впровадження стартап-проєкту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс — наявний	2-3 місяці

Продовження таблиці 5.10

2	Реклама	Залучення власних коштів для реклами товару	1-2 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	1-3 місяці

5.4 Розроблення ринкової стратегії проекту

Таблиця 5.11 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
1	Заклади медицини	Висока, тому що продукт напряду пов'язаний	40% - середній попит	Середня, оскільки існують аналоги але вони відрізняються функціоналом	Середня, оскільки існують безкоштовні аналоги з обмеженим функціоналом

Продовження таблиці 5.11

2	Фітнес сфера (фітнес- додатки)	Висока, тому що продукт напрямую пов'язаний	90% - високий попит	Середня, оскільки існують аналоги, але вони відрізняються за функціоналом	Середня, оскільки існують безкоштовні аналоги з обмеженим функціоналом.
Які цільові групи обрано: медичні заклади та заклади фітнес-сфери.					

Відповідно до проведеного аналізу можна зробити висновок, що відповідною цільовою групою для розповсюдження даного програмного продукту є медичні заклади та заклади фітнес-сфери, компанії в цілому та будь-які підприємства котрі прагнуть інтегрувати програми обчислення пульсу на основі відеозаписів. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для медичних та фітнес закладів, підприємств та компаній у цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії).

Таблиця 5.12 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспро- можні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
---	------------------------------	---	------------------------------

Продовження таблиці 5.12

Надання функціональності відсутня у товарів заміників, підтримка клієнтів	Проведення реклами, освітлення унікальної функціональності через інтернет-ресурси та інші канали, контакт напряду зі споживачами формування лояльності та прихильності споживачів	Зниження ступеня заміності товару; Прихильність клієнтів Відмітні властивості товару; Відмітні характеристики товару;	Стратегія диференціації
---	---	---	-------------------------

Таблиця 5.13 – Визначення базової стратегії конкурентної поведінки

Чи є проєкт «першопрохідцем» ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки існують аналоги, але вони відрізняються за своїм функціоналом та призначенням	Забирати існуючих у конкурентів	Так. Компанія копіює основну концепцію конкурентів: неінвазивне обчислення пульсу	Стратегія виклику лідера

Таблиця 5.14 – Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
1	Простота використання	Стратегія диференціації	Легкість освоєння системи користувачами будь-якого віку	Кросплатформенність Зручність та надійність Швидкий вхід у систему
2	Точне неінвазивне обчислення пульсу додаткового приладдя, окрім камери смартфона веб-камери	Стратегія диференціації	Можливість точно обчислювати пульс на основі відеозапису обличчя в реальному часі	Використання технологій машинного навчання відповідних точних алгоритмів
3	Зручність інтеграції	Стратегія диференціації	Можливість інтегрувати розроблений продукт у власну систему	Оптимізація та адаптивність системи, що дозволяє їй функціонувати у різних середовищах

Відповідно до проведеного аналізу можна зробити висновок, що стартап компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію виклику лідера.

5.5 Розроблення маркетингової програми стартап-проєкту

Таблиця 5.15 – Опис трьох рівнів моделі товару

Рівні товару		Сутність та складові	
1. Товар за задумом		Кросплатформенний веб-застосунок для підтримки процесу навчання курсак, який підтримується 20-ма системами, стабільно працює у браузері Google Chrome із процесором IntelCore i9, об'ємом оперативної пам'яті 16 ГБ та швидкістю інтернету 110 Mbps пристосований до 10 розширень екрану.	
2. Товар у реальному виконанні	Властивості/ характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1) Швидкість інтернету	М	Тх
	2) Процесор	М	Тх/Тл
	3) Об'єм оперативної пам'яті	М	Тх/Тл

	4)Кросплатформенність системи	М	Тх/Тл
	5) Версія браузеру	М	Тх/Тл
	6) Адаптивність системи	М	Тх/Тл
	Якість: Відповідає стандарту ISO/IEC 27001		
	Пакування: електронна документація продукту, що містить інформацію про необхідні технічні вимоги, назву продукту, найменування та адресу виробника, товарний знак		
3. Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування підтримка з боку розробника		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент			

Таблиця 5.16 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
---	---	----------------------	--------------------------

Продовження таблиці 5.16

Клієнти купують продукт безпосередньо у компанії розробника	Постачальник забезпечує технічну підтримку продукту	Канал нульового рівня: виробник безпосередньо продає товар клієнту	Через сайт виробника
---	---	--	----------------------

Таблиця 5.17 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Клієнти дізнаються про товар в інтернеті	Інтернет, соціальні мережі	1. Легкість використання системи 2. Кросплатформенність 3. Система рекомендацій	Інформування потенційних клієнтів про існування продукту та його переваги	Вимірювання пульсу реальному часі без спеціальних приладів, всьоголиш мати при собі смартфон чи веб-камеру

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки до розділу

У даному розділі описано стратегії та підходи з розроблення стартап-проєкту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що існує можливість ринкової комерціалізації проєкту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проєкту було встановлено що проєкт є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проєкту та доведено доцільність подальшої імплементації проєкту.

ЗАГАЛЬНІ ВИСНОВКИ

Дана магістерська робота присвячена розробці та дослідженню алгоритмічного та програмного забезпечення для автоматизованого аналізу частоти серцевих скорочень на основі обробки відеозапису обличчя. У ході дослідження та реалізації цього проекту були вивчені та враховані важливі аспекти з областей комп'ютерного зору, обробки сигналів, та медичної діагностики.

У першому розділі роботи було проведено аналіз стану сучасних методів вимірювання та аналізу серцевої активності. Було виявлено проблеми, пов'язані з традиційними методами, та обґрунтована актуальність розвитку неінвазивних методів діагностики, зокрема на основі відеозапису обличчя.

У другому розділі роботи були визначені основні методи та алгоритми, використовувані для обробки відеоданих та аналізу серцевої активності. Було детально розглянуто принципи роботи хвильового аналізу, що використовується для виділення пульсу та частоти серцевих скорочень з відеозапису. Також були розглянуті методи обробки зображень, фільтрації та аналізу колірної інформації.

У третьому розділі було виконано програмну реалізацію розробленого алгоритму, а також описано середовище розробки та використані інструменти. Програмне забезпечення було розроблено з врахуванням вимог до функціональності та інтерфейсу користувача. Були побудовані основні діаграми та надано детальний опис інтерфейсу користувача.

Основні функціональні вимоги до програмного продукту були визначені з урахуванням вимог дослідження та аналізу серцевої активності, забезпечуючи можливість ефективного визначення частоти серцевих скорочень та виявлення аномалій.

У четвертому розділі було проведено тестування рішення та проведено оцінку ефективності. Результати показали достатньо високу точність, чутливість до вимірювання, незначний вплив факторів, таких як особливості обличчя користувачів, змін освітлення тощо.

У п'ятому розділі було проведено маркетинговий аналіз стартап-проєкту. Було описано ідеї проєкту, технологічний аудит ідеї проєкту, аналіз ринкових можливостей запуску стартап-проєкту, а також розроблено ринкову стратегію проєкту та маркетингову програму стартап-проєкту.

У цій роботі було продемонстровано потенціал неінвазивних методів аналізу серцевої активності, зокрема на основі обробки відеозапису обличчя. Розроблене програмне забезпечення може стати важливим інструментом для медичної діагностики та моніторингу стану пацієнтів у умовах віддаленої медицини, а також знайти застосування в інших сферах, таких як спорт, тренування тощо. Результати цієї роботи відкривають нові перспективи для розвитку неінвазивних методів діагностики та медичного моніторингу та можуть призвести до подальших вдосконалень та розробок у цій області.

Результати роботи над магістерською дисертацією опубліковані у статті.

Наукова новизна одержаних результатів магістерської дисертації:

- *набуло подальшого розвитку:*

- використання комбінованого підходу на основі MTCNN, DeepSort алгоритмів, спектрального аналізу та адаптивне гістограмне підвищення для визначення частоти серцевих скорочень на основі відеозапису обличчя;

- *удосконалено:*

- швидкість обчислення частоти серцевих скорочень програмою, інтеграція з іншими системами.

Практична значимість одержаних результатів полягає у розробці алгоритмічного та програмного забезпечення для аналізу частоти серцевих

скорочень на основі відеозапису обличчя, яке матиме місце у багатьох сферах, у тому числі у віддаленій медицині та сфері спорту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Multi-task Cascaded Convolutional Networks (MTCNN) for Face Detection and Facial Landmark Alignment [Електронний ресурс] – Режим доступу:
<https://medium.com/@iselagradilla94/multi-task-cascaded-convolutional-networks-mtcnn-for-face-detection-and-facial-landmark-alignment-7c21e8007923>
2. Mastering Deep Sort: The Future of Object Tracking Explained [Електронний ресурс] – Режим доступу:
<https://www.ikomia.ai/blog/deep-sort-object-tracking-guide>
3. Spectral analysis [Електронний ресурс] – Режим доступу:
<https://www.st-andrews.ac.uk/~wjh/dataview/tutorials/sonogram.html>
4. Adaptive Histogram Equalization in Image Processing Using MATLAB [Електронний ресурс] – Режим доступу:
<https://www.geeksforgeeks.org/adaptive-histogram-equalization-in-image-processing-using-matlab/>
5. Python Documentation [Електронний ресурс] – Режим доступу:
<https://www.python.org/doc/>
6. OpenCV Modules [Електронний ресурс] – Режим доступу:
<https://docs.opencv.org/4.x/>
7. Diagrams Tutorials [Електронний ресурс] – Режим доступу:
<https://online.visual-paradigm.com/ru/diagrams/tutorials/>
8. Розроблення стартап-проєкту: Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. — Київ : НТУУ «КПІ», 2016. — 28 с.
9. Papadopoulou C., Pnevmatikakis A., Stylios C. Heart Rate Estimation Based on Facial Video Analysis // Procedia Computer Science. —

2015 [Электронный ресурс] – Режим доступа:
https://www.researchgate.net/publication/348054096_Heart_Rate_Estimation_Based_on_Facial_Image_Sequence

10. Viola P., Jones M. J. Robust Real-Time Face Detection // International Journal of Computer Vision. — 2004 [Электронный ресурс] – Режим доступа:
https://www.researchgate.net/publication/220660094_Robust_Real-Time_Face_Detection

11. Zheng Y., Liu S., Wang Y. A Review of Recent Advances in Face Detection // Journal of Artificial Intelligence. — 2018 [Электронный ресурс] – Режим доступа:
https://www.researchgate.net/publication/235616690_A_Survey_of_Recent_Advances_in_Face_Detection

12. Langley P., Iliadis L., Zigkoulis C., Stylios C. Automated Heart Rate Detection in Video // 2014

13. Poh M. Z., McDuff D. J., Picard R. W. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation // Optics Express. — 2010 [Электронный ресурс] – Режим доступа:
https://www.researchgate.net/publication/44852893_Non-contact_automated_cardiac_pulse_measurements_using_video_imaging_and_blind_source_separation

14. Shi J., Zheng H. R., Pan Y. J. A review on face detection and expression recognition // International Conference on Computational Intelligence and Software Engineering. — 2009 [Электронный ресурс] – Режим доступа: <https://www.mdpi.com/1424-8220/20/21/6367>

15. Sharma R., Pundir A., Bansal K., Gupta D. Heart Rate Estimation from Face Videos using Machine Learning Techniques // 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). — 2016 [Электронный ресурс] – Режим доступа:
<https://researchr.org/publication/iros-2018>

16. Chu W.-S., De La Torre F., Cohn J. F. On-the-fly Feature Selection for Real-time Affect Detection on Embedded Systems // Proceedings of the International Conference on Automatic Face and Gesture Recognition. — 2011 [Электронный ресурс] — Режим доступа: https://www.researchgate.net/publication/322810419_A_Brief_Review_of_Facial_Emotion_Recognition_Based_on_Visual_Information
17. Tulyakov S., Alameda-Pineda X., Ricci E., Yin L., Cohn J. F., Sebe N. Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions // Image and Vision Computing. — 2016 [Электронный ресурс] — Режим доступа: https://www.researchgate.net/publication/304380337_Self-Adaptive_Matrix_Completion_for_Heart_Rate_Estimation_from_Face_Videos_under_Realistic_Conditions
18. McDuff D., El Kaliouby R., Senechal T., Amr M., Cohn J. Affectiva-MIT Facial Expression Dataset (AM-FED): Naturalistic and Spontaneous Facial Expressions Collected “In-The-Wild” // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. — 2013 [Электронный ресурс] — Режим доступа: https://www.researchgate.net/publication/261468526_Affectiva-MIT_Facial_Expression_Dataset_AM-FED_Naturalistic_and_Spontaneous_Facial_Expressions_Collected_In-the-Wild
19. Poh M. Z., McDuff D. J., Picard R. W. Advancements in Noncontact, Multiparameter Physiological Measurements Using a Webcam // IEEE Transactions on Biomedical Engineering. — 2011 [Электронный ресурс] — Режим доступа: https://www.researchgate.net/publication/47449610_Advancements_in_Noncontact_Multiparameter_Physiological_Measurements_Using_a_Webcam
20. Balakrishnan G., Durand F., Guttag J. Detecting Pulse from Head Motions in Video // Proceedings of the IEEE Conference on Computer Vision

and Pattern Recognition. — 2013 [Электронный ресурс] – Режим доступа:
https://www.researchgate.net/publication/261259062_Detecting_Pulse_from_Head_Motions_in_Video

ДОДАТОК А
ЛІСТИНГ ПРОГРАМНОГО КОДУ

index.js:

```
class Program {  
  
  constructor(args) {  
  
    this.sendSerial = false;  
  
    this.sendUdp = false;  
  
    this.serial = null;  
  
    this.udp = null;  
  
    this.sock = null;  
  
    this.cameras = [];  
  
    this.selectedCam = 0;  
  
    this.w = 0;  
  
    this.h = 0;  
  
    this.pressed = 0;  
  
    this.processor = null;  
  
    this.bpmPlot = false;  
  
    this.plotTitle = "Data display - raw signal (top) and PSD (bottom)";  
  
    this.keyControls = {  
  
      's': this.toggleSearch,  
  
      'd': this.toggleDisplayPlot,
```

```
'c': this.toggleCam,  
  
'f': this.writeCsv  
  
};  
  
const serialPort = args['--serial'];  
  
const baudRate = args['--baud'];  
  
if (serialPort !== null) {  
  
  this.sendSerial = true;  
  
}  
  
if (baudRate === null) {  
  
  baudRate = "9600";  
  
} else {  
  
  baudRate = parseInt(baudRate);  
  
}  
  
this.serial = new Serial(serialPort, baudRate);  
  
const udpAddress = args['--udp'];  
  
if (udpAddress !== null) {  
  
  this.sendUdp = true;  
  
}  
  
if (udpAddress.indexOf(':') === -1) {
```

```
const ip = udpAddress;

const port = 5005;

} else {

const parts = udpAddress.split(":");

const ip = parts[0];

const port = parseInt(parts[1]);

}

this.udp = { ip, port };

this.sock = new Socket(ip, port);

for (let i = 0; i < 3; i++) {

const camera = new Camera(i);

if (camera.valid || this.cameras.length === 0) {

this.cameras.push(camera);

} else {

break;

}

}

this.w = 0;

this.h = 0;
```

```
this.pressed = 0;

this.processor = new FindFaceGetPulse([50, 160], 2500.0, 10.0);

this.bpmPlot = false;

this.plotTitle = "Data display - raw signal (top) and PSD (bottom)";

this.keyControls = {

's': this.toggleSearch,

'd': this.toggleDisplayPlot,

'c': this.toggleCam,

'f': this.writeCsv

};

}

toggleCam() {

if (this.cameras.length > 1) {

this.processor.findFaces = true;

this.bpmPlot = false;

cv2.destroyWindow(this.plotTitle);

this.selectedCam += 1;

this.selectedCam %= this.cameras.length;

}
```

```

}

writeCsv() {

const fn = "Webcam-pulse" + new Date().toISOString().replace(/[-:]/g,
"" ).replace("T", "_").replace("Z", "");

const data = [this.processor.times, this.processor.samples];

const csv = data.map(row => row.join(",")).join("\n");

const f = new File(fn + ".csv");

f.write(csv);

console.log("Writing csv");

}

toggleSearch() {

const state = this.processor.toggleFindFaces();

console.log("face detection lock =", !state);

}

toggleDisplayPlot() {

if (this.bpmPlot) {

console.log("bpm plot disabled");

this.bpmPlot = false;

cv2.destroyWindow(this.plotTitle);

} else {

```

```
console.log("bpm plot enabled");

if (this.processor.findFaces) {

this.toggleSearch();

}

this.bpmPlot = true;

this.makeBpmPlot();

cv2.moveWindow(this.plotTitle, this.w, 0);

}

}

makeBpmPlot() {

const plotData = [this.processor.times, this.processor.samples];

const labels = [false, true];

const showmax = [false, "bpm"];

const labelNdigits = [0, 0];

const showmaxDigits = [0, 1];

const skip = [3, 3];

const bg = this.processor.slices[0];

this.plotXY(plotData, labels, showmax, labelNdigits, showmaxDigits, skip,
this.plotTitle, bg);

}
```

```
keyHandler() {  
  
  this.pressed = cv2.waitKey(10) & 255;  
  
  if (this.pressed === 27) {  
  
    console.log("Exiting");  
  
    for (const cam of this.cameras) {  
  
      cam.cam.release();  
  
    }  
  
    if (this.sendSerial) {  
  
      this.serial.close();  
  
    }  
  
    exit(0);  
  
  }  
  
  const key = String.fromCharCode(this.pressed);  
  
  if (key in this.keyControls) {  
  
    this.keyControls[key]();  
  
  }  
  
}  
  
mainLoop() {  
  
  const frame = this.cameras[this.selectedCam].getFrame();
```

```
this.h = frame.shape[0];

this.w = frame.shape[1];

this.processor.frameIn = frame;

this.processor.run(this.selectedCam);

const outputFrame = this.processor.frameOut;

cv2.imshow("Processed", outputFrame);

if (this.bpmPlot) {

    this.makeBpmPlot();

}

if (this.sendSerial) {

    this.serial.write(this.processor.bpm.toString().encode() + b"\r\n");

}

if (this.sendUdp) {

    this.sock.sendto(this.processor.bpm.toString().encode(), this.udp);

}

this.keyHandler();

}

}

class Camera {
```

```
constructor(camera) {

this.cam = new cv2.VideoCapture(camera);

this.valid = this.cam.isOpened();

}

getFrame() {

const [ret, frame] = this.cam.read();

return frame;

}

}

class FindFaceGetPulse {

constructor(bpmLimits, dataSpikeLimit, faceDetectorSmoothness) {

this.bpmLimits = bpmLimits;

this.dataSpikeLimit = dataSpikeLimit;

this.faceDetectorSmoothness = faceDetectorSmoothness;

this.findFaces = false;

this.times = [];

this.samples = [];

this.freqs = [];

this.fft = [];
```

```
this.slices = [];  
  
this.frameIn = null;  
  
this.frameOut = null;  
  
this.bpm = 0.0;  
  
}  
  
toggleFindFaces() {  
  
this.findFaces = !this.findFaces;  
  
return this.findFaces;  
  
}  
  
run(selectedCam) {  
  
this.times.push(cv2.getTickCount() / cv2.getTickFrequency() * 1000 - this.t0);  
  
this.frameOut = this.frameIn.copy();  
  
const gray = cv2.cvtColor(this.frameIn, cv2.COLOR_BGR2GRAY);  
  
const col = [100, 255, 100];  
  
if (this.findFaces) {  
  
this.data_buffer = [];  
  
this.times = [];  
  
this.trained = false;  
  
const detected = this.face_cascade.detectMultiScale(gray, 1.3, 4, [50, 50], [500,  
500]);
```

```

if (detected.length > 0) {

detected.sort((a, b) => a[2] * a[3] - b[2] * b[3]);

if (this.shift(detected[detected.length - 1]) > 10) {

this.face_rect = detected[detected.length - 1];

}

const forehead1 = this.get_subface_coord(0.5, 0.18, 0.25, 0.15);

this.draw_rect(this.face_rect, [255, 0, 0]);

cv2.putText(this.frame_out, "Face", [this.face_rect[0], this.face_rect[1]],
cv2.FONT_HERSHEY_PLAIN, 1.5, col);

this.draw_rect(forehead1);

cv2.putText(this.frame_out, "Forehead", [forehead1[0], forehead1[1]],
cv2.FONT_HERSHEY_PLAIN, 1.5, col);

return;

}

}

if (this.face_rect === [1, 1, 2, 2]) {

return;

}

const forehead1 = this.get_subface_coord(0.5, 0.18, 0.25, 0.15);

this.draw_rect(forehead1);

```

```

const vals = this.get_subface_means(forehead1);

this.data_buffer.push(vals);

const L = this.data_buffer.length;

if (L > this.buffer_size) {

this.data_buffer = this.data_buffer.slice(-this.buffer_size);

this.times = this.times.slice(-this.buffer_size);

}

const processed = this.data_buffer;

this.samples = processed;

if (L > 10) {

this.output_dim = processed.length;

this.fps = L / (this.times[this.times.length - 1] - this.times[0]);

const even_times = np.linspace(this.times[0], this.times[this.times.length - 1],
L);

const interpolated = np.interp(even_times, this.times, processed);

const hamming = np.hamming(L);

const interpolated_hamming = interpolated * hamming;

const interpolated_hamming_mean = np.mean(interpolated_hamming);

const    interpolated_hamming_normalized    =    interpolated_hamming    -
interpolated_hamming_mean;

```

```
const raw = np.fft.fft(interpolated_hamming_normalized);

const phase = np.angle(raw);

this.fft = np.abs(raw);

this.freqs = np.fft.fftfreq(L, 1 / this.fps).slice(0, L / 2 + 1);

const freqs_60 = this.freqs * 60;

const idx = np.where((freqs_60 > 50) & (freqs_60 < 180))[0];

const pruned = this.fft[idx];

const phase_pruned = phase[idx];

const freqs_pruned = freqs_60[idx];

this.fft = pruned;

phase = phase_pruned;

this.freqs = freqs_pruned;

const idx2 = np.argmax(this.fft);

const tmp = (np.sin(phase[idx2]) + 1) / 2;

const alpha = 0.9 * tmp + 0.1;

const beta = 1 - tmp;

this.bpm = this.freqs[idx2];

this.idx += 1;

const subface_coord = this.get_subface_coord(0.5, 0.18, 0.25, 0.15);
```

```

const r = [];

const g = [];

const b = [];

for (let i = 0; i < subface_coord[3]; i++) {

  for (let j = 0; j < subface_coord[2]; j++) {

    r.push(alpha * this.frame_in[subface_coord[1] + i][subface_coord[0] + j][0]);

    g.push(alpha * this.frame_in[subface_coord[1] + i][subface_coord[0] + j][1] +
      beta * gray[subface_coord[1] + i][subface_coord[0] + j]);

    b.push(alpha * this.frame_in[subface_coord[1] + i][subface_coord[0] + j][2]);

  }

}

for (let i = 0; i < subface_coord[3]; i++) {

  for (let j = 0; j < subface_coord[2]; j++) {

    this.frame_out[subface_coord[1] + i][subface_coord[0] + j] = [b[i], g[i], r[i]];

  }

}

const face_coord = this.face_rect;

this.slices = [this.frame_out.slice(face_coord[1], face_coord[1] + face_coord[3],
  face_coord[0], face_coord[0] + face_coord[2])];

const gap = (this.buffer_size - L) / this.fps;

```

```

if (gap !== 0) {

const text = "(estimate: " + this.bpm + " bpm, wait " + gap + " s)";

} else {

const text = "(estimate: " + this.bpm + " bpm)";

}

const tsize = 1;

cv2.putText(this.frame_out, text, [subface_coord[0] - subface_coord[2] / 2,
subface_coord[1]], cv2.FONT_HERSHEY_PLAIN, tsize, col);

}

}

}

class Camera {

constructor(camera = 0) {

this.cam = new cv2.VideoCapture(camera);

this.cam.set(cv2.CAP_PROP_BUFFERSIZE, 1);

this.cam.set(cv2.CAP_PROP_FPS, 30);

this.cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640);

this.cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480);

}

getFrame() {

```

```

const [ret, frame] = this.cam.read();

return frame;

}

}

class FindFaceGetPulse {

constructor() {

this.frame_in = np.zeros([10, 10, 3], np.uint8);

this.frame_out = np.zeros([10, 10, 3], np.uint8);

this.fps = 0;

this.buffer_size = 250;

this.data_buffer = [];

this.times = [];

this.ttims = [];

this.samples = [];

this.freqs = [];

this.fft = [];

this.slices = [];

this.t0 = cv2.getTickCount() / cv2.getTickFrequency() * 1000;

this.bpms = [];

```

```

this.bpm = 0;

this.face_cascade = new
cv2.CascadeClassifier('haarcascade_frontalface_alt.xml');

this.face_rect = [1, 1, 2, 2];

this.last_center = [0, 0];

this.last_wh = [0, 0];

this.output_dim = 13;

this.trained = false;

this.idx = 1;

this.find_faces = true;

}

toggleFindFaces() {

this.find_faces = !this.find_faces;

return this.find_faces;

}

run(cam) {

this.times.push(cv2.getTickCount() / cv2.getTickFrequency() * 1000 - this.t0);

this.frame_out = this.frame_in.copy();

const gray = cv2.cvtColor(this.frame_in, cv2.COLOR_BGR2GRAY);

const col = [100, 255, 100];

```

```
if (this.find_faces) {

this.data_buffer = [];

this.times = [];

this.trained = false;

const detected = this.face_cascade.detectMultiScale(gray, 1.3, 4, [50, 50], [500,
500]);

if (detected.length > 0) {

detected.sort((a, b) => a[2] * a[3] - b[2] * b[3]);

if (this.shift(detected[detected.length - 1]) > 10) {

this.face_rect = detected[detected.length - 1];

}

const forehead1 = this.get_subface_coord(0.5, 0.18, 0.25, 0.15);

this.draw_rect(this.face_rect, [255, 0, 0]);

cv2.putText(this.frame_out, "Face", [this.face_rect[0], this.face_rect[1]],
cv2.FONT_HERSHEY_PLAIN, 1.5, col);

this.draw_rect(forehead1);

cv2.putText(this.frame_out, "Forehead", [forehead1[0], forehead1[1]],
cv2.FONT_HERSHEY_PLAIN, 1.5, col);

return;

}
```

```

}

if (this.face_rect === [1, 1, 2, 2]) {

return;

}

const forehead1 = this.get_subface_coord(0.5, 0.18, 0.25, 0.15);

this.draw_rect(forehead1);

const vals = this.get_subface_means(forehead1);

this.data_buffer.push(vals);

const L = this.data_buffer.length;

if (L > this.buffer_size) {

this.data_buffer = this.data_buffer.slice(-this.buffer_size);

this.times = this.times.slice(-this.buffer_size);

}

const processed = this.data_buffer;

this.samples = processed;

if (L > 10) {

this.output_dim = processed.length;

this.fps = L / (this.times[this.times.length - 1] - this.times[0]);

const even_times = np.linspace(this.times[0], this.times[this.times.length - 1],
L);

```

```
const interpolated = np.interp(even_times, this.times, processed);

const hamming = np.hamming(L);

const interpolated_hamming = interpolated * hamming;

const interpolated_hamming_mean = np.mean(interpolated_hamming);

const interpolated_hamming_normalized = interpolated_hamming -
interpolated_hamming_mean;

const raw = np.fft.fft(interpolated_hamming_normalized);

const phase = np.angle(raw);

this.fft = np.abs(raw);

this.freqs = np.fft.fftfreq(L, 1 / this.fps).slice(0, L / 2 + 1);

const freqs_60 = this.freqs * 60;

const idx = np.where((freqs_60 > 50) & (freqs_60 < 180))[0];

const pruned = this.fft[idx];

const phase_pruned = phase[idx];

const freqs_pruned = freqs_60[idx];

this.fft = pruned;

phase = phase_pruned;

this.freqs = freqs_pruned;

const idx2 = np.argmax(this.fft);

const tmp = (np.sin(phase[idx2]) + 1) / 2;
```

```

const alpha = 0.9 * tmp + 0.1;

const beta = 1 - tmp;

this.bpm = this.freqs[idx2];

this.idx += 1;

const subface_coord = this.get_subface_coord(0.5, 0.18, 0.25, 0.15);

const r = [];

const g = [];

const b = [];

for (let i = 0; i < subface_coord[3]; i++) {

  for (let j = 0; j < subface_coord[2]; j++) {

    r.push(alpha * this.frame_in[subface_coord[1] + i][subface_coord[0] + j][0]);

    g.push(alpha * this.frame_in[subface_coord[1] + i][subface_coord[0] + j][1] +
      beta * gray[subface_coord[1] + i][subface_coord[0] + j]);

    b.push(alpha * this.frame_in[subface_coord[1] + i][subface_coord[0] + j][2]);

  }

}

for (let i = 0; i < subface_coord[3]; i++) {

  for (let j = 0; j < subface_coord[2]; j++) {

    this.frame_out[subface_coord[1] + i][subface_coord[0] + j] = [b[i], g[i], r[i]];

  }

}

```

```

}

const face_coord = this.face_rect;

this.slices = [this.frame_out.slice(face_coord[1], face_coord[1] + face_coord[3],
face_coord[0], face_coord[0] + face_coord[2])];

const gap = (this.buffer_size - L) / this.fps;

if (gap !== 0) {

const text = "(estimate: " + this.bpm + " bpm, wait " + gap + " s)";

} else {

const text = "(estimate: " + this.bpm + " bpm)";

}

const tsize = 1;

cv2.putText(this.frame_out, text, [subface_coord[0] - subface_coord[2] / 2,
subface_coord[1]], cv2.FONT_HERSHEY_PLAIN, tsize, col);

}

}

}

class ArgumentParser {

constructor(description) {

this.description = description;

this.arguments = [];

```

```

}

addArgument(name, help, type = null, defaultValue = null) {

this.arguments.push({ name, help, type, defaultValue });

}

parseArgs(args) {

const namespaceDict = this.arguments.reduce((acc, arg) => {

acc[arg.name] = arg.defaultValue;

return acc;

}, {});

for (let i = 0; i < args.length; i += 2) {

const argName = args[i];

const argValue = args[i + 1];

if (argName in namespaceDict) {

namespaceDict[argName] = argValue;

}

}

return { namespaceDict };

}

}

```

```
class Argument {  
  
    constructor(name, help, type, defaultValue) {  
  
        this.name = name;  
  
        this.help = help;  
  
        this.type = type;  
  
        this.defaultValue = defaultValue;  
  
    }  
  
}  
  
class Serial {  
  
    constructor(serialPort, baudRate) {  
  
        this.serialPort = serialPort;  
  
        this.baudRate = baudRate;  
  
    }  
  
}  
  
class Socket {  
  
    constructor(ip, port) {  
  
        this.ip = ip;  
  
        this.port = port;  
  
    }  
  
}}
```

```

class File {

constructor(filename) {

this.filename = filename;

}

}

const parser = new ArgumentParser("Webcam pulse detector.");

parser.addArgument("--serial", "serial port destination for bpm data");

parser.addArgument("--baud", "Baud rate for serial transmission");

parser.addArgument("--udp", "udp address:port destination for bpm data");

const parsedArgs = parser.parseArgs([]);

const app = new Program(parsedArgs);

while (true) {

app.mainLoop();

}

pulse.py

import cv2

import numpy as np

class Pulse:

def __init__(self):

```

```
self.maxSignalSize = 100

self.relativeMinFaceSize = 0.4

self.deleteFaceIn = 1

self.holdPulseFor = 30

self.fps = 0

self.faceDetectionEnabled = True

self.evmMagnify = True

self.evmAlpha = 100

self.classifier = cv2.CascadeClassifier()

self.now = 0

self.lastFaceDetectionTimestamp = 0

self.lastBpmTimestamp = 0

self.minFaceSize = None

self.bboxes = []

self.faces = []

self.nextFaceId = 0

def load(self, filename):

    self.classifier.load(filename)

def start(self, width, height):
```

```

self.now = 0

self.lastFaceDetectionTimestamp = 0

self.lastBpmTimestamp = 0

self.minFaceSize = (min(width, height) * self.relativeMinFaceSize, min(width,
height) * self.relativeMinFaceSize)

self.faces.clear()

self.nextFaceId = 1

def on_frame(self, frame):

self.now = cv2.getTickCount()

if self.faceDetectionEnabled:

if (self.now - self.lastFaceDetectionTimestamp) * 1000.0 /
cv2.getTickFrequency() >= 1000:

self.lastFaceDetectionTimestamp = cv2.getTickCount()

gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)

self.classifier.detectMultiScale(frame, self.bboxes, 1.1, 3, 0, self.minFaceSize)

if len(self.faces) <= len(self.bboxes):

for i in range(len(self.faces)):

face = self.faces[i]

boxIndex = face.nearest_box(self.bboxes)

face.delete_in = self.deleteFaceIn

```

```
face.update_box(self.bboxes[boxIndex])

self.on_face(frame, face, self.bboxes[boxIndex])

self.bboxes.pop(boxIndex)

for i in range(len(self.bboxes)):

    self.faces.append(Face(self.nextFaceId, self.bboxes[i], self.deleteFaceIn))

    self.on_face(frame, self.faces[-1], self.bboxes[i])

else:

    for i in range(len(self.faces)):

        self.faces[i].selected = False

    for i in range(len(self.bboxes)):

        faceIndex = self.nearest_face(self.bboxes[i])

        face = self.faces[faceIndex]

        face.selected = True

        face.delete_in = self.deleteFaceIn

        face.update_box(self.bboxes[i])

        self.on_face(frame, face, self.bboxes[i])

    for i in range(len(self.faces)):

        face = self.faces[i]

        if not face.selected:
```

```
if face.delete_in <= 0:

    self.faces.pop(i)

    i -= 1

else:

    face.delete_in -= 1

    self.on_face(frame, face, face.box)

else:

    for i in range(len(self.faces)):

        face = self.faces[i]

        self.on_face(frame, face, face.box)

    else:

        if len(self.faces) == 0 or self.faces[-1].id != FaceDetection.disabled_face_id:

            self.faces.clear()

            r = self.relativeMinFaceSize

            w = frame.shape[1]

            h = frame.shape[0]

            tl = ((1 - r) * w / 2, (1 - r) * h / 2)

            br = ((1 + r) * w / 2, (1 + r) * h / 2)

            face = Face(self.nextFaceId, (tl, br), self.deleteFaceIn)
```

```

face.evm.box = (tl, br)

self.faces.append(face)

FaceDetection.disabled_face_id = face.id

self.on_face(frame, self.faces[-1], self.faces[-1].box)

else:

if len(self.faces) == 0 or self.faces[-1].id != FaceDetection.disabled_face_id:

self.faces.clear()

r = self.relativeMinFaceSize

w = frame.shape[1]

h = frame.shape[0]

tl = ((1 - r) * w / 2, (1 - r) * h / 2)

br = ((1 + r) * w / 2, (1 + r) * h / 2)

face = Face(self.nextFaceId, (tl, br), self.deleteFaceIn)

face.evm.box = (tl, br)

self.faces.append(face)

FaceDetection.disabled_face_id = face.id

self.on_face(frame, self.faces[-1], self.faces[-1].box)

def nearest_face(self, box):

index = -1

```

```

min_dist = -1

p = None

for i in range(len(self.faces)):

    if not self.faces[i].selected:

        index = i

        p = box[0] - self.faces[i].box[0]

        min_dist = p[0] * p[0] + p[1] * p[1]

        break

    if index == -1:

        return -1

    for i in range(index + 1, len(self.faces)):

        if not self.faces[i].selected:

            p = box[0] - self.faces[i].box[0]

            dist = p[0] * p[0] + p[1] * p[1]

            if dist < min_dist:

                min_dist = dist

                index = i

    return index

def on_face(self, frame, face, box):

```

```

roi = frame[face.evm.box[0][1]:face.evm.box[1][1],
face.evm.box[0][0]:face.evm.box[1][0]]

if self.evmMagnify:

if face.evm.evm.first or face.evm.evm.alpha != self.evmAlpha:

face.reset()

face.evm.evm.alpha = self.evmAlpha

face.evm.evm.on_frame(frame[face.evm.box[0][1]:face.evm.box[1][1],
face.evm.box[0][0]:face.evm.box[1][0]])

else:

if not face.evm.evm.first:

face.reset()

if face.raw.shape[0] >= self.maxSignalSize:

face.raw = np.roll(face.raw, -1, axis=0)

face.raw[-1] = np.mean(roi[:, :, 1])

face.timestamps = np.roll(face.timestamps, -1, axis=0)

face.timestamps[-1] = cv2.getTickCount()

else:

face.raw = np.append(face.raw, np.mean(roi[:, :, 1]))

face.timestamps = np.append(face.timestamps, cv2.getTickCount())

raw_stddev = np.std(face.raw)

```

```

stable = raw_stddev <= (self.evmMagnify * self.evmAlpha * 0.045 + 1)

if stable:

    current_fps = self.fps

    if current_fps == 0:

        diff = (face.timestamps[-1] - face.timestamps[0]) * 1000.0 /
cv2.getTickFrequency()

        current_fps = face.timestamps.shape[0] * 1000 / int(diff)

        self.detrend(face.raw, face.pulse, current_fps / 2)

        self.normalization(face.pulse, face.pulse)

        self.mean_filter(face.pulse, face.pulse)

        self.peaks(face)

    else:

        face.existsPulse = False

        if self.faceDetectionEnabled:

            face.reset()

            if face.existsPulse:

                self.bpm(face)

            if not face.existsPulse:

                if face.pulse.shape[0] == face.raw.shape[0]:

                    face.pulse = np.zeros((face.raw.shape[0], 1))

```

```

else:

    face.pulse = np.zeros((face.raw.shape[0], 1))

    face.peaks.clear()

    face.bpms = face.bpms[:face.bpms.shape[0]]

    face.bpm = 0

    self.draw(frame, face, box)

    def peaks(self, face):

        face.peaks.clear()

        last_index = 0

        l_peak_ind = 0

        last_peak_timestamp = face.timestamps[0]

        last_peak_value = face.pulse[0]

        peak_value_threshold = 0

        for i in range(1, face.raw.shape[0]):

            diff = (face.timestamps[i] - face.timestamps[last_index]) * 1000.0 /
            cv2.getTickFrequency()

            if diff >= 200:

                rpi = np.zeros((2,), dtype=int)

                p_v = 0

                cv2.minMaxIdx(face.pulse[last_index:i + 1], None, p_v, None, rpi)

```

```

pi = last_index + rpi[0]

if p_v > pvt and last_index < pi and pi < i:

    peak_timestamp = face.timestamps[peak_index]

    peak_diff = (peak_timestamp - last_peak_timestamp) * 1000.0 /
cv2.getTickFrequency()

    if peak_diff <= 200 and p_v > lpv:

        face.peaks.pop()

    if peak_diff > 200 or p_v > lpv:

        face.peaks.push(p_v, peak_timestamp, peapVkValue)

    lpi = pi

    last_peak_timestamp = peak_timestamp

    lpv = pv

    peak_value_threshold = 0.6 * np.mean(face.peaks.values)

    lastIndex = i

    peak_values_stddev = np.std(face.peaks.values)

    diff = (face.timestamps[-1] - face.timestamps[0]) / cv2.getTickFrequency()

    peak_timestamps_stddev = 0

    if face.peaks.indices.shape[0] >= 3:

        peak_timestamps_stddev = np.std((face.peaks.timestamps[1:] -
face.peaks.timestamps[:-1]) / cv2.getTickFrequency())

```

```

valid_pulse = (

2 <= face.peaks.indices.shape[0] and

40.0 / 60.0 * diff <= face.peaks.indices.shape[0] and

face.peaks.indices.shape[0] <= 240.0 / 60.0 * diff and

peak_values_stddev <= 0.5 and

peak_timestamps_stddev <= 0.5

)

if not face.existsPulse and valid_pulse:

face.noPulseIn = self.holdPulseFor

face.existsPulse = True

elif face.existsPulse and not valid_pulse:

if face.noPulseIn > 0:

face.noPulseIn -= 1

else:

face.existsPulse = False

def bpm(self, face):

power_spectrum = np.zeros(face.raw.shape, dtype=complex)

cv2.dft(face.pulse, power_spectrum)

total = face.raw.shape[0]

```

```

low = int(total * 40.0 / 60.0 / self.fps) + 1

high = int(total * 240.0 / 60.0 / self.fps) + 1

power_spectrum[:min(low, total)] = 0

power_spectrum = power_spectrum[:min(total - high, total)]

power_spectrum = np.power(power_spectrum, 2)

if not power_spectrum.size == 0:

    idx = np.zeros((2,), dtype=int)

    cv2.minMaxIdx(power_spectrum, None, None, None, idx)

    face.bpms = np.append(face.bpms, idx[0] * self.fps * 30.0 / total)

    if face.bpm == 0 or (self.now - self.lastBpmTimestamp) * 1000.0 /
    cv2.getTickFrequency() >= 1000:

        self.lastBpmTimestamp = cv2.getTickCount()

        face.bpm = np.mean(face.bpms)

        face.bpms = face.bpms[:face.bpms.shape[0]]

        if face.bpm <= 40:

            face.existsPulse = False

def draw(self, frame, face, box):

    cv2.rectangle(frame, box, (255, 0, 0), 1)

    cv2.rectangle(frame, face.box, (255, 0, 0), 2)

    cv2.rectangle(frame, face.evm.box, (0, 255, 0), 1)

```

```

bl = (face.box[0][0], face.box[1][1])

for i in range(face.raw.shape[0]):

    g = (bl[0] + i, bl[1] - face.raw[i] + 50)

    cv2.line(frame, g, g, (0, 255, 0), 1)

    g = (bl[0] + i, bl[1] - face.pulse[i] * 10 - 50)

    cv2.line(frame, g, g, (0, 0, 255), 2 if face.existsPulse else 1)

for i in range(face.peaks.indices.shape[0]):

    index = face.peaks.indices[i]

    g = (bl[0] + index, bl[1] - face.pulse[index] * 10 - 50)

    cv2.circle(frame, g, 1, (255, 0, 0), 2)

def detrend(self, src, dst, windowSize):

    dst = np.zeros(src.shape, dtype=src.dtype)

    for i in range(src.shape[0]):

        start = max(0, i - windowSize)

        end = min(src.shape[0], i + windowSize + 1)

        mean, _ = cv2.meanStdDev(src[start:end])

        dst[i] = src[i] - mean[0]

    def normalization(self, src, dst):

        dst = np.zeros(src.shape, dtype=src.dtype)

```

```

min_val, max_val = cv2.minMaxLoc(src)

if max_val - min_val > 0:

    dst = (src - min_val) / (max_val - min_val)

def mean_filter(self, src, dst):

    dst = np.zeros(src.shape, dtype=src.dtype)

    for i in range(src.shape[0]):

        start = max(0, i - 2)

        end = min(src.shape[0], i + 3)

        mean, _ = cv2.meanStdDev(src[start:end])

        dst[i] = mean[0]

class Face:

    def __init__(self, id, box, deleteIn):

        self.Id = id

        self.Box = box

        self.DeleteIn = deleteIn

        self.UpdateBox(self.Box)

        self.ExistsPulse = False

        self.NoPulseIn = 0

    def NearestBox(self, boxes):

```

```

if len(boxes) == 0:

    return -1

    index = 0

    p = self.Box.tl() - boxes[0].tl()

    min = p.x * p.x + p.y * p.y

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

        p = self.Box.tl() - boxes[i].tl()

        d = p.x * p.x + p.y * p.y

        if d < min:

            min = d

            index = i

    return index

def UpdateBox(self, a):

    p = self.Box.tl() - a.tl()

    d = (p.x * p.x + p.y * p.y) / (self.Box.width / 3.0) ** 2.0

    cv2.interpolate(self.Box, a, self.Box, min(1.0, d))

    self.Evm.Box = Rect(self.Box.tl() + Point(self.Box.width * 0.275,
self.Box.height * 0.425), self.Box.tl() + Point(self.Box.width * 0.725,
self.Box.height * 0.575))

def Reset(self):

```

```
self.Evm.Evm.First = True

self.Raw.popBack(self.Raw.rows)

self.Timestamps.popBack(self.Timestamps.rows)

class Evm:

    def __init__(self):

        self.Box = Rect()

        self.Out = Mat()

        self.Evm = EvmEvm()

    class EvmEvm:

        def __init__(self):

            self.First = True

            self.Alpha = 100

            def OnFrame(self, frame, roi):

                # apply Eulerian video magnification on face box

            class Peaks:

                def __init__(self):

                    self.Indices = Mat()

                    self.Timestamps = Mat()

                    self.Values = Mat()
```

```
def Push(self, index, timestamp, value):

    self.Indices.pushBack(index)

    self.Timestamps.pushBack(timestamp)

    self.Values.pushBack(value)

def Pop(self):

    self.Indices.popBack(min(self.Indices.rows, 1))

    self.Timestamps.popBack(min(self.Timestamps.rows, 1))

    self.Values.popBack(min(self.Values.rows, 1))

def Clear(self):

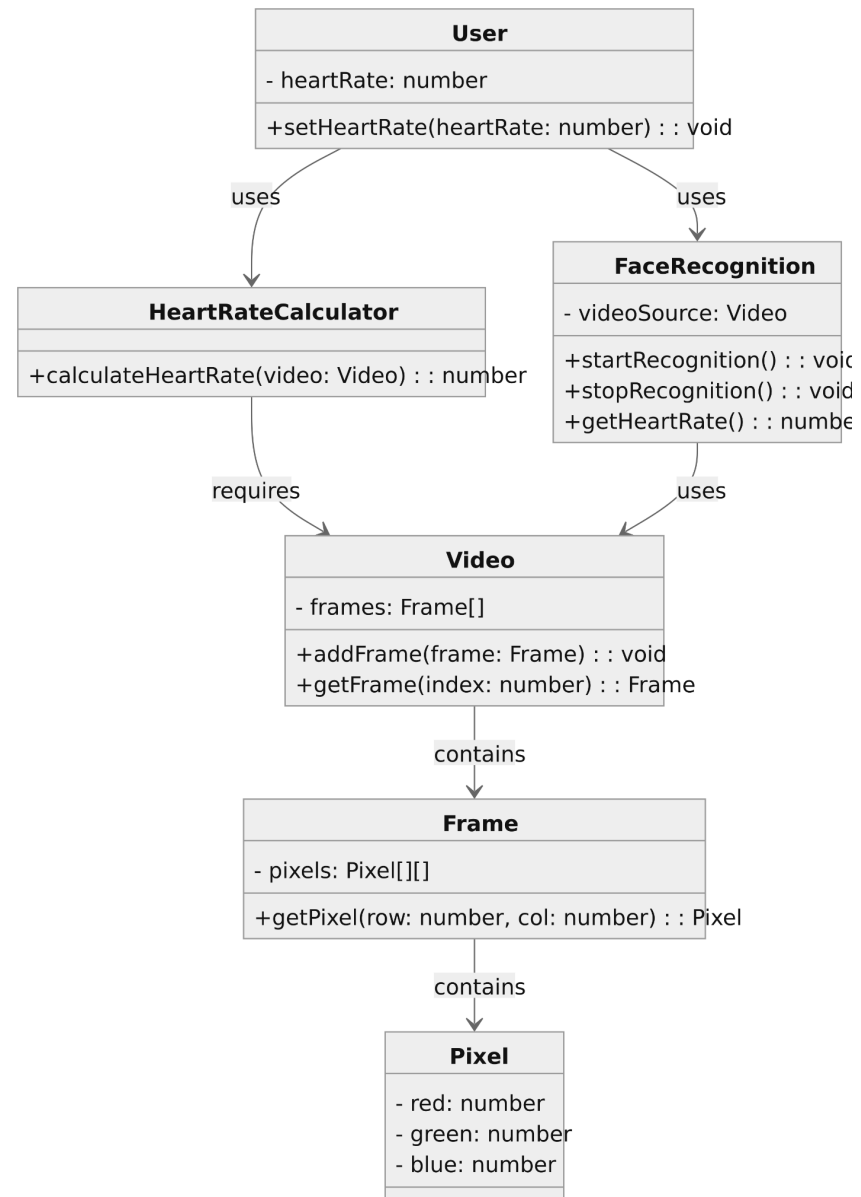
    self.Indices.popBack(self.Indices.rows)

    self.Timestamps.popBack(self.Timestamps.rows)

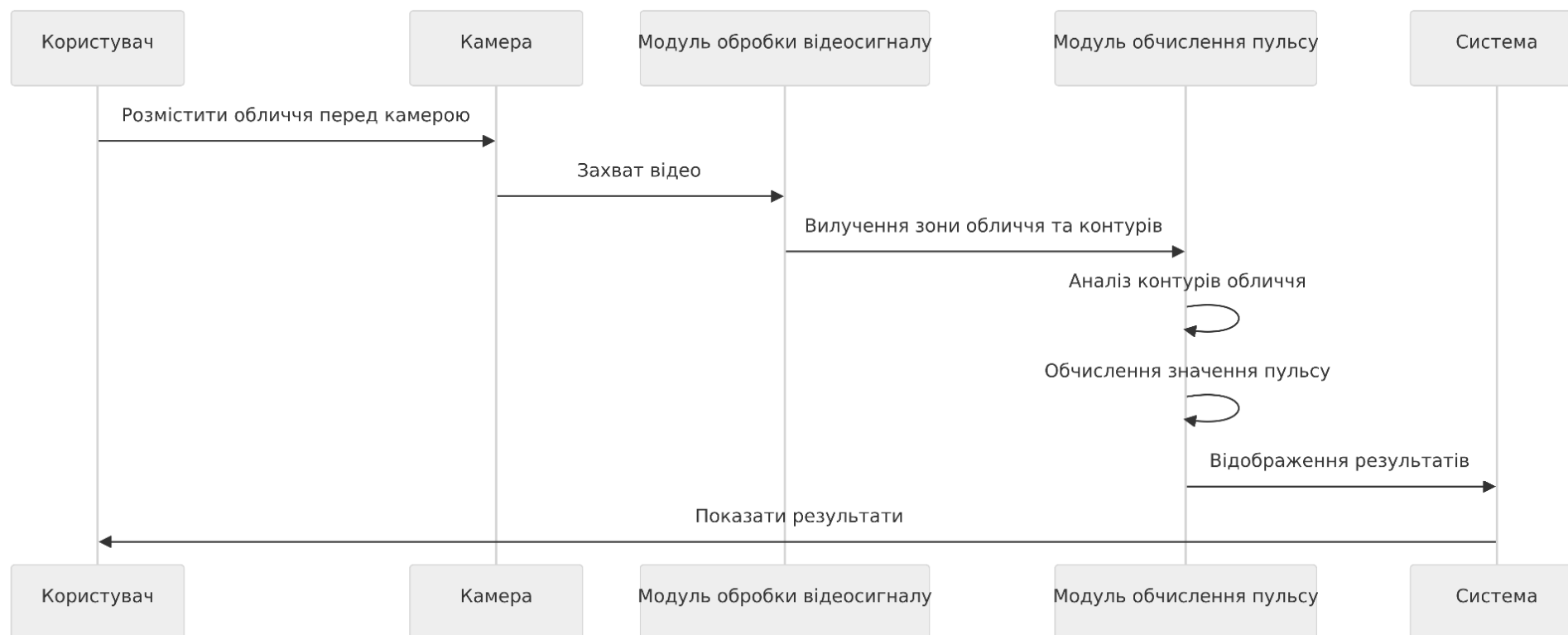
    self.Values.popBack(self.Values.rows)
```

ДОДАТОК Б
ГРАФІЧНИЙ МАТЕРІАЛ

СХЕМА СТРУКТУРНА КЛАСІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



ДІАГРАМА ПОСЛІДОВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



ДОДАТОК Г
РЕЗУЛЬТАТИ ПЕРЕВІРКИ НА ПЛАГІАТ

Ім'я користувача:
Лісовиченко Олег Іванович

Дата перевірки:
14.01.2024 10:59:38 EET

Дата звіту:
14.01.2024 11:02:04 EET

ID перевірки:
1016059536

Тип перевірки:
Doc vs Internet + Library

ID користувача:
76913

Назва документа: ІП-22мп_Онофрійчук_ПЗ

Кількість сторінок: 64 Кількість слів: 10182 Кількість символів: 78595 Розмір файлу: 6.71 MB ID файлу: 1015763086

7.09% Схожість

Найбільша схожість: 1.63% з джерелом з Бібліотеки (ID файлу: 1015681882)

4.08% Джерела з Інтернету 122 Сторінка 66

6.67% Джерела з Бібліотеки 487 Сторінка 67

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел