

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ННК “Інститут прикладного системного аналізу”

(повна назва інституту/факультету)

Кафедра Системного проектування

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ А.І.Петренко
(підпис) (ініціали, прізвище)

“ _____ ” _____ 2016 р.

Дипломна робота

першого (бакалаврського)

рівня вищої освіти

(першого (бакалаврського), другого (магістерського))

зі спеціальності 7.05010102, 8.05010102 Інформаційні технології проектування

7.05010103, 8.05010103 Системне проектування

(код та назва спеціальності)

на тему: **Програма вимірювання частоти серцевих скорочень по
відеозображенню обличчя людини**

Виконав (-ла): студент (-ка) 4 курсу, групи ДА-22
(шифр групи)

Калманович Антон Віталійович

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник _____ ст. викладач Бритов О.А.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант _____

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище, ініціали)

_____ (підпис)

Рецензент _____ доцент к.т.н. Тимошенко Ю.О.

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Нормоконтроль _____ ст. викладач Бритов О.А.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2016 року

Національний технічний університет України
«Київський політехнічний інститут»

Факультет (інститут) ННК “Інститут прикладного системного аналізу”
(повна назва)

Кафедра Системного проектування
(повна назва)

Рівень вищої освіти Перший (Бакалаврський) (перший
(бакалаврський), другий (магістерський) або спеціаліста)

Спеціальність 7.050102, 8.050102 Інформаційні технології проектування
7.050103, 8.050103 Системне проектування
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри _____
А.І.Петренко
(підпис) (ініціали, прізвище)

« ___ » _____ 2016 р.

ЗАВДАННЯ на дипломний проект (роботу) студенту
Калмановичу Антону Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Програма вимірювання частоти серцевих скорочень по відеозображенню обличчя людини

керівник проекту (роботи) Бритов Олексій Анатолійович, ст. викладач,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « ___ » _____ 2016 р. № _____

2. Строк подання студентом проекту (роботи) 08.06.2016

3. Вихідні дані до проекту (роботи)

USB Веб-камера

Персональний комп'ютер

Форма реалізації – у вигляді програми на мові C# та додаткових бібліотек EmguCV, LomontFFT та Accord.Statistics.

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити)

1. Розглянути можливі варіанти реалізації аналізу відеозображень для знаходження частоти серцевих скорочень.
2. Розробити різні варіанти алгоритмів для знаходження ЧСС по відеозображенню.
3. Порівняти роботу алгоритмів та знайти найоптимальніші з них.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів тощо)

1. Ілюстрація алгоритму роботи FastICA – плакат.
2. UML-діаграма програмного продукту – плакат.
3. Результати порівняння роботи алгоритмів – плакат.

6. Консультанти розділів проекту (роботи)

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

7. Дата видачі завдання 01.02.2016

Календарний план

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання	01.02.2016	
2	Збір інформації	01.03.2016	
3	Вивчення варіантів реалізації методів аналізу та вибір варіантів для розробки	15.03.2016	
4	Розробка алгоритмів аналізу та інтерфейсу програми	15.04.2016	
5	Порівняння алгоритмів аналізу та висновки	15.05.2016	
6	Аналіз економічної обґрунтованості	01.06.2016	
7	Оформлення дипломної роботи	13.06.2016	
8	Отримання допуску до захисту та подача роботи в ДЕК	21.06.2016	

Студент

_____ (підпис)

_____ (ініціали, прізвище)

Керівник проекту (роботи)

_____ (підпис)

_____ (ініціали, прізвище)

АНОТАЦІЯ

бакалаврської дипломної роботи Калмановича Антона Віталійовича
на тему «Програма вимірювання частоти серцевих скорочень по відеозображенню
обличчя людини»

Дана дипломна робота присвячена розробці програмного продукту для вимірювання частоти серцевих скорочень по відеозображенню обличчя людини та аналізу і порівнянню існуючих алгоритмів, що використовуються у цій галузі, а саме порівняльному аналізу використання алгоритму «аналізу незалежних компонентів» (ІСА). Метою роботи є розробка програмного модуля, що вимірює серцевий пульс за відеозображенням обличчя користувача.

В роботі розглянуто основні існуючі способи використання аналізу незалежних компонентів для вимірювання пульсу, було виконано порівняння якості їх роботи та запропоновано використання двох чи більше відеокамер для досягнення найбільш точних результатів вимірювання.

Загальний обсяг роботи: 73 сторінки, 38 рисунків, 6 таблиць, 33 бібліографічних найменування.

Ключові слова: серце, частота серцевих скорочень, пульс, серцебиття, відеозображення, аналіз незалежних компонентів, перетворення Фур'є.

АННОТАЦИЯ

Бакалаврской дипломной работы Калмановича Антона Витальевича
по теме «Программа измерения частоты сердцебиения по видеоизображению лица
человека»

Данная дипломная работа посвящена разработке программы для измерения частоты сердечных сокращений по видеоизображению лица человека и анализу и сравнению существующих алгоритмов, используемых в этой области, а именно сравнительному анализу использования алгоритма «анализа независимых компонентов» (ICA). Целью работы является разработка программного модуля, измеряющего сердечный пульс по видеоизображению лица пользователя.

В работе рассмотрены основные существующие способы использования анализа независимых компонентов для измерения пульса, было выполнено сравнение качества их работы и предложено использование двух или более видеокамер для достижения наиболее точных результатов измерения.

Общий объем работы 73 страницы, 38 рисунков, 6 таблиц, 33 библиографических наименования.

Ключевые слова: сердце, частота сердечных сокращений, пульс, сердцебиение, видеоизображение, анализ независимых компонентов, преобразование Фурье.

ANNOTATION

For the bachelor thesis of Kalmanovich Anton Vitalievich
on «Software for cardiac pulse measurement using facial video imaging»

This thesis is devoted to the development of a software which measures heart rate using the video of the human face, and the analysis and comparison of existing algorithms used in this area, namely independent component analysis (ICA). The aim is to develop a software that measures human heart rate using video picture of the user.

This paper examines the main existing ways of using independent components analysis in heart rate measurement, describes comparison of the measuring quality of aforementioned algorithms and suggests the use of two or more web-cameras to achieve the most accurate measurement results.

Total volume of work: 73 pages, 38 pictures, 6 tables, 33 links.

Keywords: heart, heart rate, cardiac pulse, heartbeat, video, independent component analysis, Fourier transform.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ ...	11
ВСТУП	12
1. ОГЛЯД ВИКОРИСТАНИХ АЛГОРИТМІВ	15
1.1. Знаходження обличчя на кадрі.....	15
1.2. Алгоритм Віоли-Джонса.....	16
1.2.1. Принцип скануючого вікна	17
1.2.2. Інтегральне представлення зображень	18
1.2.3. Ознаки Хаара.....	19
1.2.4. Сканування вікна	21
1.2.5. Навчання класифікатора в методі Віоли-Джонса	23
1.2.6. Застосований в алгоритмі бустінг	24
1.2.7. Каскадна модель алгоритму	25
1.3. Обчислення показників зміни кольору обличчя	27
1.3.1. Обмеження області обличчя.....	28
1.3.2. Використання маски обличчя.....	28
1.4. Нормалізація сигналу.....	30
1.5. Ресемплінг сигналу.....	32
1.6. Перетворення Фур'є.....	32
1.7. Сліпий поділ джерел	33
1.8. Висновки	38
2. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	39
2.1. Вимоги до програмної реалізації	39

2.2.	Зчитування та запис сигналів з веб-камери	39
2.2.1.	Клас CameraClass	39
2.2.2.	Клас Processor	40
2.2.3.	Підкласи класу Processor	41
2.2.4.	Клас FFTObject	42
2.3.	Висновки	42
3.	ПРАКТИЧНЕ ПОРІВНЯННЯ МЕТОДІВ АНАЛІЗУ	43
3.1.	Використання масок для фільтрації зображення обличчя	43
3.2.	Використання АНК для трьох каналів кольору	45
3.3.	Використання АНК для двох камер і всіх каналів кольору	47
3.4.	Використання АНК для двох камер і одного каналу кольору	49
3.5.	Висновки	50
4.	ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	51
4.1.	Аналіз ринку	52
4.2.	Постановка задачі	52
4.2.1.	Обґрунтування функцій програмного продукту	53
4.2.2.	Варіанти реалізації основних функцій	54
4.3.	Обґрунтування системи параметрів ПП	56
4.3.1.	Опис параметрів	56
4.3.2.	Кількісна оцінка параметрів	56
4.3.3.	Аналіз експертного оцінювання параметрів	59
4.4.	Аналіз рівня якості варіантів реалізації функцій	61
4.5.	Економічний аналіз розробки ПП	62

	10
4.6. Вибір кращого варіанта ПП техніко-економічного рівня	66
4.7. Висновки	67
ВИСНОВКИ.....	68
ПЕРЕЛІК ПОСИЛАНЬ	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЧСС	Частота серцевих скорочень
ЕКГ	Електрокардіографія
ШПФ	Швидке перетворення Фур'є
ФПГ	Фотоплетізографія
СПС	Сліпий поділ сигналів
АНК	Аналіз незалежних компонентів

ВСТУП

Регулярні оцінки серцево-судинної функції відіграють важливу роль в спостереженні за серцево-судинними захворюваннями та процесом лікування. Частота серцевих скорочень в спокої є одним з найпростіших та основних параметрів серцево-судинної системи. Її аномальні зміни були ідентифіковані як фактор ризику (у порівнянні з курінням, дисліпідемією або гіпертензією) для багатьох серцево-судинних захворювань [1].

На сьогодні методи для вимірювання ЧСС, такі як електрокардіограма (ЕКГ), вимагають, щоб пацієнти наносили клей або нагрудні ремені, які можуть викликати роздратування шкіри і дискомфорт. Комерційні датчики, пульсоксиметри, які прикріплюються до кінчиків пальців або мочок вуха також незручні для пацієнтів - пружинні затискачі можуть викликати біль, якщо їх не знімати протягом тривалого часу. Можливість моніторингу фізіологічного стану пацієнта за допомогою безконтактного засобу являє собою привабливу перспективу, що сприятиме підвищенню якості надання первинної медичної допомоги, профілактики хронічних захворювань тощо. Наприклад, ідея проведення фізіологічних вимірювань пульсу на обличчі була продемонстрована за допомогою термічних камер, що відстежують зміни температури обличчя [2].

Хоча безконтактні методи не можуть надати інформацію щодо електричних явищ, що відбуваються у серцевому м'язі, як дає ЕКГ, проте вони можуть дозволити вести довгостроковий моніторинг інших фізіологічних сигналів, таких як частота серцевих скорочень або частота дихання, записуючи їх безперервно в зручному для користувача вигляді. Крім того, така технологія може звести до мінімуму кількість кабелів і перешкод, пов'язаних з моніторингом стану новонароджених, довгостроковим моніторингом епілепсії, спостереженням за пацієнтами, що отримали опік або травми, дослідженням сну, і іншими випадками, коли довгострокове вимірювання ЧСС має важливе значення. Використання

фотоплетізографії, дешевого і неінвазивного засобу відстеження серцево-судинної пульсової хвилі через зміни у відбитому або поглинутому світлі, для безконтактного вимірювання фізіологічних показників, вже було досліджено недавно [3-4]. Цей електрооптичний метод може дати цінну інформацію про серцево-судинну систему, таку як частота серцевих скорочень, насичення киснем крові, кров'яний тиск та серцевий викид [5]. Як правило, ФПГ завжди були реалізовані з використанням спеціальних джерел світла (наприклад, червоне і/або інфрачервоне випромінювання), але деякі дослідження показали [6-7], що вимірювання ЧСС можуть бути досягнуті за допомогою цифрових відеокамер, орієнтованих на нормальне денне світло.

Крім того, ФПГ, як відомо, вразливо до змін у вхідних сигналах, що викликаються рухами людини [8-9] і подолання артефактів руху являє собою одну з найскладніших проблем у даній галузі. У більшості випадків, шум падає в тій же смузі частот, що і фізіологічний сигнал, що представляє інтерес для аналізу, таким чином роблячи використання смугової фільтрації неефективною. Для розвитку клінічно корисної технології є необхідність в допоміжних функціях для зменшення артефактів руху за рахунок ефективного і надійного аналізу зображень. Один зі способів відділення шуму від фізіологічних сигналів це метод сліпого поділу сигналів (Blind signal separation). Це метод відновлення неспостережуваних оригінальних сигналів (джерел) зі спостережуваної суміші цих сигналів без жодної інформації про процес змішування цих джерел. Як правило, спостережувані сигнали отримуються з набору датчиків, де кожен датчик приймає різну комбінацію оригінальних джерел. Одним із способів реалізації СПС є аналіз незалежних компонентів (Independent component analysis). Використання цього методу в біології та медицині швидко поширюється [10], наприклад, при видаленні шуму в електрокардіограмі [11] чи електроенцефалограмі [12], відділення ЕКГ матері від ЕКГ плоду, записаних одночасно [13], та для інших схожих випадків. АНК також застосовується для зменшення артефактів руху при вимірах ФПГ [14].

Цей алгоритм також широко використовується при спробах отримати частоту серцебиття людини за допомогою зображення обличчя з веб-камери [15-19]. В даній роботі будуть досліджені різні варіанти використання цього алгоритму та порівняно їх з іншими підходами до вирішення даної задачі.

Метою розробки є створення програмного продукту, який зможе наявно показати різницю між різними варіантами реалізації алгоритму аналізу відеозображення обличчя людини для знаходження частоти серцевих скорочень.

Для досягнення поставленої мети необхідне виконання вирішення наступних задач:

1. Аналіз існуючих праць про знаходження ЧСС за допомогою відеозображення обличчя.
2. Вибір та реалізація найпоширеніших варіантів алгоритмів аналізу зображення.
3. Створення програмного продукту, що дозволяє вільно передивлятися результати роботи різних алгоритмів у реальному часі.
4. Порівняння роботи різних алгоритмів та оцінка їх ефективності.

1. ОГЛЯД ВИКОРИСТАНИХ АЛГОРИТМІВ

У даному розділі представлено короткий опис алгоритмів, що використовуються при аналізі відеозображення обличчя людини для знаходження частоти його серцевих скорочень.

Процесинг відеозображення починається з аналізу кожного кадру відео послідовно. Перший етап – це знаходження положення обличчя на кадрі.

1.1. Знаходження обличчя на кадрі

Існує багато методів знаходження положення обличчя на кадрі відео.

1. Знаходження обличчя за кольором шкіри.

Маючи інформацію про колір зображення, можна знаходити область обличчя за допомогою порівняння пікселів зображення з типовим кольором шкіри. Цей метод має багато недоліків, включаючи чутливість до освітлення та неможливість роботи з усіма типами шкіри.

2. Знаходження обличчя за рухами.

В поточному відео майже завжди обличчя людини буде рухоме. Тому деякі методи знаходження його положення знаходять обличчя через рухи. Такий метод дуже нестабільний у випадку, коли на зображенні є інші рухомі предмети або коли рухається сама камера. Інший схожий метод використовує той факт, що люди кліпає очима синхронно раз в деякий інтервал. Цей факт може допомогти знайти положення усього обличчя.

3. Методи, що базуються на моделях обличчя.

Ці методи можна використовувати навіть у випадку, коли ми маємо чорно-біле статичне зображення. Їх принцип лежить у знаходженні контрольних точок на зображенні та порівнянні їх з моделлю обличчя людини. Таким чином можна знайти положення обличчя та відстежити його міміку та рухи. Приклади таких

алгоритмів – це алгоритм узгодження орієнтації кутів [20] або алгоритм з використанням відстані Хаусдорфа [21]. Недоліки цих алгоритмів у їх складності та повільності роботи.

4. Метод Віоли-Джонса.

Справжній прорив у знаходженні обличчя на зображенні зробила публікація Віоли і Джонса у 2001-му році [22-23]. Описаний метод мав дуже високу швидкість та високе відношення вдалих знаходжень обличчя до помилкових. Хоч цей метод був представлений більше 10 років тому, на сьогодні він досі є найбільш поширеним методом знаходження обличчя на фотографіях [23], його реалізація включена у бібліотеці OpenCV. Більш детально цей алгоритм буде описаний далі.

1.2. Алгоритм Віоли-Джонса

Основні принципи, на яких базується метод, такі:

1. Використовуються зображення в інтегральному вигляді, що дозволяє обчислювати швидко необхідні об'єкти;
2. Використовуються ознаки Хаара, за допомогою яких відбувається пошук потрібного об'єкта (в даному контексті, обличчя і його риси);
3. Використовується бустінг (від англ. boost - поліпшення, посилення) для вибору найбільш підходящих ознак для шуканого об'єкта на даній частині зображення;
4. Всі ознаки надходять на вхід класифікатора, який дає результат «вірно» або «невірно»;
5. Використовуються каскади ознак для швидкого відкидання вікон, де не знайдено обличчя.

Навчання класифікаторів йде дуже повільно, але результати пошуку обличчя дуже швидкі, саме тому був обраний даний метод розпізнавання обличчя на зображенні. Віола-Джонс є одним з кращих алгоритмів по співвідношенню

показників ефективність розпізнавання / швидкість роботи. Також цей детектор має вкрай низьку ймовірність помилкового виявлення обличчя. Алгоритм навіть добре працює і розпізнає риси обличчя під невеликим кутом, приблизно до 30 градусів. При куті нахилу більше 30 градусів відсоток правильних розпізнавань різко падає. Це не дозволяє в стандартній реалізації детектувати повернене обличчя людини під довільним кутом, що в значній мірі ускладнює або робить неможливим використання алгоритму в сучасних виробничих системах з урахуванням їх зростаючих потреб. Проте для даної дипломної роботи цього методу достатньо.

Далі буде докладніше розібрано принципи, на яких базується алгоритм Віоли-Джонса. Даний метод в загальному вигляді шукає обличчя і його риси за загальним принципом скануючого вікна.

1.2.1. Принцип скануючого вікна

У загальному вигляді, завдання виявлення обличчя та рис обличчя людини на цифровому зображенні виглядає саме так:

1. Є зображення, на якому є шукані об'єкти. Воно представлено двовимірною матрицею пікселів розміром $w * h$, в якій кожен піксель має значення:
 - від 0 до 255, якщо це чорно-біле зображення;
 - від 0 до 255³, якщо це кольорове зображення (компоненти R, G, B).
2. В результаті своєї роботи, алгоритм повинен визначити положення обличчя і їх рис та позначити їх – пошук здійснюється в активній області зображення прямокутними ознаками, за допомогою яких і описується знайдене обличчя і його риси:

$$rectangle_i = \{x, y, w, h, a\} \quad (1.1)$$

де x, y – координати центру i -го прямокутника, w - ширина, h - висота, a - кут нахилу прямокутника до вертикальної осі зображення.

Іншими словами, для малюнків і фотографій використовується підхід на основі скануючого вікна (scanning window): сканується зображення вікном пошуку

(або так зване вікно сканування), а потім застосовується класифікатор до кожного положення вікна. Система навчання і вибору найбільш значущих ознак повністю автоматизована і не вимагає втручання людини, а тому цей підхід працює швидко.

Завдання пошуку і знаходження обличчя на зображенні за допомогою даного принципу часто буває лише кроком на шляху до розпізнавання характерних рис, наприклад, верифікації особи людини по знайденому зображенню або розпізнавання міміки обличчя.

1.2.2. Інтегральне представлення зображень

Для того, щоб проводити будь-які дії з даними, використовується інтегральне представлення зображень [24] в методі Віоли-Джонса. Таке представлення використовується часто і в інших методах, наприклад, в вейвлет-перетвореннях, SURF і багатьох інших алгоритмах. Інтегральне представлення дозволяє швидко розраховувати сумарну яскравість довільного прямокутника на даному зображенні, причому який би розмір прямокутника не був, час розрахунку незмінний.

Інтегральне представлення зображення - це матриця, що збігається за розмірами з вихідним зображенням. У кожному елементі її зберігається сума яскравості всіх пікселів, що знаходяться лівіше і вище даного елемента. Елементи матриці розраховуються за такою формулою:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j) \quad (1.2)$$

де $I(i, j)$ – це яскравість пікселя вхідного зображення.

Кожен елемент матриці $L[x, y]$ являє собою суму яскравості пікселів в прямокутнику від $(0, 0)$ до (x, y) , тобто значення кожного пікселя (x, y) в інтегральному вигляді дорівнює сумі значень всіх пікселів оригінального зображення лівіше і вище даного пікселя (x, y) . Розрахунок матриці займає лінійний час, пропорційно числу пікселів в зображенні, тому інтегральне зображення прораховується за один прохід.

Розрахунок матриці можливий за формулою 1.3:

$$L(x, y) = I(x, y) - L(x - 1, y - 1) + L(x, y - 1) + L(x - 1, y) \quad (1.3)$$

де позначення залишаються такими ж, як у формулі 1.2.

За такою інтегральною матрицею можна дуже швидко вирахувати суму пікселів довільного прямокутника, довільної площі.

Нехай нас цікавить сума пікселів в прямокутнику ABCD:

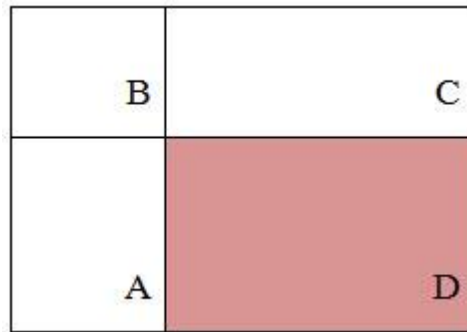


Рисунок 1.1

З малюнка зрозуміло, що суму яскравості пікселів усередині прямокутника можна виразити через суми і різниці суміжних прямокутників за такою формулою:

$$S(ABCD) = L(A_x, A_y) + L(C_x, C_y) - L(B_x, B_y) - L(D_x, D_y) \quad (1.4)$$

де A, B, C, D – це точки, а A_x , A_y і т.д. – це їх координати.

1.2.3. Ознаки Хаара

У даному контексті ознака – це відображення $f: X \Rightarrow D_f$, де D_f – безліч допустимих значень ознаки. Якщо задані ознаки f_1, \dots, f_n , то вектор ознак $x = (f_1(x), \dots, f_n(x))$ називається ознаковим описом об'єкта $x \in X$. Ознаковий опис допустимо ототожнювати з самим об'єктом. При цьому множину $X = D_{f1} * \dots * D_{fn}$ називають простором ознак [22].

Ознаки поділяються на такі типи в залежності від множини D_f :

1. бінарна ознака, $D_f = \{0,1\}$;
2. номінальна ознака: D_f – скінченна множина;
3. порядкова ознака: D_f – скінченна впорядкована множина;
4. кількісна ознака: D_f – множина дійсних чисел.

Зрозуміло, що бувають прикладні задачі з різнотипними ознаками, для їх вирішення підходять далеко не всі методи.

У стандартному методі Віоли-Джонса використовуються прямокутні ознаки, зображені на рисунку нижче, вони називаються примітивами Хаара:

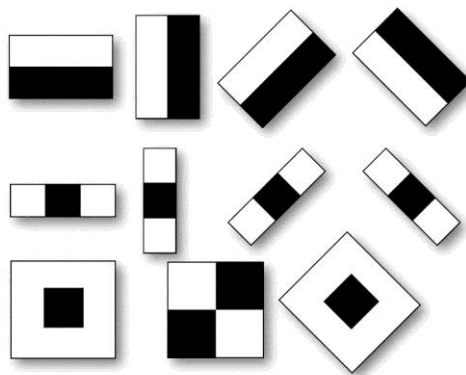


Рисунок 1.2 [28]

У розширеному методі Віоли-Джонса, що використовується в бібліотеці OpenCV використовуються додаткові ознаки:

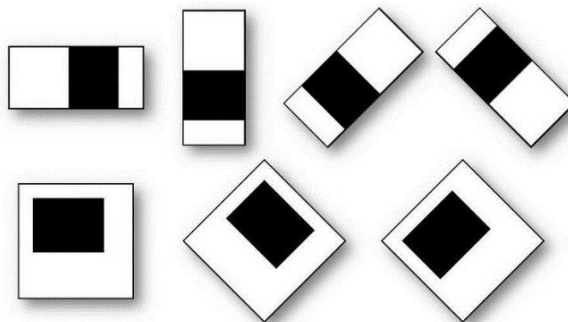


Рисунок 1.3 [29]

Обчислюваним значенням такої ознаки буде:

$$F = X - Y, \quad (1.5)$$

де X - сума значень яскравості точок, що закриваються світлою частиною ознаки, а Y - сума значень яскравості точок, що закриваються темною частиною ознаки.

Для їх обчислення використовується поняття інтегрального зображення, розглянуте вище. Ознаки Хаара дають точкове значення перепаду яскравості по осі X і Y відповідно.

1.2.4. Сканування вікна

Візуалізація скануючого вікна в програмі:

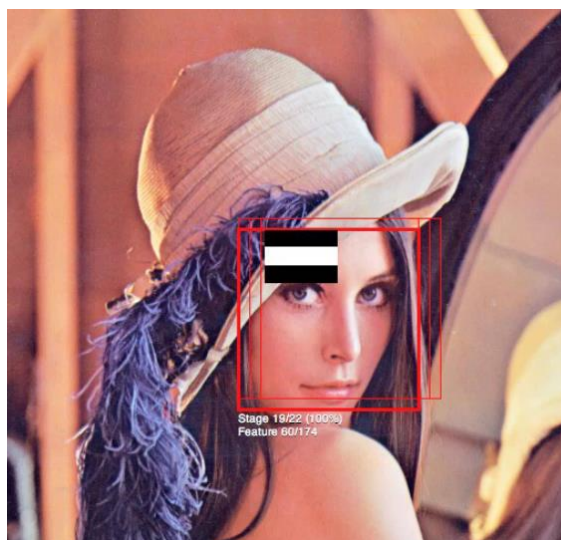


Рисунок 1.4 [30]

Алгоритм сканування вікна з ознаками виглядає так:

1. Є досліджуване зображення, вибрано вікно сканування, обрані використовувані ознаки;
2. Вікно сканування починає послідовно рухатися по зображенню з кроком в 1 клітину вікна (припустимо, розмір самого вікна є $24 * 24$ клітини);

3. При скануванні зображення в кожному вікні обчислюється приблизно 200 000 варіантів розташування ознак, за рахунок зміни масштабу ознак і їх положення в вікні сканування;
4. Сканування проводиться послідовно для різних масштабів;
5. Масштабується не саме зображення, а скануюче вікно (змінюється розмір клітини);
6. Всі знайдені ознаки потрапляють до класифікатора, який визначає, чи знайдено об'єкт у прямокутнику.

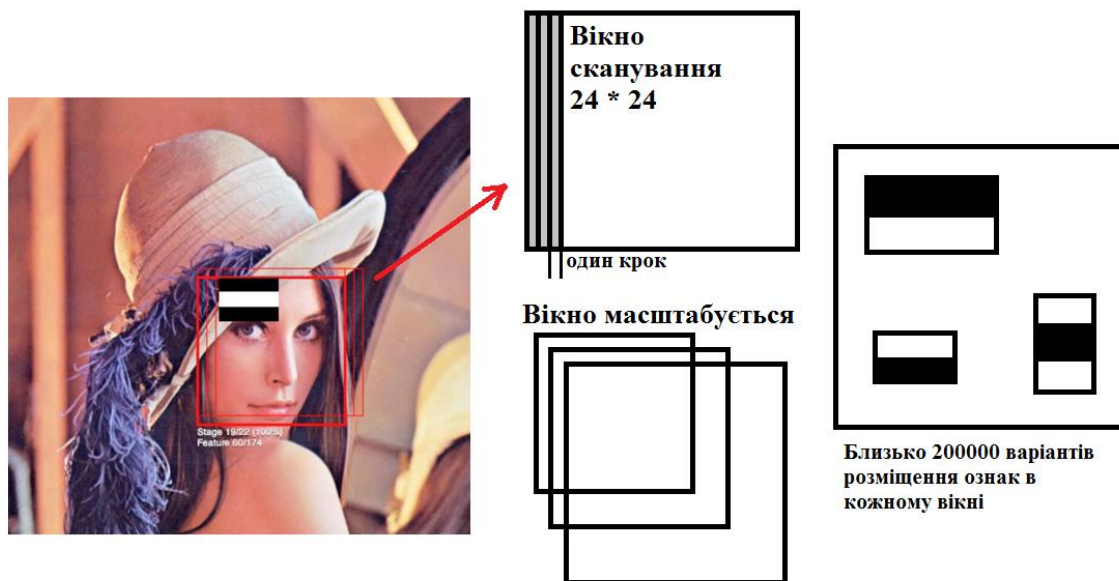


Рисунок 1.5

У процесі пошуку обчислювати всі ознаки на малопотужних персональних комп'ютерах стає неможливо. Отже, класифікатор повинен реагувати тільки на певну, обрану підмножину всіх ознак. Абсолютно логічно, що треба навчити класифікатор знаходженню обличчя за допомогою даної певної підмножини. Це можна зробити, навчаючи обчислювальну машину автоматично.

1.2.5. Навчання класифікатора в методі Віоли-Джонса

У контексті алгоритму, є множина об'єктів (зображень), розділених деяким чином на класи. Визначена скінченна множина зображень, для яких відомо, до якого класу вони відносяться (наприклад, це може бути клас «фронтальне зображення носа»). Ця множина називається навчальною вибіркою. Класова приналежність інших об'єктів невідома. Потрібно побудувати алгоритм, здатний класифікувати довільний об'єкт з початкової множини [25].

Класифікувати об'єкт - це вказати номер (або найменування класу), до якого належить даний об'єкт.

Класифікація об'єкта - номер або найменування класу, що видаються алгоритмом класифікації в результаті його застосування до даного конкретного об'єкту.

Класифікатор (classifier) - в задачах класифікації це апроксимуюча функція, що виносить рішення, до якого саме класу даний об'єкт належить.

Навчальна вибірка - скінченне число даних.

У машинному навчанні завдання класифікації відноситься до розділу навчання з учителем, коли класи роздільні та не перетинаються. Розпізнавання образів по суті і є класифікація зображень і сигналів. У випадку алгоритму Віоли-Джонса для ідентифікації та розпізнавання обличчя класифікація є двухкласовою – об'єкт може бути обличчям або не обличчям.

Постановка класифікації виглядає наступним чином:

Є X - множина, в якій зберігається опис об'єктів, Y – скінченна множина номерів, що належать класам. Між ними є залежність - відображення $Y^*: X \Rightarrow Y$. Навчальна вибірка представлена $X_m = \{(x_1, y_1), \dots, (x_m, y_m)\}$. Конструюється функція f від вектора ознак X , яка видає відповідь для будь-якого можливого спостереження X і здатна класифікувати об'єкт $x \in X$. Дане просте правило повинно добре працювати і на нових даних.

1.2.6. Застосований в алгоритмі бустінг

Для вирішення проблеми даного, такого складного навчання існує технологія бустінга.

Бустінг - комплекс методів, що сприяють підвищенню точності аналітичних моделей. Ефективна модель, яка припускає мало помилок класифікації, називається «сильною». «Слабка» ж, навпаки, не дозволяє надійно розділяти класи або давати точні прогнози, робить в роботі велику кількість помилок. Тому бустінг (від англ. Boosting - підвищення, посилення, поліпшення) означає дослівно «посилення» «слабких» моделей [26] - це процедура послідовної побудови композиції алгоритмів машинного навчання, коли кожен наступний алгоритм прагне компенсувати недоліки композиції всіх попередніх алгоритмів.

Ідея бустінга була запропонована Робертом Шапіро (Schapire) в кінці 90-х років [27], коли треба було знайти вирішення питання про те, щоб маючи безліч поганих (незначно відрізняються від випадкових) алгоритмів навчання, отримати один хороший. В основі такої ідеї лежить побудова ланцюжка (ансамблю) класифікаторів [26-27], який називається каскадом, кожен з яких (крім першого) навчається на помилках попереднього. Наприклад, один з перших алгоритмів бустінга Boost1 використовував каскад з 3-х моделей, перша з яких навчалася на всьому наборі даних, друга - на вибірці прикладів, в половині з яких перша дала правильні відповіді, а третя - на прикладах, де «відповіді» перших двох розійшлися. Таким чином, має місце послідовна обробка прикладів каскадом класифікаторів, причому так, що завдання для кожного наступного стає важче. Результат визначається шляхом простого голосування: приклад відноситься до того класу, який виданий більшістю моделей каскаду.

Бустінг є жадібний алгоритм побудови композиції алгоритмів (greedy algorithm) - це алгоритм, який на кожному кроці робить локально найкращий вибір в надії, що підсумкове рішення буде оптимальним. Бустінг над деревами вибору

вважається одним з найбільш ефективних методів з точки зору якості класифікації. У багатьох експериментах спостерігалось практично необмежене зменшення частоти помилок на незалежній тестовій вибірці при нарощуванні композиції. Більш того, якість на тестовій вибірці часто продовжувала поліпшуватися навіть після досягнення безпомилкового розпізнавання всієї навчальної вибірки [27].

У наші дні підхід посилення простих класифікаторів є популярним і, ймовірно, найбільш ефективним методом класифікації за рахунок високої швидкості та ефективності роботи і відносної простоти реалізації.

1.2.7. Каскадна модель алгоритму

Алгоритм бустінга для пошуку обличчя є таким:

1. Визначення слабких класифікаторів з прямокутними ознаками;
2. Для кожного переміщення скануючого вікна обчислюється прямокутна ознака на кожному прикладі;
3. Вибирається найбільш відповідний поріг для кожної ознаки;
4. Відбираються кращі ознаки і кращий відповідний поріг;
5. Перезважається вибірка.

Каскадна модель сильних класифікаторів - це по суті те ж дерево прийняття рішень, де кожен вузол дерева побудований таким чином, щоб детектувати майже всі образи, що нас цікавлять і відхиляти регіони, які не є образами. Крім цього, вузли дерева розміщені таким чином, що чим ближче вузол знаходиться до кореня дерева, тим з меншої кількості примітивів він складається і тим самим вимагає меншого часу на прийняття рішення. Даний вид каскадної моделі добре підходить для обробки зображень, на яких загальна кількість детектуємих образів мала.

У цьому випадку метод може швидше прийняти рішення про те, що даний регіон не містить образ, і перейти до наступного. Приклад каскадної моделі сильних класифікаторів:

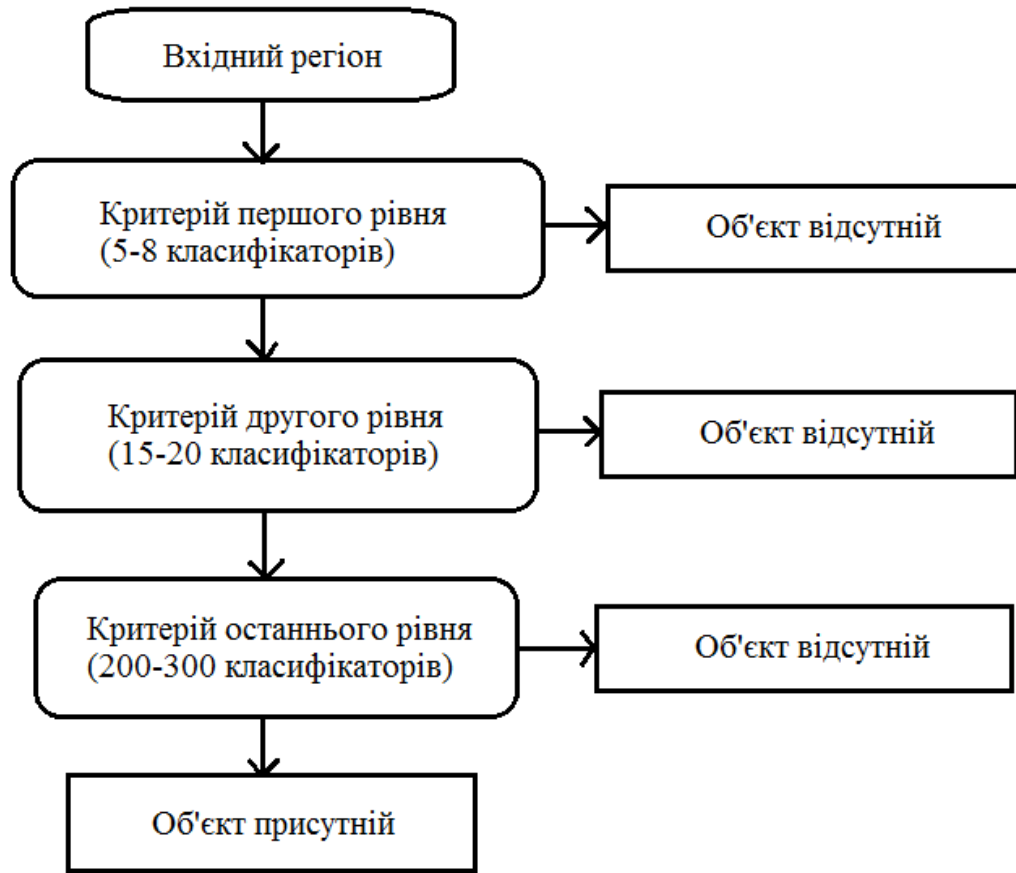


Рисунок 1.6

Складність навчання таких каскадів дорівнює $O(xyz)$, де застосовується x етапів, у прикладів і z ознак.

Далі, каскад застосовується до зображення:

1. Робота з «простими» класифікаторами - при цьому відкидається частина «негативних» вікон;
2. Позитивне значення першого класифікатора запускає другий, більш пристосований і так далі;
3. Негативне значення класифікатора на будь-якому етапі призводить до негайного переходу до наступного скануючого вікна, старе вікно відкидається;
4. Ланцюжок класифікаторів стає складнішим, тому помилок стає набагато менше.

Для тренування такого каскаду будуть потрібні наступні дії:

1. Задаються значення рівня помилок для кожного етапу (попередньо їх треба кількісно переглянути при застосуванні до зображення з навчального набору) - вони називаються detection та false positive rates - треба щоб рівень detection був високий, а рівень false positive rates низький;
2. Додаються ознаки доти, поки параметри обчислюваного етапу не досягнуть поставленого рівня, тут можливі такі допоміжні етапи, як:
 - а. Тестування додаткового маленького тренувального набору;
 - б. Зниження порогу AdaBoost з метою знайти більше об'єктів, але в зв'язку з цим можлива більша кількість неточних визначень об'єктів;
3. Якщо false positive rates залишається високим, то необхідно додати етап або шар;
4. Помилкові виявлення в поточному етапі використовуються як негативні вже на наступному шарі або етапі.

1.3. Обчислення показників зміни кольору обличчя

На цьому етапі ми маємо положення обличчя людини як координати квадрату, що оточує його. Також ми маємо дані про колір кожного пікселя усередині квадрату.

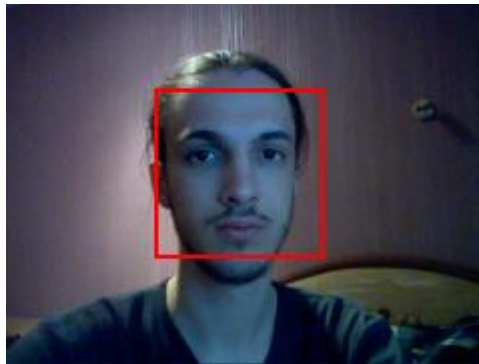


Рисунок 1.7

Для трекінгу необхідних нам даних потрібно знайти середній колір шкіри на кожному кадрі. Для цього є два основних підходи:

1. Обмежити квадрат обличчя меншою областю, що буде містити у собі переважно лише зображення обличчя або шкіри.
2. Використовувати маску, що надає пріоритет пікселям, чий колір менш відрізняється від кольору шкіри.

1.3.1. Обмеження області обличчя

Цей спосіб є найпростішим та використовується у більшості методів по відстеженню ЧСС через відеозображення [15-19].

Пропорції обличчя людини у середньому приблизно дорівнюють 6:10, тому оптимальним варіантом обмеження зони інтересу буде використання середніх 60% області обличчя.



Рисунок 1.8

Таким чином, як бачимо, в зону інтересу потрапляє лише обличчя користувача, а не фон зображення.

1.3.2. Використання маски обличчя

Навіть на обмеженій області зображення присутні очі, волосся, іноді окуляри. Ці елементи зображення не дають корисної інформації про зміну кольору шкіри

обличчя та заважають якісному аналізу зображення. Деякі минулі праці використовували фільтрацію пікселів зображення за допомогою знаходження їх різниці між кольором шкіри [17-19].

Для реалізації цього варіанту необхідно знайти колір шкіри та порівняти його з кольором кожного пікселя. При наявності зображення обличчя кольором шкіри буде той колір, який в регіоні інтересу зустрічається якнайчастіше.

Для знаходження кольору шкіри обчислюється частота зустрічання кожного кольору в області обличчя. Два найближчих до ідеалу способи знайти колір, що зустрічається найчастіше – це мода та медіана.

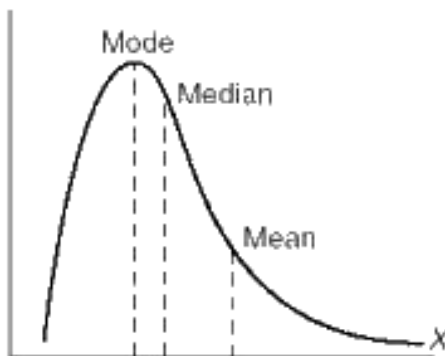


Рисунок 1.9 – Мода, медіана та середнє [33]

Після знаходження кольору шкіри необхідно побудувати маску обличчя. Чим ближче значення кольору пікселя до кольору шкіри, тим світліше повинен бути піксель маски.

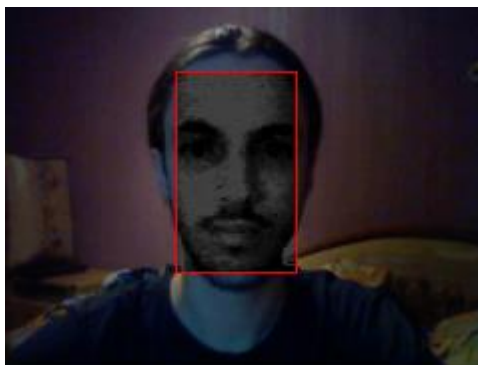


Рисунок 1.10 – Маска, побудована за допомогою моди

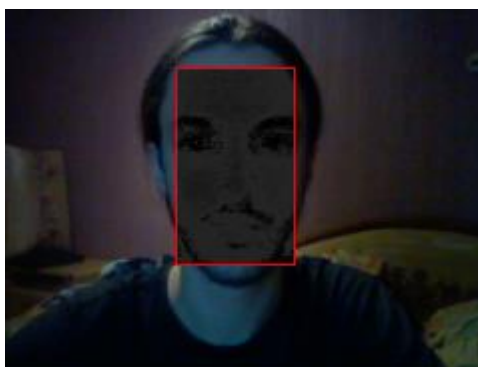


Рисунок 1.11 – Маска, побудована за допомогою медіани

Чим яскравіше піксель маски, тим більше він буде впливати на середнє значення кольору пікселя при розрахунку середнього значення для області обличчя.

1.4. Нормалізація сигналу

При послідовному запису середнього значення кольору обличчя ми отримаємо нестабільні значення, що можуть сильно змінюватись через зміну освітлення або рухи голови.



Рисунок 1.12 – Сигнал зміни кольору обличчя

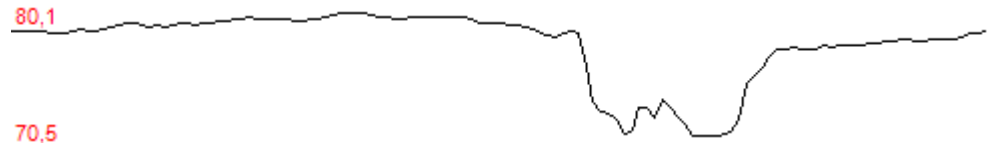


Рисунок 1.13 – Сигнал зміни кольору обличчя

Для того, щоб краще відслідкувати невеликі коливання сигналу, що відповідають серцебиттю, необхідно нормалізувати сигнал.

Для цього використовується проста формула:

$$y_i = \frac{x_i - \mu}{\sigma} \quad (1.6)$$

де x_i – це оригінальний сигнал, μ – це середнє значення сигналу на найближчій ділянці, σ – це стандартне відхилення на тій же ділянці.

Такая формула дає чудові результати для нормалізації каналу та значно знижує чутливість до змін освітлення та рухів.



Рисунок 1.14 – Нормалізований сигнал



Рисунок 1.15 – Нормалізований сигнал

Теперь неозброєним оком видно коливання графіку в такт серцебиттю користувача.

1.5. Ресемплінг сигналу

Сигнал, що отримує програма з відеокамери, не отримується через статичні проміжки часу. Камера може іноді швидше, а іноді повільніше працювати в залежності від завантаженості комп'ютера та умов освітлення.

Для того, щоб можна було провести наступний крок аналізу – ШПФ – необхідно перетворити сигнал на дискретний сигнал, елементи якого у часі знаходяться на рівних відстанях. Для цього встановлюється статичний період запису даних та використовується апроксимація кожної точки графіку сигналу до найближчих двох точок у часі, отриманих з камери.

1.6. Перетворення Фур'є

Коли ми отримуємо готовий до аналізу сигнал, щоб знайти частоту, з якою він змінюється, необхідно провести перетворення Фур'є цього сигналу. Для цього використовується широковідомий алгоритм Швидкого перетворення Фур'є, який доступний на будь-якій мові програмування.

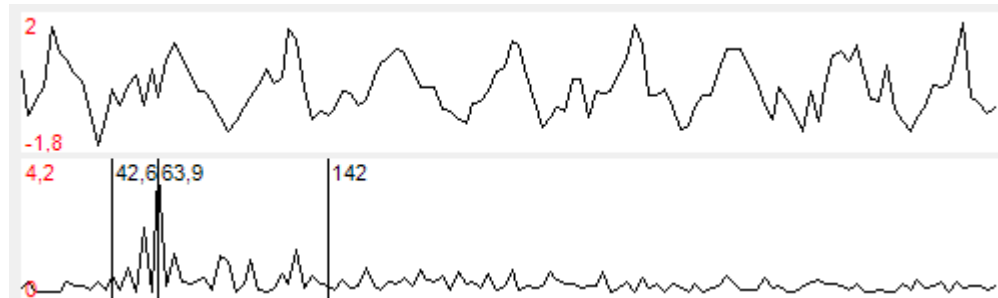


Рисунок 1.16 – Перетворення Фур'є для сигналі кольору шкіри

Як видно с рис. 1.16, нормалізований сигнал подається на вхід алгоритму ШПФ, а на виході можна отримати спектр, що описує, які частоти переважають у вхідному сигналі.

На рис. 1.16 показано дві вертикальні лінії, що відповідають максимальному и мінімальному здоровому пульсу людини, а одна лінія між ними, що відповідає максимуму графіку перетворення Фур'є. Знаючи частоту, з якою був отриманий вхідний сигнал, можна отримати частоту кожної точки на графіку Фур'є. В даному випадку максимум показує, що пульс користувача дорівнює 64 удари на хвилину.

1.7. Сліпий поділ джерел

Хоча у більшості випадків, коли обличчя користувача не рухається, достатньо лише описаних вище методів, розповсюдженням є використання алгоритмів сліпого поділу джерел для отримання сигналу, що описує серцебиття людини.

Сліпий поділ джерел – це відокремлення набору сигналів з деяких джерел з набору змішаних сигналів, при відсутності будь-якої додаткової інформації про процес змішування або природу джерел. Ця проблема в основному дуже недетермінована, проте корисні рішення можуть бути знайдені в великій кількості випадків.

Аналіз незалежних компонентів (англ. independent component analysis (ICA)) – це частковий випадок сліпого поділу джерел, при якому субкомпоненти, змішані в сигнал, що отримує датчик, є не-Гауссівського типу та статистично незалежні один від одного.

Розповсюджений приклад проблеми, яку вирішує АНК – це проблема «вечірки з коктейлями». Ця проблема полягає в відділенні записаного голосу диктора від звуку шумної зали. Маючи два записи одної та тієї ж промови диктора, можна за допомогою АНК отримати запис його слів без шуму та запис шуму без його слів окремо.

Для того, щоб пояснити принцип роботи алгоритму, наведемо приклад.

Нехай ми маємо два вхідні сигнали, що є змішаними сигналами деяких джерел.

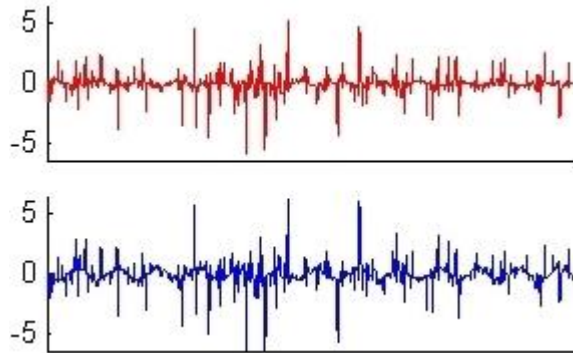


Рисунок 1.17 [31]

Ці сигнали змінюються у часі та приймають значення, близькі до 0, найчастіше.

Якщо побудувати графік спільного розподілу для цих двох сигналів, отримаємо зображене на рис. 1.19.

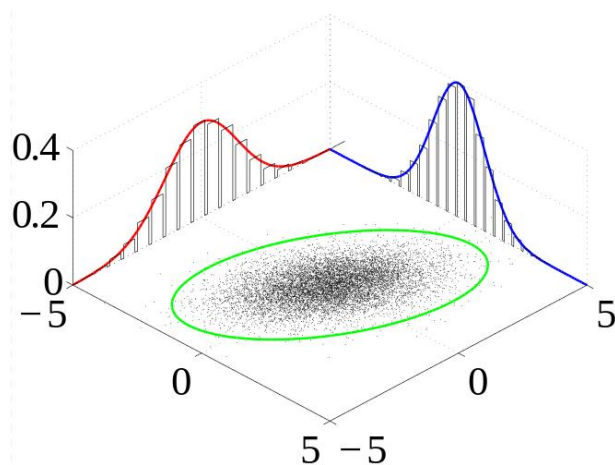


Рисунок 1.18 – Побудова графіку спільного розподілу [32]

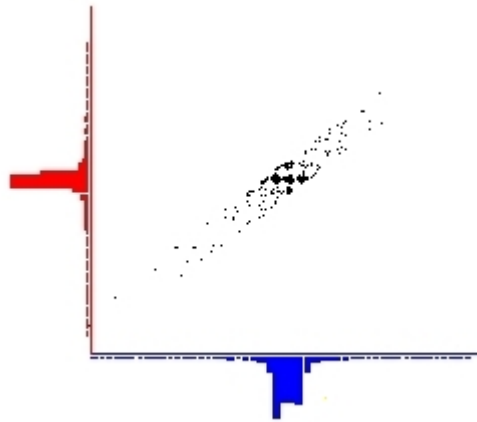


Рисунок 1.19 [31]

Цей графік показує, що точки графіків збираються у центральних точках. Оскільки гістограми показують гауссівський розподіл, то можна зробити висновок, що сигнали змішані та мають спільну компоненту.

Перший крок роботи алгоритму АНК це «відбілювання» даних. Цей крок видаляє всілякі кореляції у вхідних даних.

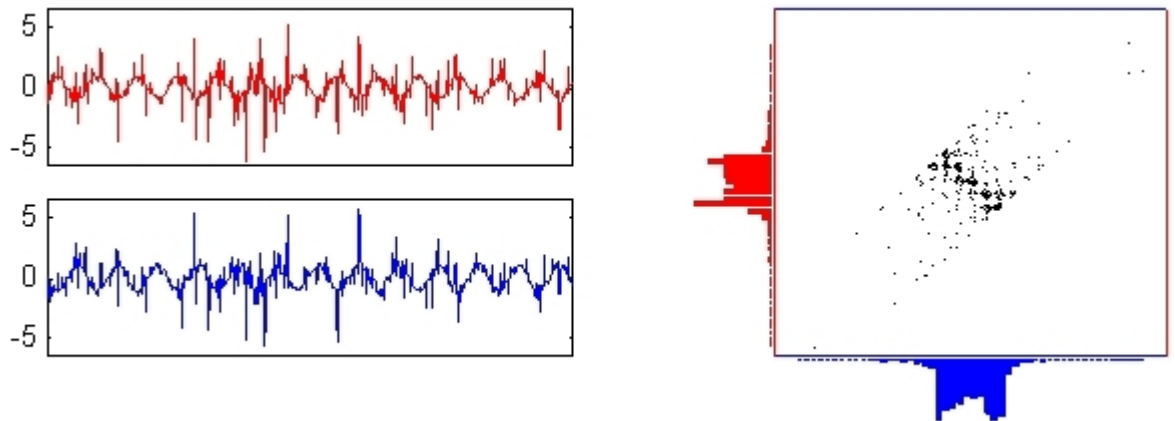


Рисунок 1.20 – Data whitening in ICA [31]

Після цього етапу, оригінальні сигнали можна буде знайти, якщо ортогонально трансформувати сигнали. Наявно це відображається як поворот

графіку спільного розподілу. Це впливає з факту, що лінійна комбінація незалежних змінних завжди більш гауссівська, ніж оригінальне джерело.

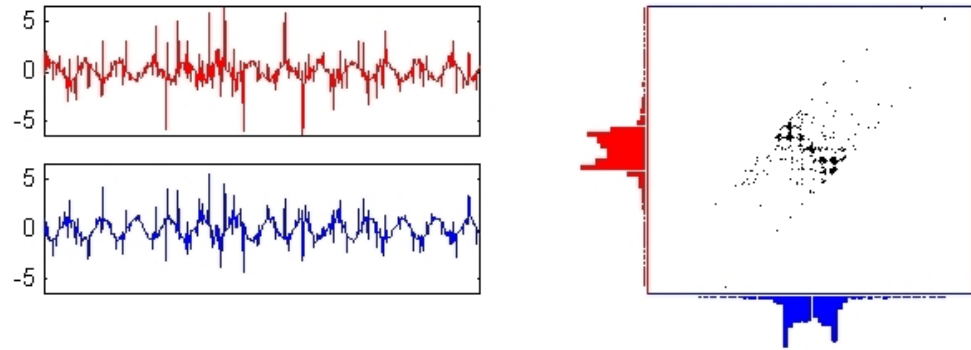


Рисунок 1.21 – Крок 1 [31]

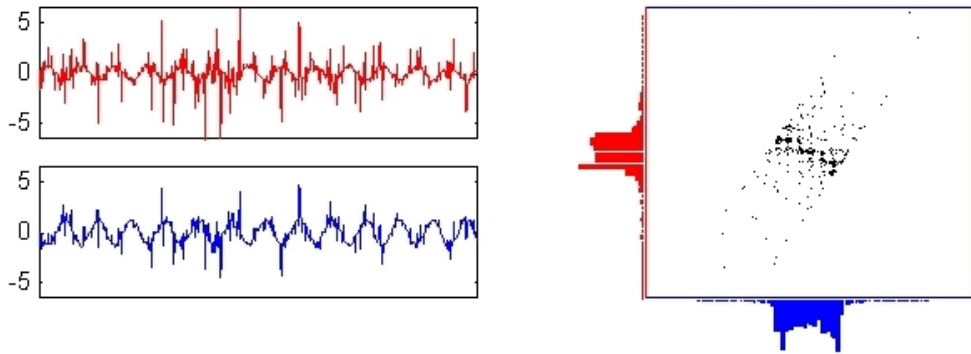


Рисунок 1.22 – Крок 2 [31]

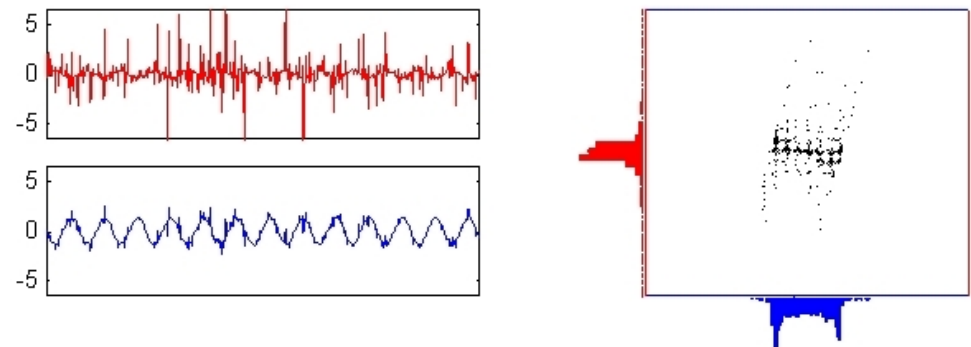


Рисунок 1.23 – Крок 3 [31]

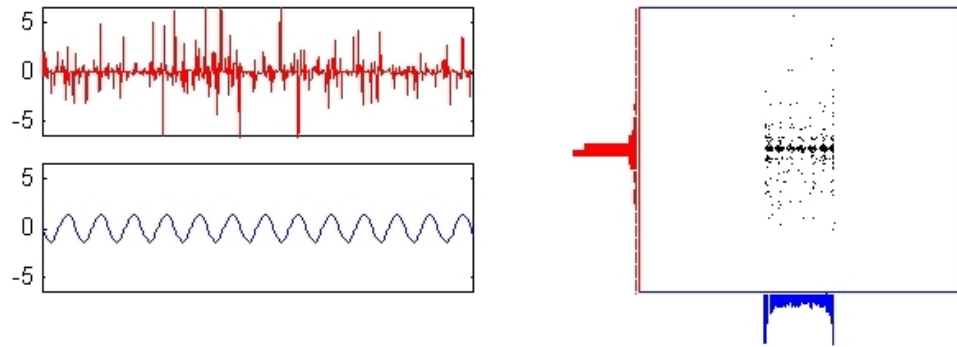


Рисунок 1.24 – Крок 4 [31]

Роблячи перетворення та повертаючи графік спільного розподілу, алгоритм АНК знаходить індивідуальні компоненти зі змішаних сигналів тоді, коли на кроці розподіл точок є найменш гауссівським.

Для кожного із вихідних сигналів тепер можна провести ШПФ та знайти сигнал, який має найбільш виражену частоту коливань.

Існує декілька алгоритмів, що вирішують цю проблему. Одні з найпоширеніших з них – це FastICA, JADE та RobustICA [18]. При вимірюванні серцебиття через зображення обличчя найшвидше та найточніше працює алгоритм FastICA [18], що взагалі є одною з найвідоміших реалізацій даного алгоритму. Тому саме цю реалізацію алгоритму ми будемо використовувати в цій роботі.

Цей алгоритм можна використовувати декількома способами для знаходження сигналу, що відповідає зміні кров'яного тиску в шкірі обличчя.

По-перше, можна використати три компоненти кольору – червоний, зелений та синій, як суміші сигналів джерел та використати АНК на них [15-19].

По-друге, можна використати більш ніж одну камеру при дослідженні та в якості сигналів використати всі канали кольорів.

По-третє, можна використати більш ніж одну камеру, проте лише один канал кольору для кожної камери.

По-четверте, можна подати на вхід алгоритму три канали кольору кожної камери, а потім результат роботи алгоритму для кожної веб-камери використати як вхідний сигнал ще однієї ітерації алгоритму.

1.8. Висновки

Було описано основні алгоритми, що використовуються при дослідженні серцебиття людини за допомогою зображення обличчя на відео.

Було описано недоліки та переваги деяких алгоритмів серед подібних їм, наочно показано роботу найскладніших алгоритмів з наявних, коротко описано послідовність використання алгоритмів під час аналізу зображення.

Ця інформація є хорошою базою для розробки програмного продукту, що буде реалізовувати всі з наведених методів вирішення проблеми та покаже на практиці їх переваги і недоліки.

2. ПРОГРАМНА РЕАЛІЗАЦІЯ

2.1. Вимоги до програмної реалізації

Програма, що реалізована в рамках цієї дипломної роботи, має містити у собі функціонал для опробовування, порівняння та використання усіх способів аналізу відеозображення обличчя людини, що були описані у першому розділі цієї роботи.

Програма має працювати достатньо швидко, щоб можна було порівнювати хоча б два методи аналізу одночасно без сильного впливу на швидкість роботи програми.

Програма має показати, що хоча б один з запропонованих методів аналізу зображення дає задовільні результати, та показати, які з методів дають результат більш надійний, ніж інші.

2.2. Зчитування та запис сигналів з веб-камери

Для роботи з веб-камерою та знаходженням положення обличчя на камері найкраще підходить бібліотека OpenCV. Вона має вбудовані методи підключення наявних у комп'ютері веб-камер та реалізацію алгоритму Віюлі-Джонса. Ця бібліотека є безкоштовною та може вільно використовуватися. Для мови програмування C# існує вряпер для OpenCV під назвою EmguCV. Версія EmguCV 2.4 має весь необхідний функціонал, багату документацію та працює на архітектурах x86 та x64. Саме тому в даному проекті буде використовуватися EmguCV.

2.2.1. Клас CameraClass

Клас CameraClass використовується для роботи з веб-камерою, знаходження положення обличчя, запису інформації щодо середнього кольору обличчя та виводу зображення на екран.

При створенні екземпляру цього класу підключається за допомогою EmguCV конкретна наявна у системі веб-камера, та починається аналіз надходжуючих з неї кадрів. Також ініціалізується об'єкт HaarCascade, що відповідає за локалізацію обличчя.

При появі кожного нового кадру робота алгоритма є наступною:

- 1) Знайти положення обличчя за допомогою HaarCascade.
- 2) Зменшити знайдену область обличчя до 60% по ширині.
- 3) Розрахувати дві маски для обличчя:
 - a. Маска по медіані кольору обличчя
 - b. Маска по моді кольору обличчя
- 4) Записати середні кольори обличчя для трьох випадків:
 - a. Середній колір обличчя без маски
 - b. Середній колір за маскою по медіані
 - c. Середній колір за маскою по моді
- 5) Вивести зображення на головну форму програми.

Цей клас записує лише деяку кількість останніх записів сигналу, що детермінується параметром конструктору.

Клас CameraClass є основним класом, що уособлює считування сигналу. Він запускається завжди у новому потоці, тому швидкість роботи на багатоядерних комп'ютерах буде досить високою.

2.2.2. Клас Processor

Цей клас є абстрактним класом, що описує основний функціонал, потрібний для аналізу даних, що записує клас CameraClass.

У цьому класі реалізовані такі функції:

- 1) Нормалізація
- 2) Ресемплінг
- 3) Аналіз незалежних компонент

Для реалізації аналізу незалежних компонент було використано безкоштовну бібліотеку для C# Accord.Statistics, тому що це єдина бібліотека на C#, що реалізує цей алгоритм.

2.2.3. Підкласи класу Processor

Абстрактний клас Processor є лише базою для різноманіття видів аналізу зібраних з веб-камер даних. Можна створити будь-який породжений від нього клас, який може вільно використовувати усі дані, наявні у класах CameraClass.

1) FirstCamSimpleProcessor

Це клас, що використовує завжди лише одну камеру, нормалізує та ресемплює її дані, та використовує зелений канал кольору для перетворення Фур'є.

2) FirstCamICAProcessor

Це клас, що використовує лише одну камеру, нормалізує та ресемплює її дані, та використовує АНК, надаючи три канали кольору як вхідні дані для АНК. Три результуючі компоненти АНК проходять через перетворення Фур'є та з них вибирається компонента з найбільш вираженою коливальною природою.

3) ICAProcessor

Це клас, що використовує усі підключені камери, нормалізує та ресемплює їх дані, та використовує АНК, надаючи три канали кольору кожної з камер як вхідні дані для АНК. Результуючі компоненти АНК проходять через перетворення Фур'є та з них вибирається компонента з найбільш вираженою коливальною природою.

4) MultiCamGreenChICAProcessor

Це клас, що використовує усі підключені камери, нормалізує та ресемплює їх дані, та використовує АНК, надаючи лише зелений канал кольору кожної з камер як вхідні дані для АНК. Результуючі компоненти АНК проходять через перетворення Фур'є та з них вибирається компонента з найбільш вираженою коливальною природою.

2.2.4. Клас FFTObject

Цей клас містить у собі вхідні дані для ШПФ та результат розрахунку ШПФ. Окрім того, під час обчислення ШПФ, цей клас записує наступні дані для подальшого використання та аналізу:

- 1) Частоту дискретизації
- 2) Максимум спектру Фур'є
- 3) Індекс максимуму спектру в масиві
- 4) Частота в Герцах максимуму спектру
- 5) «Фактор шуму»

Останній показник є ключовим у всіх методах аналізу, що включають АНК.

«Фактор шуму» (noiseFactor) обчислюється за такою формулою:

$$z = \frac{\sum x_i}{n * A} \quad (1.7)$$

де z – це фактор шуму, x – це елемент результату перетворення Фур'є, n – це кількість елементів в масиві, A – це максимум амплітуди спектру Фур'є.

Цей фактор, по суті, є відношенням середньої амплітуди шумів до амплітуди сигналу на спектрі Фур'є. Чим менше це число, тим точніше і більш вираженим є деяка частота коливального руху у вхідному сигналі. Саме цей фактор є визначним для вирішення, чи є деякий метод більш точним за інший.

2.3. Висновки

Було описано структуру основних класів, що реалізують необхідний програмний інструментарій для аналізу та порівняння найчастіше використовуваних методів для аналізу серцевого пульсу людини по відеозображенню.

3. ПРАКТИЧНЕ ПОРІВНЯННЯ МЕТОДІВ АНАЛІЗУ

3.1. Використання масок для фільтрації зображення обличчя

В даному тесті було використано та порівняно три підходи до аналізу:

- 1) Аналіз Фур'є без АНК для зеленого каналу без маски
- 2) З маскою по моді кольору обличчя
- 3) З маскою по медіані кольору обличчя

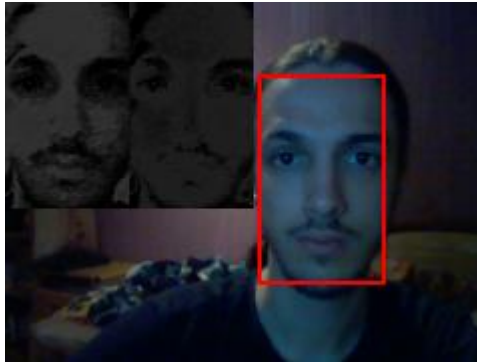


Рисунок 3.1

На рисунку 3.1 зображено дві маски, що відповідають описаним у пункту 1.3.2 алгоритмам. Також в даному тесті записуються дані середнього кольору обличчя без маски взагалі.

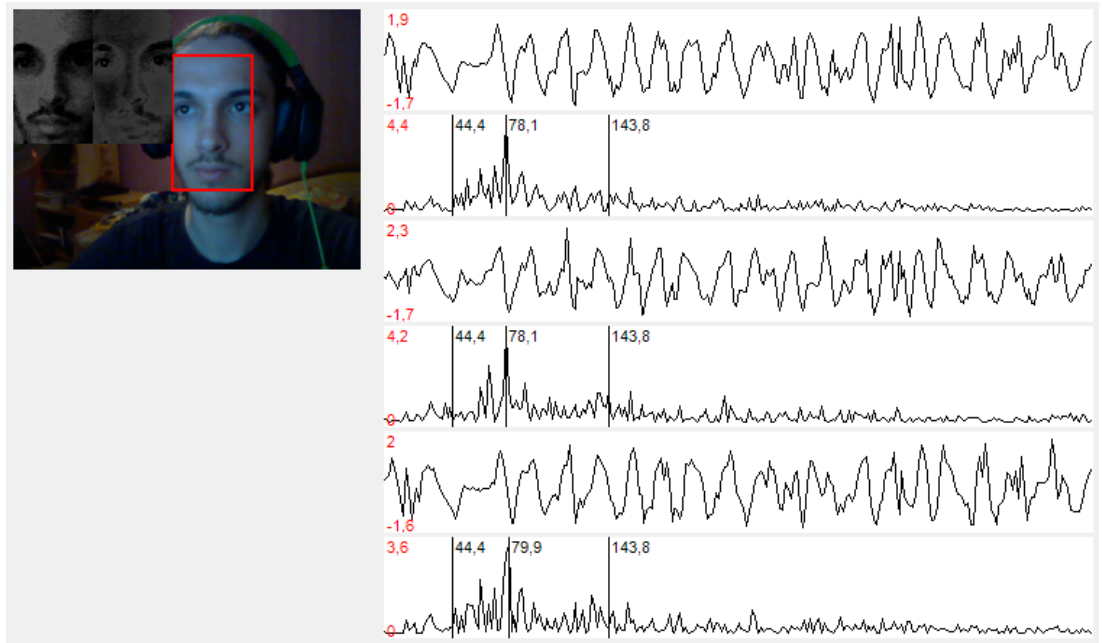


Рисунок 3.2 – Порівняння роботи трьох методів

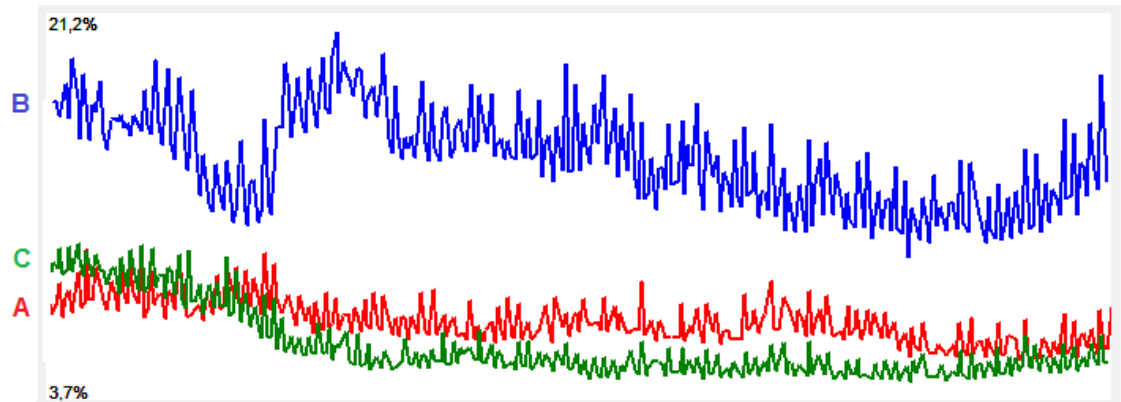


Рисунок 3.3 – Вихідні дані порівняння фактору шуму

На рисунку 3.3 графік А – це фактор шуму варіанту без маски, В – варіанту з маскою по медіані, С – варіант з маскою по моді.

Як бачимо з графіку, найкращий результат показує варіант з використанням маски по моді, проте варіант без маски майже не відрізняється від нього за точністю. Варіант з маскою по медіані показує погані результати.

Можливою причиною того, що використання маски не дає набагато кращих результатів, може бути той факт, що для побудови маски необхідно знайти колір шкіри. Оскільки колір шкіри розраховується динамічно, то вигляд маски змінюється на кожному кадрі, що призводить до постійних невеликих змін вигляду маски, що додає артефактів та шумів до сигналу.

Висновок: використання маски не покращує суттєво результат аналізу.

3.2. Використання АНК для трьох каналів кольору

В даному тесті було використано та порівняно два підходи до аналізу:

- 1) АНК для трьох каналів кольору однієї камери
- 2) Аналіз лише зеленого каналу однієї камери

На рисунку 3.4 зображено порівняння результату роботи цих двох методів. Перший – аналіз зеленого каналу, другий – АНК.

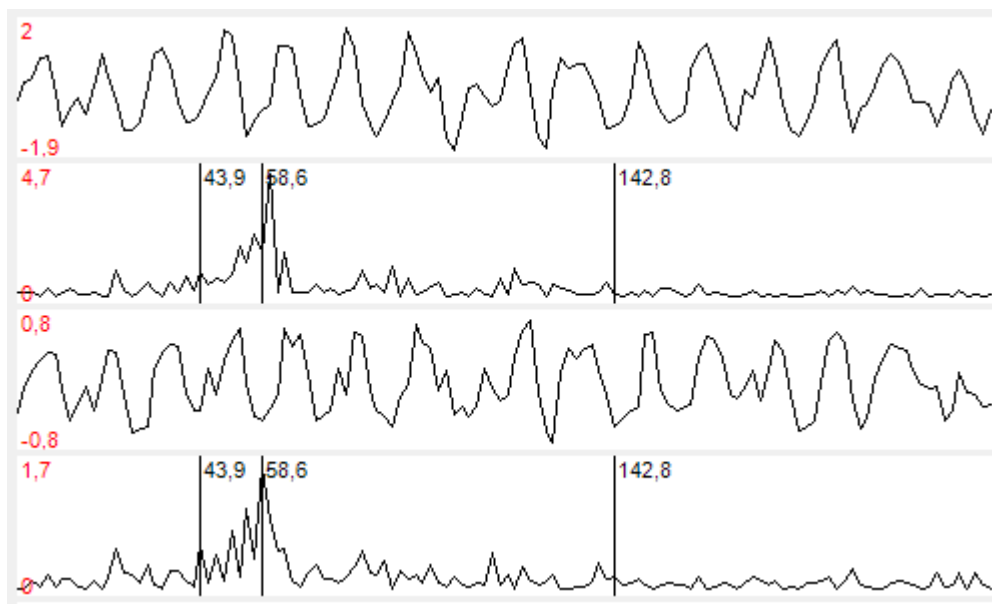


Рисунок 3.4 – Фур'є зеленого каналу на АНК

Як бачимо, результати роботи в реальному часі не дуже сильно відрізняються. АНК робить свою роботу та знаходить необхідний незалежний компонент.

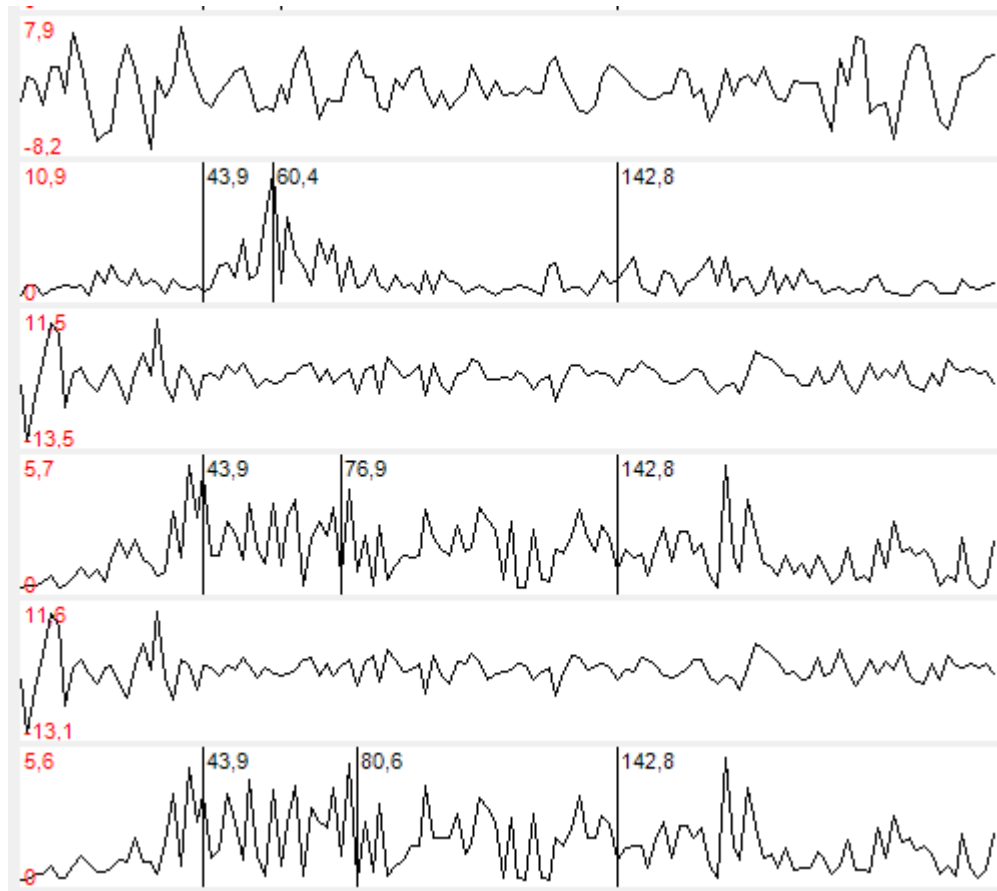


Рисунок 3.5 – Порівняння трьох компонент на виході АНК

На рисунку 3.5 можна побачити три перетворення Фур'є для трьох виходів алгоритму АНК. Вони відсортовані так, що вище розміщений компонент з більш вираженою коливальною природою. Як бачимо, алгоритм працює як потрібно, знаходить необхідний незалежний компонент, який в спектрі Фур'є показує пульс користувача.

Проте, якщо поглянути на графік порівняння фактору шуму:

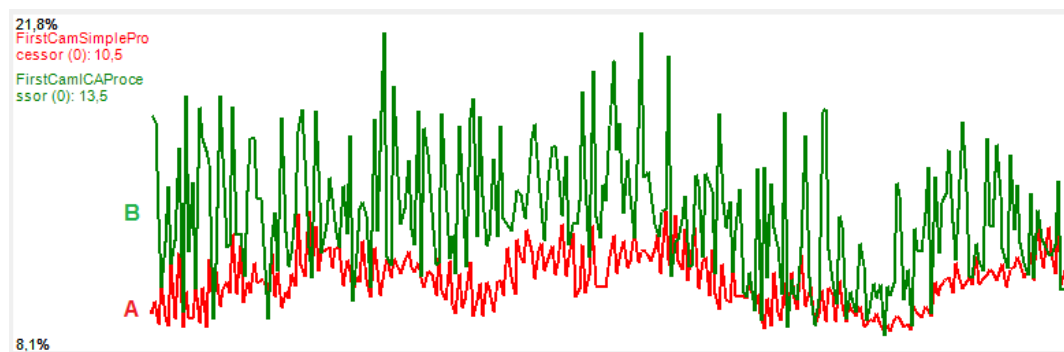


Рисунок 3.6 – Фактори шуму

То можна побачити, що АНК (лінія В) має набагато менш стабільну роботу, аніж аналіз зеленого каналу світла (лінія А).

Це пов'язано з тим, що АНК має справу з двома гаусовськими сигналами шуму окрім одного сигналу джерела, і іноді вони викликають сбої в роботі алгоритму.

3.3. Використання АНК для двох камер і всіх каналів кольору

В даному тесті було використано та порівняно два підходи до аналізу:

- 1) АНК для трьох каналів кольору кожної з двох камер
- 2) Аналіз лише зеленого каналу однієї камери

На рисунку 3.7 показано зображення, що передається з двох камер. На рисунку 3.8 зображено порівняння фактору шуму вищеописаних методів.

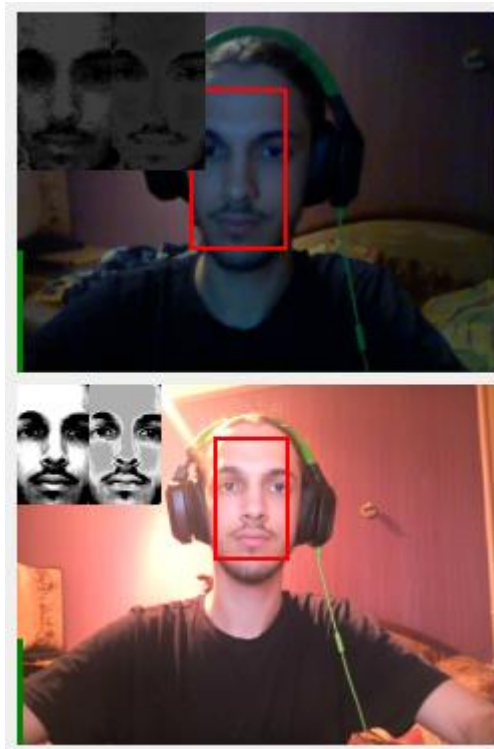


Рисунок 3.7 – Одночасне використання двох камер

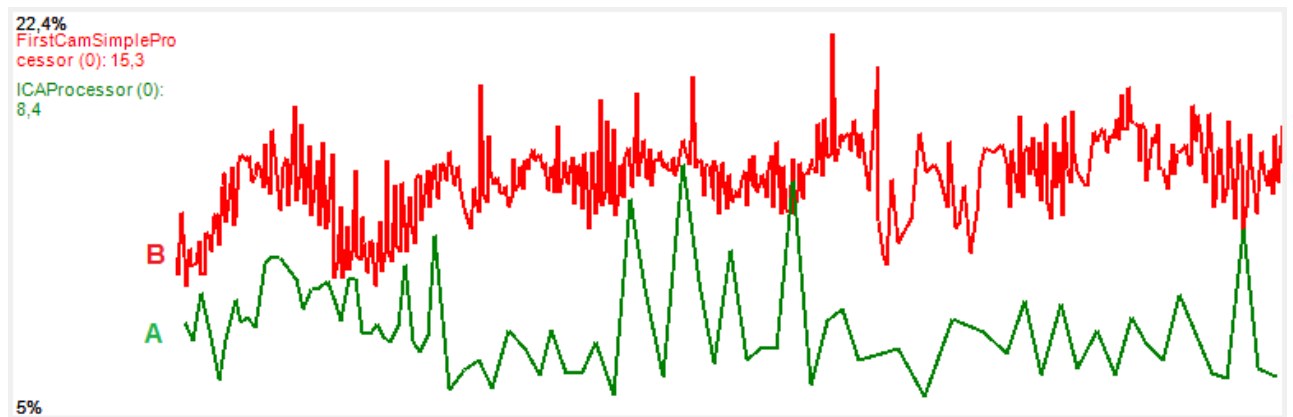


Рисунок 3.8 – Фактор шуму

Як бачимо, у випадку використання двох незалежних джерел, АНК (лінія А) виграє у аналізу Фур'є зеленого каналу (лінія В). Проте АНК все ще має проблеми з нестабільністю – графік час від часу має різкі зміни точності, де потрібний незалежний компонент не був знайдений.

3.4. Використання АНК для двох камер і одного каналу кольору

В даному тесті було використано та порівняно два підходи до аналізу:

- 1) АНК для зеленого каналу кольору кожної з двох камер
- 2) Аналіз лише зеленого каналу однієї камери

На рисунку 3.9 зображено порівняння фактору шуму цих методів.

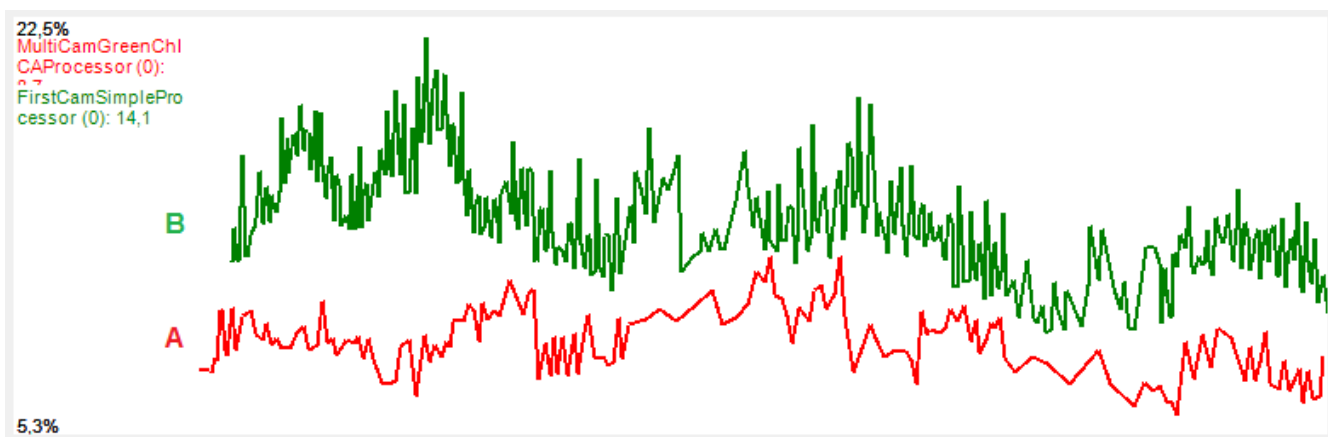


Рисунок 3.9 – Фактор шуму

Як можна побачити з останнього результату, АНК в цьому випадку (лінія А) дає набагато стабільніший результат та ще більше віддаляється від аналізу однією камерою (лінія В) в плані точності розрахунку.

Причина цього в тому, що включення трьох каналів кольору в АНК є поганим вибором, адже два з трьох незалежних компонентів у результаті повинні бути гаусовського типу. Також три вхідні компоненти з кожної камери містять шуми однакового типу, адже вони були зняті на одну камеру. Найкращі результати АНК дає тоді, коли використовуються два повністю різних джерела – тобто дві камери.

3.5. Висновки

- 1) Використання маскуванню для фільтрації регіону інтересу за кольором пікселів не набагато покращує якість алгоритмів, а частіше не покращує зовсім.
- 2) Використання АНК дає найкращі результати лише тоді, коли присутні справжні два різних прилади спостереження, тобто дві камери.
- 3) Найкращі два підходи з проаналізованих:
 - a. Більш дешевий та простий варіант:

Запис усередненого кольору шкіри в прямокутнику з пропорціями 6:10 в області обличчя; Аналіз Фур'є зміни зеленої компоненти цього кольору у часі;
 - b. Більш дорогий, проте більш точний варіант:

Запис усередненого кольору шкіри в прямокутнику з пропорціями 6:10 в області обличчя на двох камерах одночасно; Аналіз незалежних компонент для зелених каналів кольору з декількох камер, аналіз Фур'є для отриманих результатів АНК.

4. ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту, призначеного для моніторингу частоти серцевих скорочень людини за допомогою вебкамери. Інтерфейс користувача був розроблений за допомогою мови програмування C# у середовищі розробки Microsoft Visual Studio 2012. Інтерфейс користувача створений за допомогою технології Windows Forms, OpenCV, FFT.

Програмний продукт призначено для використання на персональних комп'ютерах під управлінням операційної системи Windows.

Нижче наведено аналіз різних варіантів реалізації продукту з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

4.1. Аналіз ринку

Програмний продукт, який розробляється в даній дипломній роботі, не є присутнім на ринку в теперішній час. Програм, що реалізують моніторинг серцевих скорочень по відео, зараз ще немає в широкому використанні, немає стабільних реалізацій цих алгоритмів, що були б достатньо розвинутими для комерційного випуску. По даній тематиці є декілька наукових робіт, пояснюючих принципи вирішення задачі, є декілька експериментальних реалізацій подібних програмних продуктів, проте вони не присутні на ринку, а є лише науковими роботами.

Програму, що обчислює серцеві скорочення, може зустріти попит у двох галузях: медицина та розваги. Подібну програму можуть використовувати лікарі або лікарні для діагностики хворих на відстані. А також цю програму можуть використовувати гравці в комп'ютерні ігри для моніторингу стану свого здоров'я та нервової системи, наприклад, для відстеження напружливих моментів у відеоіграх, які впливають на частоту серцевих скорочень.

Друга галузь є достатньо перспективною для комерційної реалізації, можна використовувати рекламу або платне розповсюдження програмного продукту для монетизації створеної реалізації.

4.2. Постановка задачі

У роботі застосовується метод ФВА для проведення техніко-економічного огляду програмного продукту.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

– програмний продукт повинен функціонувати на персональних комп'ютерах зі стандартним набором компонент;

- забезпечувати зручність і простоту взаємодії з користувачем;
- достатньо якісно виконувати свою функцію
- передбачати мінімальні витрати на впровадження програмного продукту.

4.2.1. Обґрунтування функцій програмного продукту

Можна виділити наступні основні функції ПП:

F_1 – вибір C# реалізації бібліотеки OpenCV;

F_2 – вибір способу знаходження регіону інтересу;

F_3 – вибір алгоритму трекінгу регіону інтересу.

F_4 – використання Independent Component Analysis

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція F_1 :

- а) EmguCV;
- б) OpenCVSharp;

Функція F_2 :

- а) Маска по кольору обличчя;
- б) Статичний прямокутний регіон;

Функція F_3 :

- а) Трекінг області зображення;
- б) Повторне знаходження обличчя алгоритмом Хаара в кожному кадрі;

Функція F_4 :

- а) Використання FastICA та трьох каналів кольору;
- б) Відмова від ICA та використання нормалізованого зеленого каналу.

4.2.2. Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 4.1).

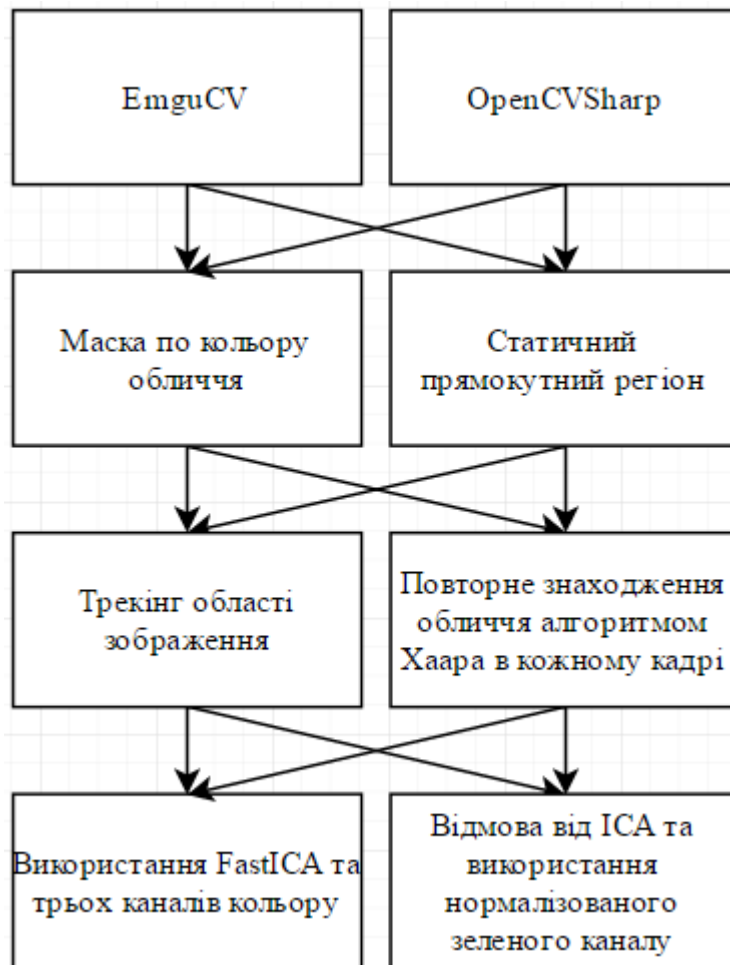


Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів ПП.

Таблиця 4.1 – Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
<i>F1</i>	<i>A</i>	Краща документація та якість коду	Лише версія OpenCV 2.4 працює на всіх машинах
	<i>B</i>	Остання версія OpenCV	Погана документація та реалізація
<i>F2</i>	<i>A</i>	Потенційно краща якість отриманих даних	Неоднозначність реалізації та параметрів алгоритму
	<i>B</i>	Простота та однозначність реалізації	Потенційно гірша якість отриманих даних
<i>F3</i>	<i>A</i>	Відсутність затримки при зміні положення обличчя	Складність реалізації, можливість помилок трекінгу
	<i>B</i>	Простота реалізації, постійна вневненість в точності відображення обличчя	Затримка при зміні положення обличчя
<i>F4</i>	<i>A</i>	Більш точне обчислення	Більші навантаження на процесор, повільніша робота
	<i>B</i>	Швидка робота	Менш точні обчислення

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція *F1*:

Оскільки версія 2.4 OpenCV повністю відповідає вимогам програми та має функціонал знаходження обличчя за допомогою каскадів Хаара, то варіант *B* можна відкинути.

Функція F3:

Під час користування програмою постійні рухи обличчя людини не є частою проблемою, обличчя людини повинно бути статичним або рухатися повільно. Тому варіант А є зайвим та не несе користі.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

1. F1A – F2A – F3B – F4A
2. F1A – F2B – F3B – F4A
3. F1A – F2A – F3B – F4B
4. F1A – F2B – F3B – F4B

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3. Обґрунтування системи параметрів ПП

4.3.1. Опис параметрів

На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- $X1$ – час обробки одного кадру з камери;
- $X2$ – кількість шумів у вихідному сигналі;
- $X3$ – чутливість до зміни освітлення;
- $X4$ – об'єм програмного коду

4.3.2. Кількісна оцінка параметрів

Гірші, середні і кращі значення параметрів вибираються на основі вимог до програми.

Таблиця 4.2 – Основні параметри ПП

Опис параметру	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Час обробки одного кадру з камери	X1	мс	200	110	70
Кількість шумів у вихідному сигналі	X2	%	60	30	15
Чутливість до зміни освітлення	X3	%	50	20	0
Об'єм програмного коду	X4	строки	1500	800	500

За даними таблиці 4.2 будуються графічні характеристики параметрів – рис. 2 – рис. 5.

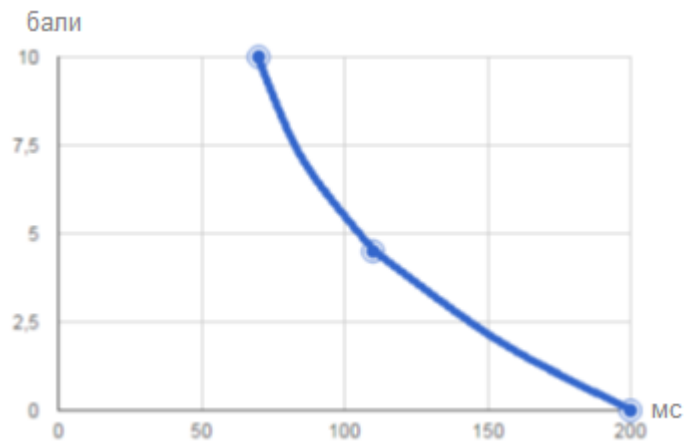


Рисунок 4.2 – X1, Час обробки одного кадру з камери

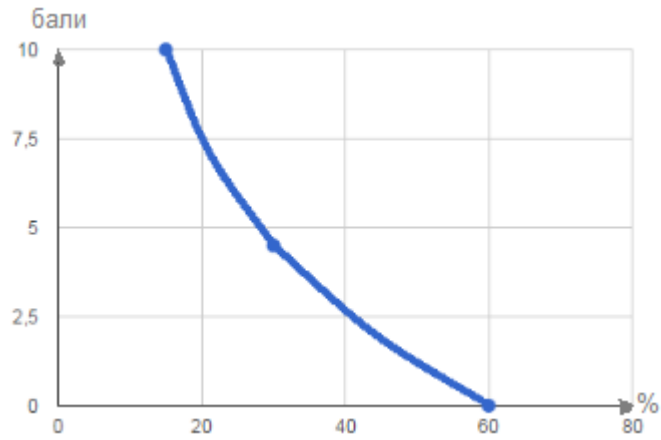


Рисунок 4.3 – X2, Кількість шумів у вихідному сигналі

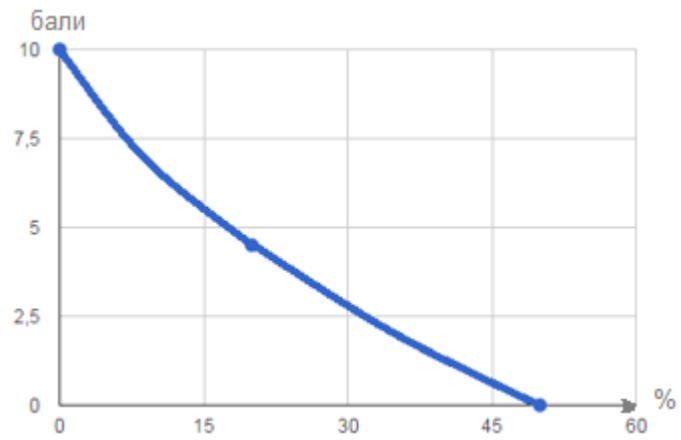


Рисунок 4.4 – X3, Чутливість до зміни освітлення

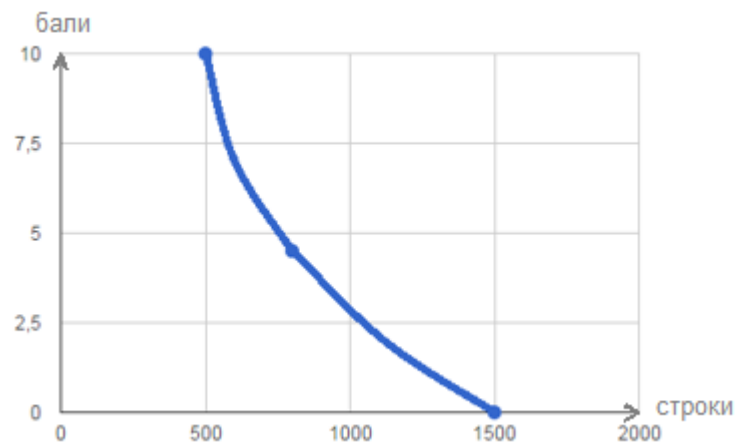


Рисунок 4.5 – X4, Об'єм програмного коду

4.3.3. Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Для перевірки степені достовірності експертних оцінок, визначимо коефіцієнт конкордації:

$$W = 12 * 177 / (49 * (64 - 4)) = 2124 / 2940 = 0.72$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.3 – Результати ранжування параметрів

Позначення	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Квадрат відхилення Δ_i^2
			1	2	3	4	5	6	7			
X1	Час обробки одного кадру з камери	мс	2	1	3	2	2	2	1	13	-4,5	20,25
X2	Кількість шумів у вихідному сигналі	%	3	3	4	3	3	4	4	24	6,5	42,25
X3	Чутливість до зміни освітлення	%	4	4	2	4	4	3	3	24	6,5	42,25
X4	Об'єм програмного коду	строки	1	2	1	1	1	1	2	9	-8,5	72,25
	Разом									70	0	177

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	=	<	<	=	<	<	<	<	0.5
X1 і X3	<	<	=	<	<	=	<	<	0.5
X1 і X4	>	=	>	>	>	>	<	>	1.5
X2 і X3	<	<	>	=	=	>	=	=	1.0
X2 і X4	>	>	>	>	>	>	>	>	1.5
X3 і X4	>	>	>	>	>	>	>	>	1.5

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметри	Параметри				Перша ітер.		Друга ітер.	
	X1	X2	X3	X4	bi	Kbi	bi	Kbi
X1	1.0	0.5	0.5	1.5	3.5	0.22	12,25	0.21
X2	1.5	1.0	1.0	1.5	5	0.31	19	0.32
X3	1.5	1.0	1.0	1.5	5	0.31	19	0.32
X4	0.5	0.5	0.5	1.0	2.5	0.16	9,25	0.16
Всього:					16	1	59,5	1

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 1%, тому більшої кількості ітерацій не потрібно.

4.4. Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Коефіцієнт технічного рівня якості для кожного варіанта реалізації ПП розраховується в таблиці 4.6.

За даними з таблиці 4.6 визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 1,85 + 3,15 + 2,59 + 4,41 = 11,16$$

$$K_{K2} = 3,33 + 3,15 + 2,59 + 4,41 = \mathbf{13,44}$$

$$K_{K3} = 1,85 + 3,15 + 3,33 + 3,15 = 5,37$$

$$K_{K4} = 3,33 + 3,15 + 3,33 + 3,15 = 12,74$$

Як видно з розрахунків, кращим є другий варіант, для якого коефіцієнт технічного рівня має найбільше значення.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметр	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F2(X)	А	X1	105	5	0,21	1,05
F4(X)	А	X1	95	7	0,21	1,47
F2(X)	Б	X1	75	9	0,21	1,89
F4(X)	Б	X1	75	9	0,21	1,89
F2(X)	А	X2	30	5	0,32	1,6
F4(X)	А	X2	20	7	0,32	2,24
F2(X)	Б	X2	30	5	0,32	1,6
F4(X)	Б	X2	30	5	0,32	1,6
F2(X)	А	X3	40	3	0,32	0,96
F4(X)	А	X3	10	8	0,32	2,56
F2(X)	Б	X3	20	5	0,32	1,6
F4(X)	Б	X3	20	5	0,32	1,6
F2(X)	А	X4	800	5	0,16	0,8
F4(X)	А	X4	1000	3	0,16	0,48
F2(X)	Б	X4	500	10	0,16	1,6
F4(X)	Б	X4	700	6	0,16	0,96

4.5. Економічний аналіз розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Усі варіанти розробки ПП мають загальні властивості:

- 1 макет вхідної інформації
- 1 різновид вихідної інформації
- Новизна групи Б
- Складність рівня 2
- Одноманітність вхідної і вихідної інформації (5.2, 5.4)
- Використовується змінний тип інформації (ЗІ)
- Можливе використання стандартних модулів.

Проте, вибір варіантів розробки К1 та К2 змінює складність до **рівня 1**.

Проведемо розрахунок норм часу на розробку та програмування для двох варіантів, що об'єднують по два інших: К1-К2 та К3-К4 (оскільки їх характеристика однакова).

Загальна трудомісткість обчислюється як

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (4.1)$$

де T_P – трудомісткість розробки ПП; K_{Π} – поправочний коефіцієнт; $K_{СК}$ – коефіцієнт на складність вхідної інформації; K_M – коефіцієнт рівня мови програмування; $K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм; $K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення.

Для першого варіанту, виходячи із норм часу для завдань розрахункового характеру степеню новизни Б та групи складності алгоритму 1, трудомісткість дорівнює: $T_P = 64$ людино-днів. Поправочний коефіцієнт, який враховує вид інформації для першого завдання: $K_{\Pi} = 2,02$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх варіантів рівний: $K_{СК} = 1$. Оскільки при розробці всіх варіантів використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.7$. Тоді, за формулою 5.1, загальна трудомісткість програмування першого завдання дорівнює:

$$T_{12} = 64 \cdot 2,02 \cdot 0.7 = 90,5 \text{ людино-днів} = 724 \text{ людино-годин.}$$

Для варіантів К3-К4 складність алгоритму становить рівень 2, тому трудомісткість змінюється:

$$T_{34} = 27 \cdot 1,8 \cdot 0,7 = 34,02 \text{ людино-днів} = 272,16 \text{ людино-годин.}$$

В розробці бере участь один програміст з окладом 15000 грн. Визначимо зарплату за годину за формулою:

$$C_q = \frac{M}{T_m \cdot t} \text{ грн.,}$$

де M – місячний оклад працівників; T_m – кількість робочих днів тиждень; t – кількість робочих годин в день.

$$C_q = \frac{15000}{21 \cdot 8} = 89,3 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою

$$C_{3П} = C_q \cdot T_i \cdot K_d,$$

де C_q – величина погодинної оплати праці програміста; T_i – трудомісткість відповідного завдання; K_d – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$\text{К1-К2: } C_{3П} = 89,3 \cdot 724 \cdot 1,2 = 77583,84 \text{ грн.}$$

$$\text{К3-К4: } C_{3П} = 89,3 \cdot 272,16 \cdot 1,2 = 29164,67 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$\text{К1-К2: } C_{ВІД} = C_{3П} \cdot 0,22 = 77583,84 \cdot 0,22 = 17068,44 \text{ грн.}$$

$$\text{К3-К4: } C_{ВІД} = C_{3П} \cdot 0,22 = 29164,67 \cdot 0,22 = 6416,23 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. (C_M)

Так як один комп'ютер обслуговує одного програміста з окладом 15000 грн., з коефіцієнтом зайнятості 0,8 то для однієї машини отримаємо:

$$C_{Г} = 12 \cdot M \cdot K_3 = 12 \cdot 15000 \cdot 0,8 = 144000 \text{ грн в рік}$$

З урахуванням додаткової заробітної плати:

$$C_{3П} = C_{Г} \cdot (1 + K_3) = 144000 \cdot (1 + 0,2) = 172800 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$C_{\text{ВІД}} = C_{\text{ЗП}} \cdot 0.22 = 172800 \cdot 0.22 = 38016 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 10000 грн.

$$C_A = K_{\text{ТМ}} \cdot K_A \cdot C_{\text{ПР}} = 1.15 \cdot 0.25 \cdot 10000 = 2875 \text{ грн.},$$

де $K_{\text{ТМ}}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача; K_A – річна норма амортизації; $C_{\text{ПР}}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{\text{ТМ}} \cdot C_{\text{ПР}} \cdot K_P = 1.15 \cdot 10000 \cdot 0.05 = 575 \text{ грн.},$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{\text{ЕФ}} = (D_K - D_B - D_C - D_P) \cdot t_z \cdot K_B = (365 - 104 - 8 - 16) \cdot 8 \cdot 0.9 = 1706.4 \text{ годин},$$

де D_K – календарна кількість днів у році; D_B , D_C – відповідно кількість вихідних та святкових днів; D_P – кількість днів планових ремонтів устаткування; t – кількість робочих годин в день; K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot C_{\text{ЕН}} = 1706.4 \cdot 0.4 \cdot 0.8 \cdot 2.0218 = 1104 \text{ грн.},$$

де N_C – середньо-споживча потужність комп'ютера (400Вт); K_3 – коефіцієнт зайнятості приладу (0,8); $C_{\text{ЕН}}$ – тариф за 1 кВт*год електроенергії.

Тариф за 1кВт*год для компаній, що потребують менш, ніж 27,5 кВ, дорівнює 2,0218 грн/кВтгод з НДС.

Накладні витрати розраховуємо за формулою:

$$C_H = C_{\text{ПР}} \cdot 0.67 = 15000 \cdot 0.67 = 10050 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H$$

$$C_{\text{ЕКС}} = 172800 + 38016 + 2875 + 575 + 1104 + 10050 = 225420 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 225420 / 1706,4 = 132,1 \text{ грн/час.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_{\text{М}} = C_{\text{М-Г}} \cdot T$$

$$K1-K2: C_{\text{М}} = 132,1 \cdot 724 = 95640,40 \text{ грн.};$$

$$K3-K4: C_{\text{М}} = 132,1 \cdot 272,16 = 35952,34 \text{ грн.};$$

Накладні витрати складають 67% від заробітної плати:

$$C_{\text{Н}} = C_{\text{ЗП}} \cdot 0,67$$

$$K1-K2: C_{\text{Н}} = 77583,84 \cdot 0,67 = 51981,17 \text{ грн.};$$

$$K3-K4: C_{\text{Н}} = 29164,67 \cdot 0,67 = 19540,33 \text{ грн.};$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{\text{ПП}} = C_{\text{ЗП}} + C_{\text{Від}} + C_{\text{М}} + C_{\text{Н}}$$

$$K1-K2: C_{\text{ПП}} = 77583,84 + 17068,44 + 95640,40 + 51981,17 = 242273,85 \text{ грн.};$$

$$K3-K4: C_{\text{ПП}} = 29164,67 + 6416,23 + 35952,34 + 19540,33 = 91073,57 \text{ грн.};$$

4.6. Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{ТЕР}j} = K_{\text{К}j} / C_{\text{Ф}j},$$

$$K_{K1} = 1,85 + 3,15 + 2,59 + 4,41 = 11,16$$

$$K_{K2} = 3,33 + 3,15 + 2,59 + 4,41 = \mathbf{13,44}$$

$$K_{K3} = 1,85 + 3,15 + 3,33 + 3,15 = 5,37$$

$$K_{K4} = 3,33 + 3,15 + 3,33 + 3,15 = 12,74$$

$$K_{\text{TEP1}} = 11,16 / 242273,85 = 0,46 \cdot 10^{-4};$$

$$K_{\text{TEP2}} = 13,44 / 242273,85 = 0,55 \cdot 10^{-4};$$

$$K_{\text{TEP3}} = 5,37 / 91073,57 = 0,59 \cdot 10^{-4};$$

$$K_{\text{TEP4}} = 12,74 / 91073,57 = \mathbf{1,4 \cdot 10^{-4}};$$

Як бачимо, найбільш економічно ефективним є 4-й варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{TEP4}} = 1,4 \cdot 10^{-4}$.

4.7. Висновки

В даному розділі проведено повний функціонально-вартісний аналіз ПП, який було розроблено в рамках дипломного проекту. Процес аналізу можна умовно розділити на дві частини.

В першій з них проведено дослідження ПП з технічної точки зору: було визначено основні функції ПП та сформовано множину варіантів їх реалізації; на основі обчислених значень параметрів, а також експертних оцінок їх важливості було обчислено коефіцієнт технічного рівня, який і дав змогу визначити оптимальну з технічної точки зору альтернативу реалізації функцій ПП.

Другу частину ФВА присвячено вибору найбільш вигідного варіанту х економічної точки зору. Порівняння запропонованих варіантів реалізації в рамках даної частини виконувалось за коефіцієнтом ефективності, для обчислення якого були обчислені такі допоміжні параметри, як трудомісткість, витрати на заробітну плату, накладні витрати тощо.

Оскільки проект не є комерційним та не націлений на економічний прибуток, аналіз з економічної точки зору не береться до уваги при розробці дипломного проекту. Для розробки було вибрано варіант, що найбільше відповідав результатам ФВА, варіант 2.

Цей варіант розробки дасть змогу створити найоптимальніший варіант реалізації проекту.

ВИСНОВКИ

В результаті виконання роботи було розроблено програмний продукт, що реалізує основні методи аналізу відеозображення обличчя для вимірювання частоти серцевих скорочень людини.

Було реалізовано такі алгоритми:

1. Перетворення Фур'є нормалізованого зеленого каналу середнього кольору обличчя.
2. Використання маски для знаходження регіону інтересу на зображенні обличчя.
3. Аналіз незалежних компонентів, де як джерела використані три канали середнього кольору обличчя.
4. Аналіз незалежних компонентів, де як джерела використані три канали середнього кольору обличчя з декількох підключених камер.
5. Аналіз незалежних компонентів, де як джерела використані зелені канали середнього кольору обличчя з декількох підключених камер.

В результаті порівняння точності вимірювання пульсу цими алгоритмами, було досягнуто таких висновків:

1. Використання маски для знаходження регіону інтересу може покращити результат вимірювання, проте незначно.
2. Аналіз незалежних компонентів, що використовує як три джерела три канали кольору обличчя, є менш точним методом, ніж аналіз лише зеленого каналу. Цей метод використовується у багатьох минулих публікаціях, проте є недостатньо надійним. Причинами цього спостереження може бути той факт, що шуми, що наявні в усіх трьох каналах, мають спільне джерело, адже були зареєстровані однією і тією ж камерою. Зміни в освітленні, якість зображення з камери – ці шуми, як було спостережено експериментально, мають нормальний розподіл на

графіку, що спричинює погану роботу алгоритму аналізу незалежних компонент, адже цей алгоритм спирається на максимізацію негаусівської природи джерел сигналів. Однією із вимог до використання цього алгоритму є негаусівська природа оригінального сигналу.

3. При використанні двох відеокамер та аналізу незалежних компонент точність вимірювання значно покращується, адже тепер джерела для алгоритму АНК є дійсно незалежними та спостережують за одним явищем. При використанні лише зеленого каналу двох чи більше відеокамер для аналізу незалежних компонент, можна отримати чітко виражений сигнал, що описує частоту серцебиття людини.

Проведене дослідження показує високий потенціал використання двох чи більше камер для досягнення високої точності вимірювання фізіологічних показників людини.

Подальші дослідження у цій галузі можуть привести до надійних та дешевих неінвазивних методів моніторингу серцевого пульсу людини у госпіталях або домашніх умовах.

Програма, розроблена у цій дипломній роботі, може слугувати для домашнього використання для періодичної перевірки серцевого пульсу, проте для впровадження схожої системи на комерційній основі в медицині необхідні подальші дослідження, адже медична сфера вимагає гранично точних вимірювань при будь-якому аналізі стану людини.

ПЕРЕЛІК ПОСИЛАНЬ

1. S. Cook, High heart rate: a cardiovascular risk factor? / S. Cook, M. Togni, M. C. Schaub, P. Wenaweser, O. M. Hess // *European Heart Journal*, 27(20), 2387–2393 (2006)
2. M. Garbey, Contact-free measurement of cardiac pulse based on the analysis / M. Garbey, N. Sun, A. Merla, I. Pavlidis // *IEEE Transactions on Biomedical Engineering*, 54(8), 1418–1426 (2007)
3. F. P. Wieringa, Contactless multiple wavelength photoplethysmographic imaging: a first step toward “SpO2 camera” technology / F. P. Wieringa, F. Mastik, A. F. van der Steen // *Ann. Biomed. Eng.* 33(8), 1034–1041 (2005)
4. K. Humphreys, Noncontact simultaneous dual wavelength photoplethysmography: a further step toward noncontact pulse oximetry / K. Humphreys, T. Ward, C. Markham // *Rev. Sci. Instrum.* 78(4), 044304 (2007)
5. J. Allen, Photoplethysmography and its application in clinical physiological measurement / J. Allen // *Physiol. Meas.* 28(3), R1–R39 (2007)
6. C. Takano, Heart rate measurement based on a time-lapse image / C. Takano, Y. Ohta // *Med. Eng. Phys.* 29(8), 853-857 (2007)
7. W. Verkruysse, Remote plethysmographic imaging using ambient light / W. Verkruysse, L. O. Svaasand, J. S. Nelson // *Opt. Express* 16(26), 21434–21445 (2008)
8. S. Rhee, Artifact-resistant power-efficient design of finger-ring plethysmographic sensors / S. Rhee, B. H. Yang, H. H. Asada // *IEEE Trans. Biomed. Eng.* 48(7), 795–805 (2001)
9. M. Z. Poh, Motion-tolerant magnetic earring sensor and wireless earpiece for wearable photoplethysmography / M. Z. Poh, N. C. Swenson, R. Picard // *IEEE Trans Inf Technol Biomed* (Epub 2010 Feb)

- 10.C. J. James, Independent component analysis for biomedical signals / C. J. James, C. W. Hesse // *Physiol. Meas.* 26(1), R15–R39 (2005)
- 11.M. P. Chawla, Artifacts and noise removal in electrocardiograms using independent component analysis / M. P. Chawla, H. K. Verma, V. Kumar // *Int. J. Cardiol.* 129(2), 278–281 (2008)
- 12.T. P. Jung, Removing electroencephalographic artifacts by blind source separation / T. P. Jung, S. Makeig, C. Humphries, T. W. Lee, M. J. McKeown, V. Iragui, T. J. Sejnowski // *Psychophysiology* 37(2), 163–178 (2000)
- 13.J.-F. Cardoso, Multidimensional independent component analysis / J.-F. Cardoso // *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (IEEE, 1998)*, pp. 1941–1944
- 14.B. S. Kim, Motion artifact reduction in photoplethysmography using independent component analysis / B. S. Kim, S. K. Yoo // *IEEE Trans. Biomed. Eng.* 53(3), pp. 566–568 (2006)
- 15.M. Z. Poh, Noncontact, automated cardiac pulse measurements using video imaging and blind source separation / M. Z. Poh, Ming-Zher, Daniel J. McDuff, Rosalind W. Picard // *Optics Express* 18 pp. 10762-10774 (2010)
- 16.Antony Lam, Robust Heart Rate Measurement from Video Using Select Random Patches / Antony Lam and Yoshinori Kuno // *IEEE International Conference on Computer Vision (ICCV)* pp. 3640-3648 (2015)
- 17.Magdalena Lewandowska, Measuring Pulse Rate with a Webcam – a Non-contact Method for Evaluating Cardiac Activity / Magdalena Lewandowska, Jacek Rumiński, Tomasz Kocejko, Jędrzej Nowak // *Proceedings of the Federated Conference on Computer Science and Information Systems* pp. 405–410 (2011)
- 18.Eirini Christinaki, Comparison of Blind Source Separation Algorithms for Optical Heart Rate Monitoring / Eirini Christinaki, Giorgos Giannakakis, Franco Chiarugi, Matthew Pediaditis, Galatea Iatraki, Dimitris Manousos, Kostas Marias, Manolis Tsiknakis // *Wireless Mobile Communication and Healthcare* pp. 339-342 (2014)

19. Dragos Datcu, Noncontact automatic heart rate analysis in visible spectrum by specific face regions / Dragos Datcu, Marina Cidota, Stephan Lukosch, Leon Rothkrantz // CompSysTech '13 Proceedings of the 14th International Conference on Computer Systems and Technologies pp. 120-127 (2013)
20. Bernhard Fröba, Real-Time Face Detection Using Edge-Oriented Matching / Bernhard Fröba, Christian Küblbeck // AVBPA '01 Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication pp. 78-83 (2001)
21. Oliver Jesorsky, Robust Face Detection Using the Hausdorff Distance / Oliver Jesorsky, Klaus J. Kirchberg, and Robert W. Frischholz // Springer, Lecture Notes in Computer Science, LNCS-2091, pp. 90-95 (2001)
22. P. Viola, Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola and M.J. Jones // Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001
23. P. Viola, Robust real-time face detection / P. Viola and M.J. Jones // International Journal of Computer Vision, vol. 57, no. 2, pp.137–154 (2004)
24. Р. Гонсалес, Цифровая обработка изображений / Р.Гонсалес, Р.Вудс // ISBN 5-94836-028-8, изд-во: Техносфера, Москва, 2005. – 1072 с.
25. Местецкий Л. М., Математические методы распознавания образов / Местецкий Л. М. // МГУ, ВМиК, Москва, 2002–2004., с. 42 – 44
26. Jan Sochman, AdaBoost / Jan Sochman, Jiri Matas // Center for Machine Perception, Czech Technical University, Prague, 2010
27. Yoav Freund, A Short Introduction to Boosting / Yoav Freund, Robert E. Schapire // Shannon Laboratory, USA, 1999., pp. 771-780
28. Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц -
Режим доступа:
<https://hsto.org/storage1/b5e22c69/33e407c1/dfd2da2c/821151e7.jpg> -
Дата доступа: 17.06.2016

- 29.Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц -
Режим доступа:
<https://hsto.org/storage1/d3c33a53/2f27fbf9/4e78708a/1c60fe94.jpg> -
Дата доступа: 17.06.2016
- 30.OpenCV Face Detection: Visualized - Режим доступа:
<https://vimeo.com/12774628> - Дата доступа: 17.06.2016
- 31.Independent Component Analysis: A Demo – Режим доступа:
<http://research.ics.aalto.fi/ica/icademo/> - Дата доступа: 17.06.2016
- 32.Joint probability distribution – Режим доступа:
https://upload.wikimedia.org/wikipedia/commons/thumb/9/95/Multivariate_normal_sample.svg/663px-Multivariate_normal_sample.svg.png - Дата доступа:
17.06.2016
- 33.Mode: Merits & Demerits – Режим доступа: <http://www.durofy.com/wp-content/uploads/2013/07/mode3.png> – Дата доступа: 17.06.2016