

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «МЕТОД ТА МІКРОПРОЦЕСОРНІ ЗАСОБИ ОПЕРАТИВНОГО
КОНТРОЛЮ СТАНУ ВОДІЯ АВТОМОБІЛЯ»

Виконав: студент 2 курсу, групи ІКІ-23м
Спеціальності 123 Комп'ютерна інженерія

Давидун М. В. _____

Керівник к.т.н., доц. каф. ОТ

Богомолів С.В. _____

" " _____ 2024р.

Опонент к.т.н., доц. каф. ІІЗ

Кательніков Д.І. _____

" " _____ 2024р.

Допущено до захисту

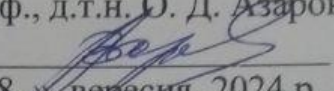
Завідувач кафедри ОТ

д.т.н., проф. Азаров О.Д. _____

« _____ » _____ 2024 р.

ВНТУ 2024

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень магістр
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
обчислювальної техніки
проф., д.т.н. О. Д. Азаров

« 18 » вересня 2024 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Давидуну Максиму Володимировичу

1 Тема роботи: Метод та мікропроцесорні засоби оперативного контролю стану водія автомобіля», керівник роботи Богомолів Сергій Віталійович, к.т.н., доцент кафедри ОТ, затверджені наказом вищого навчального закладу від 17.09.2024 року № 310.

2 Строк подання студентом роботи 18.12.2024 р.

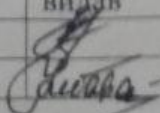
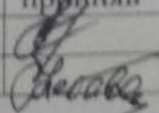
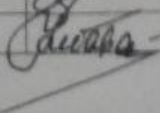
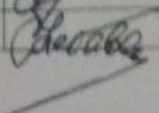
3 Вихідні дані до роботи: технічний опис мікропроцесорних платформ, технічна документація на електронні компоненти.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, огляд і аналіз методів та засобів оцінювання стану водія, об'єкти дослідження та удосконалення методу спостереження за очима водія, реалізація методу та програмного забезпечення для мікропроцесорної системи контролю водія, експериментальні дослідження, економічна частина, висновки

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): блок схема алгоритму, схема електрична структурна, схема електрична принципова.

6 Консультанти розділів роботи наведено в таблиці 1.

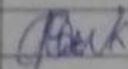
Таблиця 1 — Консультанти розділів роботи

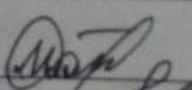
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	Богомолів С.В., к.т.н., доц. каф. ОТ		
5	к. е. н., професор Небава Микола Іванович		

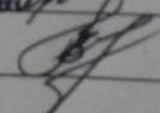
7 Дата видачі завдання 18.09. 2024 р.

8 Календарний план наведено в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Прим.
1	Постановка мети та задач роботи	20.10.24	
2	Огляд та аналіз методів та мікропроцесорних засобів оперативного контролю стану водія	30.10.24	Вик.
3	Вибір моделі мікроконтролера	04.11.24	Вик.
4	Проектування апаратної частини	08.11.24	Вик.
5	Розробка електричної принципової схеми	19.11.24	Вик.
6	Проектування програмної частини	24.11.24	Вик.
7	Розрахунок економічної частини роботи	01.12.24	Вик.
8	Оформлення пояснювальної записки та ілюстративного матеріалу	10.12.24	Вик.
9	Аналіз виконання роботи, висновки		Вик.
10	Перевірка якості виконання магістерської роботи та усунення недоліків		Вик.

Студент  Давидун М.В.

Керівник роботи  Богомолів С.В.

АНОТАЦІЯ

УДК 629.113:004.35:612.821

Давидун М.В. Метод та мікропроцесорні засоби оперативного контролю стану водія автомобіля. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна інженерія, освітня програма — Комп'ютерна інженерія. Вінниця: ВНТУ, 2024, 107 с.

На укр.мові. Бібліогр.: 25 назв, рис 61, табл.18.

У магістерській кваліфікаційній роботі проведено дослідження та розробку методів та мікропроцесорних засобів для оперативного контролю стану. Метою роботи є розробка приладу для контролю стану водія за кермом автомобіля.

У роботі проведено огляд та аналіз основних параметрів та методів контролю водія, також наведено існуючі мікропроцесорні системи та постановка задачі проектування. Також розроблено структурної схема та схеми електричної принципової. Проведено вибір електронних компонентів для вимірювальної системи. Вибрана методика моніторингу, а також розроблено алгоритм роботи приладу. Вибрано бібліотеки для реалізації лістингу програми, також наведено середовище розробки скетчу, а також інструкцію користувача.

Ключові слова: мікропроцесорна система, комп'ютерний зір, безпека дорожнього руху, машинне навчання, запобігання аваріям.

ANNOTATION

Davydun M.V. Methods and microprocessor means of operational control of the condition of the car driver. Master's thesis on specialty 123 — Computer engineering, educational program — Computer engineering. Vinnytsia: VNTU, 2024.

This master's thesis is devoted to the research and development of methods and microprocessor tools for operational state control. The purpose of the work is the development of a device for monitoring the state of the driver behind the wheel of a car model.

The work provides an overview and analysis of the main parameters and methods of driver control, as well as the existing microprocessor systems and the formulation of the design problem. A structural diagram and an electrical schematic diagram have also been developed. The selection of electronic components for the measuring system was carried out. The monitoring method was selected, as well as the device operation algorithm was developed. Libraries are selected for the implementation of the program listing, the sketch development environment is also given, as well as the user manual.

Keywords: microprocessor system, computer vision, traffic safety, machine learning, accident prevention.

ЗМІСТ

ВСТУП.....	5
1 ОГЛЯД І АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ОЦІНЮВАННЯ СТАНУ ВОДІЯ	9
1.1 Аналіз факторів, що впливають на безпеку транспорту	9
1.2 Соціальна політика щодо інформування водіїв.	11
1.3 Аналіз наявних рішень у сфері дорожньої безпеки.....	13
1.4 Аналіз методів розпізнавання обличчя і машинного зору.....	23
1.4.1 Метод гнучкого порівняння на графах.	27
1.4.2 Метод нейронних мереж	29
1.4.3 Сховані Марківські моделі (СММ, НММ).....	31
1.4.4 Метод головних компонентів або principal component analysis (PCA).....	32
1.5 Порівняння методів детектування обличчя.....	35
2 ОБ'ЄКТИ ДОСЛІДЖЕННЯ ТА УДОСКОНАЛЕННЯ МЕТОДУ СПОСТЕРЕЖЕННЯ ЗА ОЧИМА ВОДІЯ.....	36
2.1 Цифрове зображення, методи їх обробки та основи комп'ютерного зору.....	36
2.2 Методи розпізнавання об'єктів у цифрових зображеннях	38
2.3 Аналіз методів детектування стану обличчя бібліотеки dlib.....	41
2.4 Побудова удосконаленого методу системи детектування очей водія.....	43
3 РЕАЛІЗАЦІЯ МЕТОДУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ КОНТРОЛЮ СТАНУ ВОДІЯ	47

					08-54.МКР.005.00.000 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.	Давидун М.В.				Метод та мікропроцесорні засоби оперативного контролю стану водія Пояснювальна записка	Літ.	Арк.
Перевір.	Богомолов С.В.						
ОпONENT.	Кательніков Д.І.					6	111
Н. Контр.	Швець С.І.					ВНТУ, 1КІ-23м	
Затверд.	Азаров О.Д.						

3.1	Аналіз мікропроцесорних платформ.....	47
3.2	Вибір електронних компонентів.....	50
3.3	Реалізація апаратного засобу	51
3.4	Мова програмування Python	52
3.5	Побудова алгоритму програми та практична реалізація системи.	59
4	ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ.....	66
4.1	Експериментальне дослідження системи	66
4.2	Опис основних етапів тестування	66
4.3	Покрокове тестування системи	68
5	ЕКОНОМІЧНА ЧАСТИНА	72
5.1	Комерційний та технологічний аудит науково-технічної розробки	72
5.2	Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	76
5.3	Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	83
	ВИСНОВКИ	90
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	91
	ДОДАТОК А Технічне завдання	94
	ДОДАТОК Б Лістинг програми	98
	ДОДАТОК В Схема електрична структурна.....	102
	ДОДАТОК Г Схема електрична принципова	103
	ДОДАТОК Д Перелік елементів.....	104
	ДОДАТОК Е Протокол перевірки кваліфікаційної роботи.....	105

ВСТУП

Керування автомобілем — це складна, багатогранна та потенційно небезпечна діяльність, що вимагає повного залучення фізіологічних і когнітивних ресурсів для підтримання продуктивності протягом часу. Будь-яка втрата цих ресурсів може мати серйозні наслідки, включаючи аварії. Крім того, з появою автономних транспортних засобів стає ще важливішим контролювати стан водія. Останні дослідження охоплюють як фундаментальні аспекти, так і практичні застосування, зокрема визначення та прогнозування погіршення стану водія. Автомобілі використовуються для перевезення людей, вантажів та іншого між різними локаціями.

Для перевезення, безпека має бути пріоритетом номер один. При перевезенні людей або будь-якого іншого предмета ми повинні думати про захищеність пасажирів транспортного засобу, громадських та приватних та громадських об'єктів. З початку 2024 року на дорогах України в результаті трапилося 23642 дорожньо-транспортних пригод (ДТП) з них для 3053 людей були фатальними. Цю інформацію розповсюдила патрульна поліція України [1]. Наведені вище статистичні дані підтверджують, що в автомобільній та транспортній промисловості має бути багато покращень для забезпечення безпеки та благополуччя людей, а також безпеки приватної та державної власності.

Існує безліч факторів, які сприяють дорожньо-транспортним травмам і летальним випадкам. Серед них — втрата контролю над транспортним засобом, перевищення швидкості, вживання алкоголю, сонливість водія, несприятливі погодні умови та технічний стан автомобіля. Багато водіїв не приділяють належної уваги своєму фізичному та психологічному стану перед тим, як сісти за кермо. Незважаючи на втому, сонливість чи вживання алкоголю, вони часто продовжують керувати транспортом без усвідомлення ризиків. Це призводить до аварій, травм, загибелі людей та матеріальних втрат. Існують системи, здатні

передбачити стан водія ще до виникнення критичної ситуації, що могло б значно допомогти уникнути нещасних випадків.

Сонливе водіння є серйозною загрозою для безпеки на дорогах. Якщо вдасться попередити водіїв ще до того, як вони стануть занадто сонними для безпечного керування, можна уникнути частини аварій, спричинених сонливістю. Однак ефективність таких попереджень залежить від надійного виявлення сонливості. Наразі існуючі методи обмежені тим, що не враховують індивідуальні відмінності між водіями, що знижує їхню точність.

Інтелектуальна технологія виявлення сонливого водіння, як важлива складова інтелектуальних мережевих технологій у транспорті, була представлена в цьому документі для розпізнавання втомленого стану водія та запобігання дорожньо-транспортним пригодам.

Отже, основна мета цього проекту — виявлення сонливості водія за допомогою комп'ютерного зору та аналізу рис обличчя, щоб надавати своєчасні попередження. Це дозволить водію ухвалити рішення щодо безпеки подальшого руху та вжити відповідних заходів у разі необхідності.

Актуальність теми дослідження полягає у необхідності розробки системи для контролю стану водія на основі сучасних мікропроцесорів і комп'ютерного зору, що допоможе зменшити кількість випадків дорожньо-транспортних пригод.

Метою роботи є створення мікропроцесорного засобу оперативного контролю стану водія, яка дозволить ефективно аналізувати і миттєво повідомляти про нестабільний стан водія. Що у свою чергу, дозволить зменшити відсоток дорожньо-транспортних пригод.

Задачі дослідження:

- виконати огляд та аналіз існуючих методів та мікропроцесорні засобів оперативного контролю стану водія автомобіля;

- запропонувати метод оперативного контролю стану водія автомобіля;

- спроектувати архітектуру системи на основі зібраних даних дослідження;
- розробити апаратну частину для системи, включаючи вибір компонентів;
- створити алгоритм роботи та розробити програмне забезпечення для системи моніторингу стану водія з використанням обробки відеопотоку в реальному часі;
- провести експериментальні дослідження розробленої системи;
- здійснити обґрунтування доцільності впровадження системи контролю стану водія, враховуючи потенційні переваги для безпеки дорожнього руху;
- провести розрахунок економічних затрат для створення мікропроцесорної системи контролю стану водія та визначити економічні вигоди від впровадження цієї технології в автомобілях.

Об’єкт дослідження — процес моніторингу стану водія шляхом аналізу відеопотоку, що дозволяє виявити ознаки втоми, відсутності уваги та інших факторів ризику, використовуючи методи комп’ютерного зору для розпізнавання обличчя та стану очей у реальному часі.

Предмет дослідження — методи комп’ютерного зору та обробки відеопотоку для виявлення стану водія, включаючи алгоритми розпізнавання обличчя, аналізу виразу обличчя та стану очей для оцінки уваги та втоми водія.

Використовувались **методи дослідження**: комп’ютерного зору для розпізнавання обличчя та стану очей, алгоритми машинного навчання для класифікації стану водія, цифрової обробки зображень для поліпшення якості відеопотоку, математичної статистики для оцінки точності виявлення ознак втоми та відсутності уваги, а також принципи об’єктно-орієнтованого програмування для реалізації програмного забезпечення системи моніторингу стану водія.

Наукова новизна одержаних результатів полягає у вдосконаленні системи моніторингу стану водія, що забезпечує точне виявлення втоми

завдяки поєднанню методів комп'ютерного зору та машинного навчання, оптимізації алгоритмів розпізнавання обличчя та стану очей.

Практичне значення одержаних результатів полягає у можливості створення ефективної системи моніторингу стану водія, яка виявляє ознаки втоми в реальному часі, забезпечує підвищення загальної безпеки дорожнього руху та зниження ризику аварій

Апробацію результатів наукової роботи було проведено на науковій конференції «Молодь в науці: дослідження, проблеми, перспективи (МН–2025)», доповідь на тему “Метод та мікропроцесорні засоби оперативного контролю стану водія автомобіля”. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/22532>

1 ОГЛЯД І АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ОЦІНЮВАННЯ СТАНУ ВОДІЯ

1.1 Аналіз факторів, що впливають на безпеку транспорту

Міжнародна статистика свідчить, що близько чверті дорожньо-транспортних пригод у різних країнах відбувається через втому водія. Засинати за кермом так само небезпечно, як і водіння в нетверезому стані, і становить серйозну загрозу для дороги. З кожним роком у світі зростає кількість аварій, пов'язаних із сонним водінням. Опитування показують, що в Україні дві третини водіїв із проблемами сну засинали за кермом, а 42% з них потрапили в ДТП через надмірну втому.

Сонливість за кермом може бути викликана різними факторами, такими як якість сну, час доби, вік, стан здоров'я та вживання алкоголю. Водії, які відчувають сонливість, часто недосипають, працюють на кількох роботах або їздять протягом тривалого часу. Нічне водіння може бути у два-шість разів небезпечніше водіння вдень, оскільки, окрім погіршення видимості, посилюється сонливість і погіршуються когнітивні функції між 2:00 і 7:00, незалежно від того, скільки часу людина відпочиває.

Переоцінка рівня пильності підвищує ризик сонливості за кермом. Втома проявляється в результаті перенапруження м'язів, погіршення сенсомоторних реакцій, погіршення пам'яті, зниження здатності до концентрації в одноманітних умовах. Багато водіїв недооцінюють свій рівень сонливості та помилково вважають, що вони достатньо уважні. Вони часто намагаються заповнити недосип кавою чи іншими стимуляторами, але переоцінюють їх ефективність.

Симптоми засинання за кермом:

- труднощі з фокусуванням зору;
- часте моргання;
- відчуття важкості в повіках;
- важко тримати голову прямо;

- часте позіхання;
- ігнорувати дорожні знаки;
- часті виїзди зі смуги руху;
- труднощі з дотриманням дистанції;
- наїзд на шумову смугу на узбіччі.

У сучасній практиці для характеристики багатьох факторів, що впливають на професійну діяльність, часто використовують терміни «втома» і «млявість». Втома — це гіпотетична концепція, яка пояснює взаємозв'язок між різними факторами, які викликають розвиток втоми та загрожують безпеці водія. До основних показників, пов'язаних з появою втоми, відносяться:

- обмежена тривалість сну, що визначається тривалістю попереднього неспання і сну;
- час доби (добовий ритм);
- характеристика роботи (тривалість, інтенсивність, складність).

Основною причиною перерв у будь-якій професійній діяльності є сонливість, яка негативно позначається на виконанні завдань. Підвищена сонливість поступово знижує здатність водія ефективно керувати автомобілем. Це впливає на пильність і концентрацію на дорозі, що може призвести до аварійних ситуацій. Хоча причини втоми та сонливості різні, їхні наслідки схожі із зниженням когнітивних і фізичних здібностей.

Одним із ключових аспектів людської діяльності є режим сну протягом 24 годин, який називається циркадним ритмом. Ризик засинання вдень і тривалість сну сильно відрізняються, Тому що на ці процеси впливає ендогенний годинник мозку. Дослідження показують, що найвища ймовірність дорожньо-транспортних пригод виникає раніше ніж максимальний рівень сонливості, зокрема між 2:00 і 3:00 ночі. Важливими факторами також є тривалість неспання, час, що минув від початку подорожі, періоди відпочинку та інші фактори, які слід брати до уваги [2].

Вивчаючи вплив тривалості робочого дня на втому, важливо враховувати її зв'язок з часом доби. Дослідження, засноване на обширних даних, включаючи

різні показники активності та рівні неспання, припускає, що рівень втоми більше залежить від часу доби, ніж від часу водіння.

Водіння є однією з найпоширеніших видів діяльності, яка серйозно впливає на депривацію сну. Дослідження показують, що 10–20% серйозних дорожньо-транспортних пригод стаються через те, що водії засинають за кермом. Були вжиті ретельні заходи для підвищення безпеки водіїв. Комітет ЄС з транспорту та енергетики висунув низку рекомендацій щодо підвищення безпеки на транспорті, спрямованих на боротьбу з втомою та запобігання ДТП. Щоб бути справді комплексними та ефективними, ці заходи мають охоплювати всі аспекти: водіїв, транспортні компанії, дороги та автомобілі [3].

Поради щодо безпеки наших доріг:

- водії повинні пройти навчання, яке включає пояснення небезпеки та наслідків підвищеного рівня втоми та сонливості;
- транспортні компанії повинні регулярно інформувати водіїв про ризики втоми за кермом та створювати умови для підтримки оптимальної системи праці та відпочинку;
- дороги мають бути обладнані віброшумовими бордюрами на узбіччях і розділених смугах, що може сприяти зменшенню аварійності, активізуючи сонливих водіїв, особливо вночі;
- автомобілі повинні бути обладнані системами, здатними виявляти зміни фізіологічного рівня збудження та параметрів водіння, які вказують на втому водія.

1.2 Соціальна політика щодо інформування водіїв.

Соціальна політика щодо обізнаності водіїв – це комплексний підхід до забезпечення безпеки дорожнього руху та підвищення обізнаності водіїв шляхом поширення інформації, навчання та контролю. Основні компоненти політики включають:

- комунікаційна діяльність, спрямована на формування правильних моделей поведінки та правил дорожнього руху через рекламні заходи та інформаційні заходи;

- освітня програма, що включає курси підготовки та навчання водіїв на всіх етапах їх діяльності;

- розробити інформаційні матеріали, що містять правила дорожнього руху, поради щодо безпечного водіння, нові технології;

- використовуйте сучасні технології, такі як мобільні додатки, веб-платформи та соціальні мережі, щоб поширювати інформацію серед водіїв;

- поліція регулярно патрулює дороги та встановлює системи відеоспостереження за дотриманням правил дорожнього руху;

З метою попередження серйозних дорожньо-транспортних пригод, спричинених засипанням за кермом, було розпочато широкомасштабну рекламну кампанію щодо небезпеки зниження уваги за кермом та підвищеної сонливості. Для обговорення пропонуються такі теми:

- уникайте алкоголю та ліків, які можуть викликати сонливість як побічний ефект;

- вживайте кофеїн (еквівалент двох чашок кави може підвищити пильність на кілька годин);

- оцініть ризик водіння в сонному стані, за що водій особисто відповідає перед іншими;

- інформувати населення про необхідність повноцінного сну та ознаки втоми;

- уникайте водіння в той час, коли ви зазвичай спите.

У країнах Західної Європи ми часто бачимо гасло «Don't drive while fatigued», що перекладається як «Не сідайте за кермо, коли ви втомлені» (рис. 1.1).

Необхідно створити подібний вислів українською мовою, який можна було б використовувати на плакатах у громадських місцях. Проте практика показує, що подібних рекламних кампаній недостатньо для того, щоб приватні

водії усвідомили шкоду сонливості за кермом. Для професійних водіїв також важливо законодавчо визначити необхідний режим роботи, сну та відпочинку, що вже запроваджено в багатьох країнах [4].



Рисунок 1.1 — Слоган "Don't drive tired" на дорогах

1.3 Аналіз наявних рішень у сфері дорожньої безпеки.

Є два способи досягнення цієї функції. Перший варіант передбачає використання спеціальних датчиків для відстеження параметрів руху автомобіля, а саме частоти і амплітуди поворотів, а також тиску на педалі газу і гальма. Цей підхід популярний серед європейських виробників, таких як Mercedes, Volkswagen, Skoda та Volvo. У свою чергу, японські компанії підходять до проблеми втоми за кермом з іншої точки зору, пропонуючи альтернативні рішення.

Першочерговим завданням виробник вважає аналіз психоемоційного стану водія. Тому ключовим компонентом системи є камера, призначена для відстеження міміки та жестів водія. Система спочатку виявляє ознаки сонливості, наприклад закривання очей. Якщо водій закриває очі, система негайно видає попереджувальний сигнал. Робота інженера полягає в тому, щоб

«навчити» систему відрізняти просте моргання від сонливості. Крім того, аналізується частота моргання, рухи очей, міміка, жести, частота і глибина дихання, що визначаються рухами грудної клітки.

На початковому етапі блок управління збирає та аналізує всю інформацію, отриману від датчиків і камер. Система визначає стиль водіння водія та умови навколишнього середовища (такі як час доби, стан дороги та силу вітру), порівнюючи ці дані з еталонними значеннями. Ця інформація є основою для подальшої ідентифікації втоми водія шляхом порівняння з поточними показниками. Для різних автомобілів збір вихідних даних потрібен по-різному: для Mercedes SLK, наприклад, потрібно близько півгодини, а для Volkswagen Passat і Skoda Octavia – 15 хвилин. Такий підхід значно розширює можливості системи ідентифікації, оскільки контроль сонливості водія базується на порівнянні з конкретними показниками.

Однак існуючі рішення мають свої недоліки. Через особливості системи моделі алгоритм розпізнавання втоми може не працювати належним чином, що принесе додаткову стимуляцію та напругу водієві. Однією з головних труднощів у реалізації такої системи є розробка алгоритму, який може точно розрізняти стани втоми водія та випадкові стани з подібними параметрами.

Неможливо оцінити сонливість лише за одним параметром, наприклад за стилем водіння. Щоб зробити точну оцінку, необхідно врахувати всі аспекти, пов'язані з автомобілем і водієм. У цьому випадку найбільш ефективним рішенням є поєднання існуючих рішень від провідних виробників автомобілів із впровадженням нових технологій.

По-перше, зверніть увагу на стан водія. Для цього слід за допомогою відеокамери або веб-камери стежити за виразом обличчя водія, а саме за ступенем розплющення очей, напрямком погляду, частотою моргань і позіхань, рухів головою (поворотів і нахилів). По-друге, важливим аспектом є визначення психоемоційного стану водія, оскільки його нестійкість може призвести до значної розсіяності та втоми.

Для цього можна використовувати вбудований в кермо датчик пульсу. Проте лише обмеження цих параметрів не дає адекватної оцінки стану водія, що може негативно вплинути на результати. Тому датчики, які відстежують переміщення автомобіля, також потрібні. Враховуються такі фактори, як положення автомобіля на смузі руху, умови руху, характер повороту керма, стан дорожнього покриття та тип руху автомобіля. Іншим параметром може бути аналіз атмосфери всередині автомобіля, особливо гучності музики або голосів водія та пасажирів.

Дані з датчиків і камер збираються та аналізуються в блоці керування. У разі необхідності, залежно від ступеня небезпеки (який відрізняється висотою та гучністю), водія попереджають голосом або звуковим сигналом. У ситуаціях, коли ризик аварії високий, можна вжити заходів безпеки, таких як зниження швидкості, повна зупинка, увімкнення аварійної сигналізації та інші профілактичні заходи.

Великі автомобільні компанії змінили свої стратегії щодо безпеки водія та захисту автомобіля. Якщо раніше основна увага приділялася зменшенню наслідків дорожньо-транспортних пригод за допомогою ременів безпеки та подушок безпеки, то зараз у пріоритеті запобігання ДТП на ранніх стадіях. Багато виробників автомобілів активно розробляють різні системи моніторингу стану водія, які можуть як мінімум повідомляти про небезпечні ситуації, так і в ідеалі втручатися в управління транспортним засобом для запобігання аварій. Ця діяльність охоплює кілька сфер, включаючи моніторинг втоми, оцінку навантаження на тіло та виявлення патології водія.

Оскільки безпека дорожнього руху є головним пріоритетом, багато виробників висококласних автомобілів інтегрують системи безпеки у свої транспортні засоби. Однак ці автомобілі досить дорогі і недоступні для багатьох людей. Тому наше дослідження спрямоване на підвищення безпеки автомобіля за допомогою інфрачервоного світла та акселерометрів, вбудованих в оправу окулярів. Це покращить практику забезпечення безпеки водія шляхом

моніторингу стану очей і положення голови. Такий підхід зробить безпеку дорожнього руху доступнішою для широкої аудиторії.

Запропонована мікропроцесорна система буде реалізована за допомогою методів довгого інфрачервоного відображення та акселерометра, що підвищить ефективність і точність системи. Інфрачервоний датчик стане основним компонентом пристрою, а акселерометр відіграє додаткову роль у забезпеченні двоетапного сповіщення користувача.

Вивчення існуючих систем є одним із ключових аспектів цієї роботи. Це дозволяє ознайомитися з наявними рішеннями та оцінити їх сильні та слабкі сторони. Такий аналіз допомагає покращити функціональність системи шляхом впровадження нових функцій або вдосконалення існуючих. З цією метою ми проаналізували три схожі системи, використовуючи інформацію з відповідних онлайн-джерел, які містять огляд літератури щодо цих систем. Отримані висновки пояснюються в наступному розділі. Крім того, вибрані системи порівнюються, щоб забезпечити більш детальний аналіз автомобільної промисловості та поточного стану безпеки та систем виявлення сонливості водія.

StopSleep — пристрій, призначений для моніторингу активності користувача та запобігання засипанню (рис. 1.2).



Рисунок 1.2 — Пристрій Stopsleep

StopSleep — це пристрій, який постійно контролює стан водія. Він не тільки записує момент, коли ви засинаєте, але й попереджає вас про зниження відповідей і можливість заснути на дві-три хвилини раніше. Завдяки такому ранньому сигналу водій може вчасно зробити перерву, потягнутися або змінити положення.

Портативний пристрій, який носять на двох пальцях і має форму кільця, використовує вібрацію, світлові сигнали та звукові сигнали, щоб попередити водіїв. Всі його елементи виготовлені з матеріалів, які не викликають алергічних реакцій, а особливо з міцного пластику.

StopSleep має вісім вбудованих датчиків, які вимірюють температуру шкіри за допомогою технології, схожої на поліграф. Пристрій генерує два типи попереджень: рівень обережності та небезпека. У небезпечних ситуаціях він вібрує та видає гучний сигнал, щоб попередити водія про потенційну загрозу [5].

Пристрій потребує регулярної зарядки та може працювати до 15 годин на повному заряді. Портативний StopSleep продається приблизно за 100 доларів.

Система безпеки StopSleep використовує тепло шкіри водія для визначення його стану, що робить її унікальною в порівнянні з іншими системами. Однак більш детальний аналіз пристрою виявив кілька проблем. Температура тіла може змінюватись через низку факторів, що викликає сумніви щодо точності системи та здатності водія ефективно керувати автомобілем. Як портативний пристрій, StopSleep необхідно носити на двох пальцях, що може бути незручним і відволікати водіїв. Крім того, датчик може не працювати належним чином під час тривалого використання, особливо під час потовиділення. Низький заряд батареї може негативно вплинути на функціональність вашої системи під час тривалих подорожей. Крім того, висока вартість пристрою може стати істотним обмеженням для багатьох користувачів.

Сонливість, неуважність і втома водія становлять серйозну загрозу і є причиною багатьох аварій. Аналізуючи поведінку водія за кермом, перші ознаки втоми можна виявити на ранній стадії. Система виявлення сонливості

Bosch (рисунок 1.3) використовує датчик кута повороту рульового колеса або систему підсилювача керма для моніторингу поведінки водія та виявлення ознак втоми.



Рисунок 1.3 — Виявлення сонливості водія BOSCH

Система виявлення сонливості Bosch також може розпізнавати раптові рухи керма. Він враховує різні параметри, такі як тривалість поїздки та час доби, щоб оцінити стан втоми водія. У разі виявлення втоми водій отримує звуковий або візуальний сигнал, що попереджає про необхідність перерви. За статистикою в 2013 році 23% нових автомобілів були оснащені цією системою. Система виявлення сонливості водія Bosch відстежує поведінку керма та прогнозує стійкість водія та можливу сонливість. Він аналізує близько 70 різних сигналів для оцінки сонливості. Bosch використовує датчик кута повороту керма для вимірювання кута повороту та швидкості повороту. Алгоритми системи відстежують моделі водіння на початку поїздки та аналізують різні параметри, такі як несподівані рухи керма та використання сигналів, щоб визначити стан втоми водія. Коли втома досягне критичного

рівня, система попередить водія, що йому або їй потрібно зробити перерву, блимаючи зображенням чашки кави або іншим способом [6].

Система виявлення сонливості водія Bosch використовує датчик кута повороту рульового колеса для виявлення сонливості. Якщо буде виявлено незвичайний рух керма, система відобразить великий символ кавової чашки на приладовій панелі автомобіля. Однак система не аналізує рухи керма в межах смуги та не має датчиків для безпосереднього виявлення стану водія. Крім того, миготливі вогні на приладовій панелі можуть бути недостатньо ефективними для попередження водія, особливо при яскравому сонячному світлі. Тому водії не можуть покладатися лише на систему запобігання ДТП.

Anti Sleep Pilot (рис. 1.4) — це пристрій, розроблений датською компанією ASP Technology Ltd. Зазвичай він встановлюється на приладовій панелі автомобіля і використовується для постійного контролю за станом водія та умовами водіння. Прилад оснащений датчиками освітленості, звуку і сенсорними елементами і працює від акумулятора. Спочатку водії проходять оцінку, щоб створити профіль ризику, а потім можуть почати використовувати пристрій.



Рисунок 1.4 — Пристрій Anti sleep pilot

Anti Sleep Pilot автоматично контролює рівень уваги водія за допомогою 26 параметрів, включаючи профілі ризику. Пристрій також проводить

випадкові тести, під час яких водій повинен натиснути на датчик. Якщо реакції водія сповільнені, це свідчить про знижену активність, а тривала сповільненість реакцій спонукає водія відпочити принаймні десять хвилин, перш ніж продовжити керування автомобілем.

Крім того, Anti Sleep Pilot відстежує час і швидкість автомобіля, щоб визначити, коли водієві дійсно потрібна перерва. Пристрій подає звуковий і світловий сигнал, який водій повинен вимкнути натисканням пристрою. Anti Sleep Pilot рекомендує зупинитися та відпочити, якщо водій відчуває сонливість, засинає за кермом або повільно реагує [7].

Anti Sleep Pilot складається з наступних компонентів:

- акселерометр, який фіксує прискорення автомобіля та використовується для аналізу стилю водіння;
- високоточний годинник, який вимірює час реакції та інтервали активності для розрахунку втоми водія;
- датчик освітлення автоматично регулює яскравість дисплея, щоб забезпечити оптимальне освітлення в автомобілі;
- звукові датчики, які адаптують звуковий сигнал до рівня шуму в салоні;
- сенсорний датчик для спрощення взаємодії з пристроєм під час водіння;
- розумна кнопка ввімкнення/вимкнення дозволяє економити заряд акумулятора, коли Anti Sleep Pilot не використовується;
- прилад працює від стандартних батарейок типу AAA і простий в установці.

Anti Sleep Pilot — система безпеки водія, яка використовує датчики світла, звуку та тиску. Він контролює стан водія, проводячи випадкові тести, під час яких водій повинен відповісти натисканням пристрою протягом заданого часу. Якщо водій реагує повільно, система видає попередження за допомогою звукового сигналу.

Хоча це здається ефективним рішенням, оскільки система слідує за основним водієм, відволікання від натискання пристрою під час водіння може стати небезпечним і відволікати водія. Таким чином, система може збільшити, а не зменшити ризик дорожньо-транспортних пригод.

Mercedes Benz став одним із перших автовиробників, який впровадив систему попередження про виїзд зі смуги руху на своїх вантажівках Actros у 2000-х роках. Відтоді більшість провідних автовиробників також впровадили подібні системи у свої моделі.

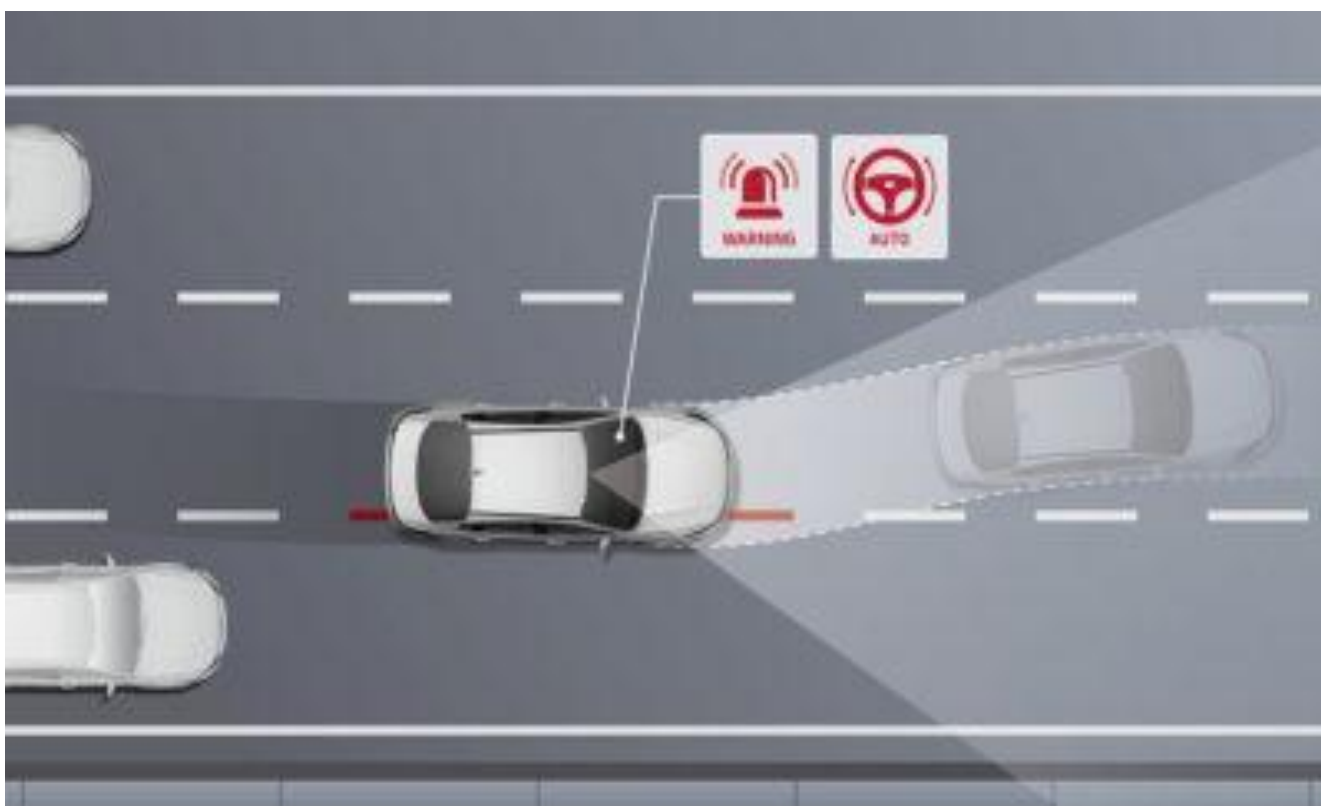


Рисунок 1.5 — Система попередження про сходження зі смуги

Більшість виробників автомобілів високого класу впровадили такі системи, як адаптивний круїз-контроль, попередження про лобове зіткнення, попередження про виїзд зі смуги руху та виявлення сліпих зон за допомогою сучасних технологій [8].

Одним із недоліків цієї системи є те, що вона може викликати плутанину, коли на дорозі є кілька смуг. Крім того, він обмежений моніторингом дорожньої розмітки, що ускладнює використання на дорогах без такої розмітки,

особливо на гравійних дорогах. Також варто зазначити, що система неефективно відстежує лінії доріг під час дощу, туману чи снігу. У цій ситуації водій не може повністю покладатися на систему безпеки. Крім того, ці системи коштують понад 1000 доларів і доступні лише на автомобілях високого класу.

Система безпеки водія Tesla Autopilot (рис. 1.5) була представлена компанією Tesla Motors у 2014 році. Він складається з восьми камер, розташованих по периметру автомобіля, що забезпечує 360-градусну видимість на відстані до 250 метрів. Система також включає 12 вдосконалених ультразвукових датчиків, які можуть виявляти тверді та м'які перешкоди на відстані до 250 метрів. Крім того, передній радар надає інформацію про погодні умови, такі як дощ, туман і пил.

Система обробляє дані, отримані від камер і датчиків, що дозволяє комп'ютеру в автомобілі взяти на себе керування без участі водія у разі помилки чи небезпеки. Це ставить систему Tesla попереду більшості інших систем безпеки водія на ринку, забезпечуючи водіям високий рівень автоматизації та безпеки.

Однак, незважаючи на численні переваги, система не є на 100% безпечною, і водії повинні залишатися пильними, оскільки автономне водіння не замінює їхню відповідальність керувати автомобілем [9].



Рисунок 1.6 — Автопілот Tesla

Система безпеки водія Autopilot від Tesla є справді однією з найпередовіших автомобільних технологій. Він складається з восьми камер і дванадцяти ультразвукових датчиків для виявлення жорстких і м'яких перешкод навколо автомобіля. Збираючи та аналізуючи ці дані, система може керувати автомобілем без участі водія, що робить систему дуже потужною та ефективною з точки зору безпеки.

Основні особливості системи:

- автономність: система здатна виконувати багато функцій без безпосереднього втручання водія, що підвищує комфорт під час подорожі;
- панорамний огляд: вісім камер забезпечують огляд на 360 градусів, дозволяючи автомобілю реагувати на різні ситуації на дорозі;
- адаптивність: система враховує інформацію про навколишнє середовище, яка може вплинути на безпеку руху, включаючи погодні умови.

Однак, незважаючи на всі переваги, висока вартість системи обмежує її доступність для багатьох потенційних користувачів. Це актуалізує питання рівного доступу до сучасних технологій безпеки дорожнього руху. Тому, хоча система Tesla може значно підвищити безпеку, вона все ще недоступна широкій громадськості через свою ціну.

1.4 Аналіз методів розпізнавання обличчя і машинного зору

Розпізнавання обличчя — це процес автоматичного визначення місцезнаходження обличчя на зображеннях або відеопотоках і, якщо необхідно, ідентифікації людини на основі даних у існуючій базі даних. Попит на такі системи значно зріс у зв'язку з їх широким застосуванням у різних сферах діяльності [10].

Розпізнавання обличчя — це біометрична програмна програма, яка може однозначно ідентифікувати або підтверджувати особу шляхом порівняння та аналізу шаблонів на основі контурів обличчя (рисунки 1.7). Кожна людина має унікальні риси обличчя, і спеціалізоване програмне забезпечення здатне

проаналізувати ці риси та зіставити їх з інформацією в базі даних для подальшої ідентифікації.

Основним недоліком технології розпізнавання осіб є погіршення якості розпізнавання в наступних ситуаціях:

- недостатнє освітлення;
- зміни положення голови.

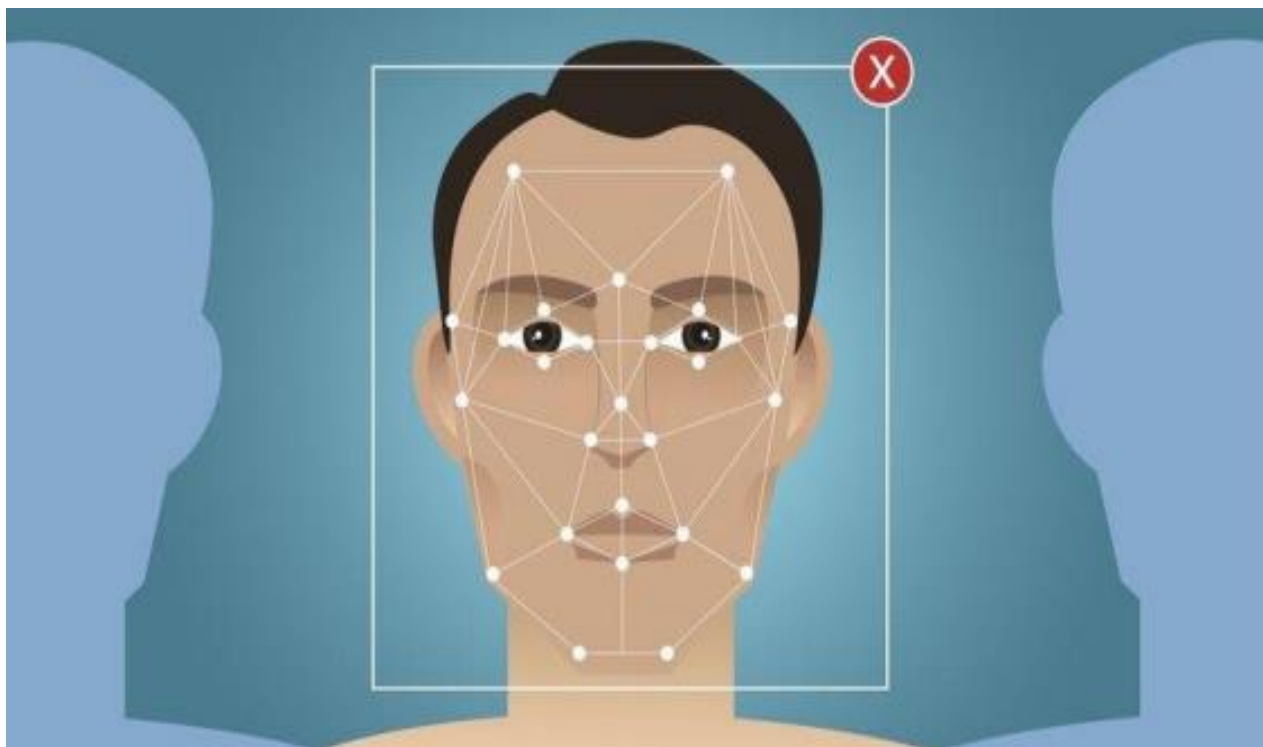


Рисунок 1.7 — Технологія розпізнавання облич

Існує багато способів розробки алгоритмів розпізнавання обличчя. Емпіричні методи, які використовувалися на початкових етапах розвитку комп'ютерного зору (рис. 1.8), ґрунтувалися на деяких правилах, за якими люди ідентифікували людей. Наприклад, чоло зазвичай отримує більше освітлення, ніж центральна частина обличчя, що призводить до більш рівномірної яскравості та кольору. Ще одна важлива особливість — наявність на зображенні основних частин обличчя, таких як ніс, рот, очі. Щоб ідентифікувати людей, вони часто різко зменшують область зображення, де ймовірно знаходиться людина, або будують вертикальну гістограму. Ці методи

прості у застосуванні, але їх ефективність обмежена, коли на задньому плані є велика кількість нерелевантних об'єктів, у кадрі кілька людей або змінюється ракурс.

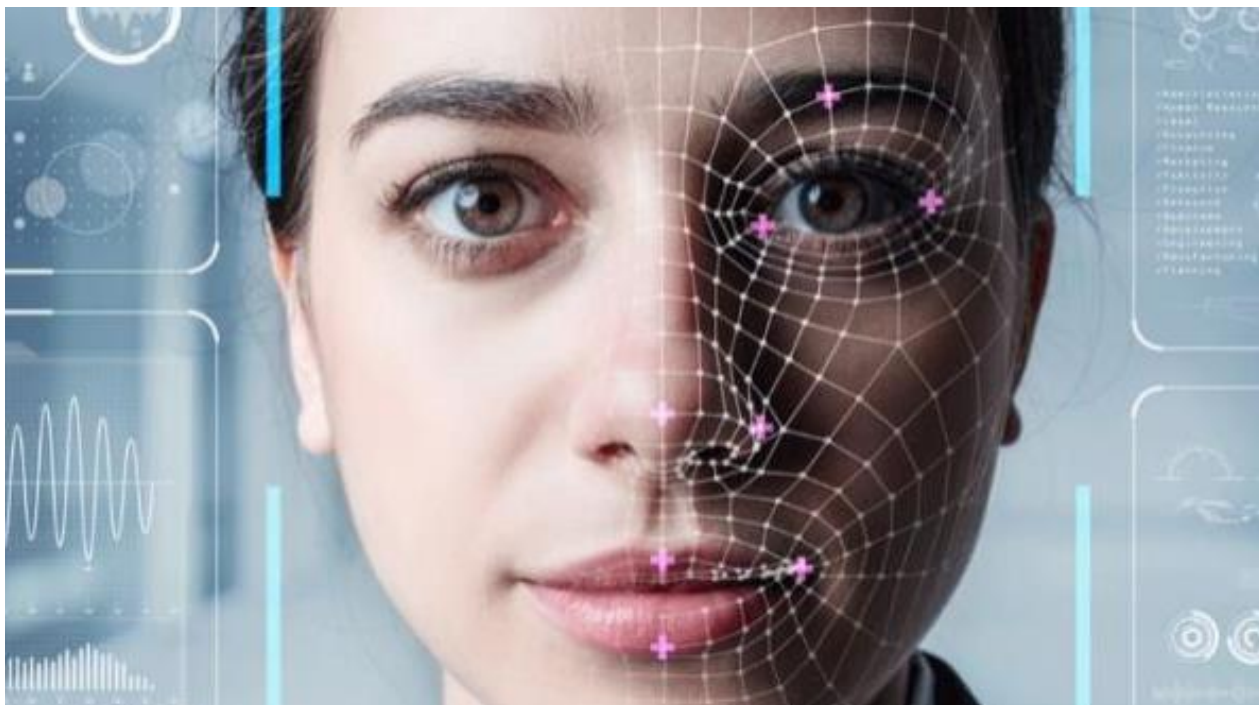


Рисунок 1.8 — Емпіричний метод розпізнавання обличчя

Наступний підхід до розпізнавання осіб заснований на інваріантних ознаках, властивих зображенням обличчя. Як і в попередньому підході, тут використовується емпіричний підхід, за допомогою якого система намагається «мислити» як людина. Цей метод спрямований на виявлення характерних частин людського обличчя, таких як контури, зміни форми та контрасту, поєднання цих ознак і проведення перевірки. Цей метод добре працює, навіть коли голова повертається, але розпізнавання ускладнюється, коли присутні інші люди або на неоднорідному фоні.

Інший підхід полягає у використанні шаблону, встановленого розробником, для виявлення людей. Людина представлена певним шаблоном або критерієм, і завдання алгоритму — перевірити кожен сегмент на наявність цього шаблону. При цьому можна проводити перевірки під різними кутами та масштабами, але така система вимагає багато розрахунків.

Усі сучасні технології розпізнавання облич використовують системи, які вивчають тестові зображення. Для цього використовується база даних із зображеннями людей і без них, як показано на малюнку 1.9. Кожен сегмент досліджуваного зображення характеризується вектором ознак, який допомагає класифікатору (алгоритм, що використовується для ідентифікації об'єктів у кадрі) визначити, чи є ця частина зображення обличчям.



Рисунок 1.9 — База даних зображень для розпізнавання облич

Технічно системи розпізнавання облич можуть сильно відрізнятися одна від одної, але вони мають загальні принципи роботи. Хоча методи відрізняються, загальну структуру процесу розпізнавання обличчя можна визначити, як показано на малюнку 1.10 [11].

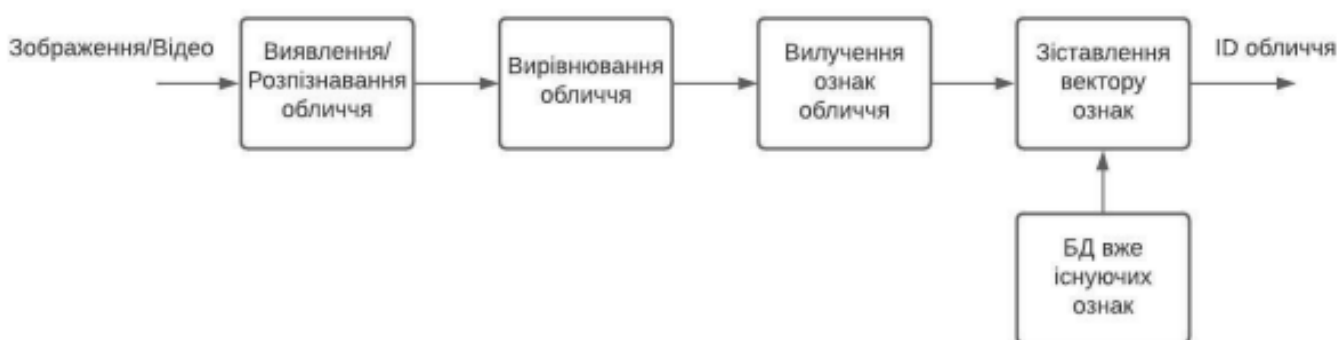


Рисунок 1.10 — Загальна структура процесу розпізнавання облич

Послідовність дій системи розпізнавання обличчя:

- система зчитує зображення або кадри з відеопотоку;
- виявлення та розпізнавання обличчя;
- розташуйте обличчя та визначте його розмір;
- вирівнювання обличчя;
- прибрати риси обличчя;
- витягніть характеристики із зареєстрованої бази даних облич;
- порівняння векторів рис обличчя з рисами обличчя в базі даних;
- отримайте ідентифікатор обличчя;

На першому етапі люди на зображенні виявляються та локалізуються. Під час фази розпізнавання виконується вирівнювання зображення обличчя (геометрично та за яскравістю), розрахунок ознак і пряме розпізнавання, яке передбачає порівняння обчислених ознак зі стандартами в базі даних..

1.4.1 Метод гнучкого порівняння на графах.

Суть цього методу полягає в еластичному узгодженні графів, що описують зображення обличчя. Грані представлені у вигляді графів зі зваженими вершинами та ребрами. Під час процесу розпізнавання одне з еталонних зображень залишається незмінним, тоді як інше зображення деформується для досягнення найкращого збігу з першим еталонним зображенням. У таких системах розпізнавання графіка може представляти прямокутні сітки та структури, створені на основі рис обличчя та антропометричних точок (рис. 1.11).

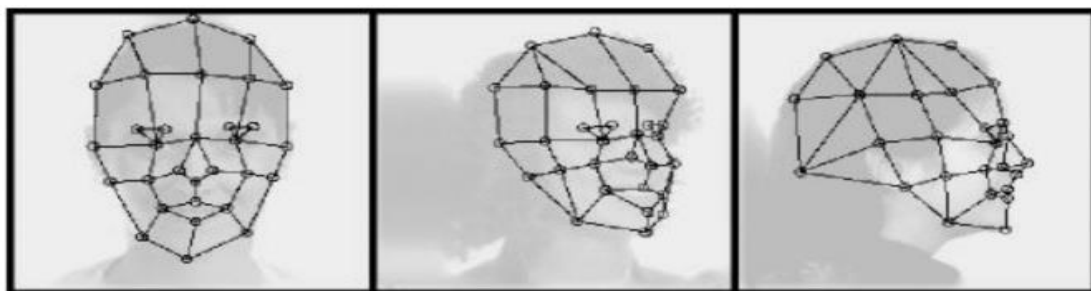


Рисунок 1.11 — Граф на основі антропометричних точок обличчя

У Для обчислення власних значень у вершинах графа зазвичай використовуються фільтри Габора або їх упорядковані набори комплексних значень — вейвлети Габора (ряди Габора). Ці значення обчислюються в локальній області у верхній частині графіка шляхом згортання значення яскравості пікселя за допомогою фільтра Габора.

Ребра графа отримують ваги, що відповідають відстані між сусідніми вершинами. Різниця (відстань або дискримінантна ознака) між двома графами обчислюється за допомогою функції ціни деформації, яка враховує як різницю в значеннях ознак, отриманих у вершинах, так і ступінь деформації на краях графа. Деформація графа відбувається шляхом переміщення кожної його вершини на певну відстань у певних напрямках від початкового положення. Виберіть місце, де різниця між власними значеннями (відповідь фільтра Габора) у деформованих вершинах графа та відповідних вершинах еталонного графа мінімальна. Алгоритм гнучкого методу порівняння наведено на рисунку 1.12.



Рисунок 1.12 — Метод гнучкого порівняння на графах

Деякі публікації стверджують, що точність розпізнавання за допомогою методів зіставлення еластичних графів може досягати 95-97% навіть при зміні виразу обличчя і куті огляду до 15 градусів. Однак розробники цього підходу стурбовані його високою обчислювальною вартістю. Наприклад, порівняння зображення обличчя з 87 тестами займає близько 25 секунд на паралельному комп'ютері з 23 трансп'ютерами.

До недоліків цього підходу можна віднести високу обчислювальну складність процесу розпізнавання, низьку технологічність при додаванні нових тестів, лінійну залежність часу виконання від розміру бази даних обличчя.

1.4.2 Метод нейронних мереж

В даний час існує близько десяти типів нейронних мереж (НМ). Найпоширенішим є багат шарова мережа на основі персептрона, яка дозволяє класифікувати вхідні зображення або сигнали на основі попередньої конфігурації та навчання мережі. Навчання нейронної мережі здійснюється на основі набору навчальних прикладів. Його суть полягає у використанні задачі оптимізації, вирішеної методом градієнтного спуску, для коригування вагових коефіцієнтів між нейронами. Під час навчання НМ автоматично виділяє ключові ознаки, визначає їх важливість і встановлює зв'язки між ними.

Згідно з результатами аналізу видання, найкращі результати для розпізнавання облич демонструють згорточні нейронні мережі (CNN), як показано на малюнках 1.13 і 1.14. Це логічний розвиток архітектур НМ, таких як когнітивні машини та нові когнітивні машини. Успіх CNN пояснюється його здатністю враховувати двовимірну топологію зображення, що відрізняє його від багат шарових персептронів.

Основними особливостями згорткової нейронної мережі (CNN) є локальні рецептивні поля, які забезпечують двовимірні зв'язки локальних нейронів, універсальні ваги, що дозволяють ідентифікувати певні ознаки в будь-якій частині зображення, а також ієрархічна структура з просторовою вибіркою. Ці

інновації надають ZNM часткову стійкість до масштабу, зсуву, обертання, зміни перспективи та інших змін спотворення.

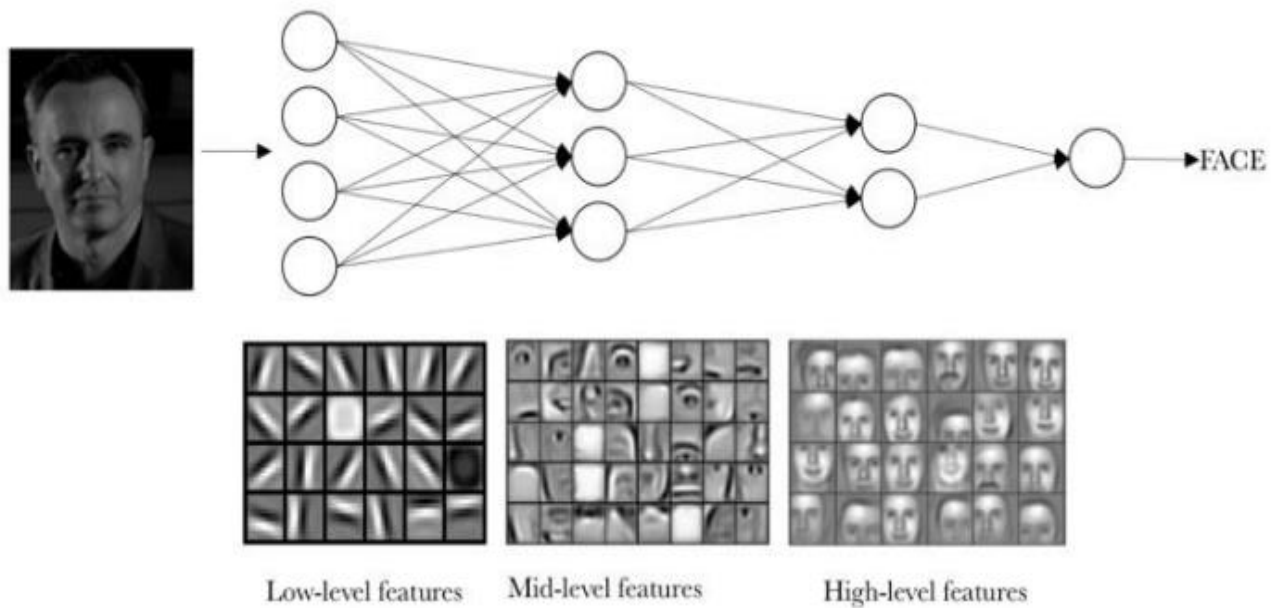


Рисунок 1.13 — Метод згорткової нейронної мережі(ЗНМ)

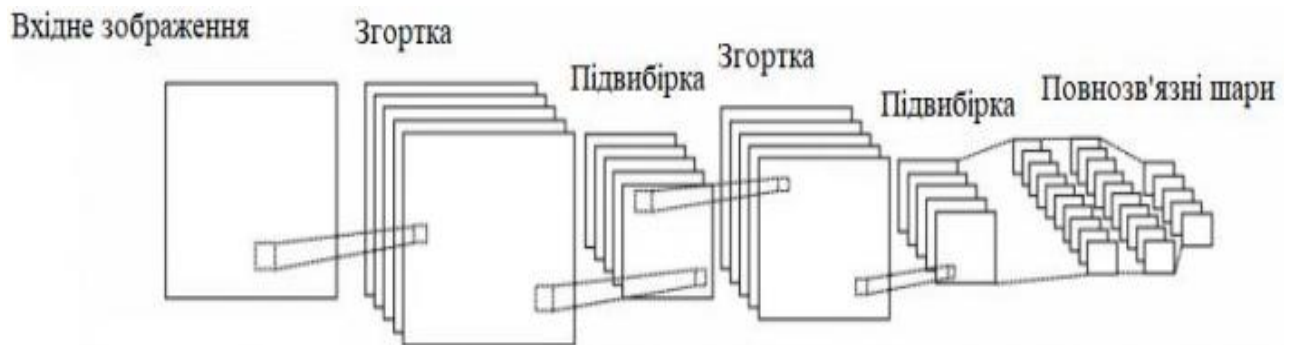


Рисунок 1.14 — Схематичне зображення архітектури згорткової нейронної мережі

Тести згорткової нейронної мережі (CNN) на базі даних ORL, яка містить зображення людей з незначними варіаціями в освітленні, масштабі, просторовому обертанні, положенні та різноманітних емоціях, показали точність розпізнавання 96%. Алгоритм розпізнавання обличчя на основі згорткової нейронної мережі показано на рисунку 1.15.

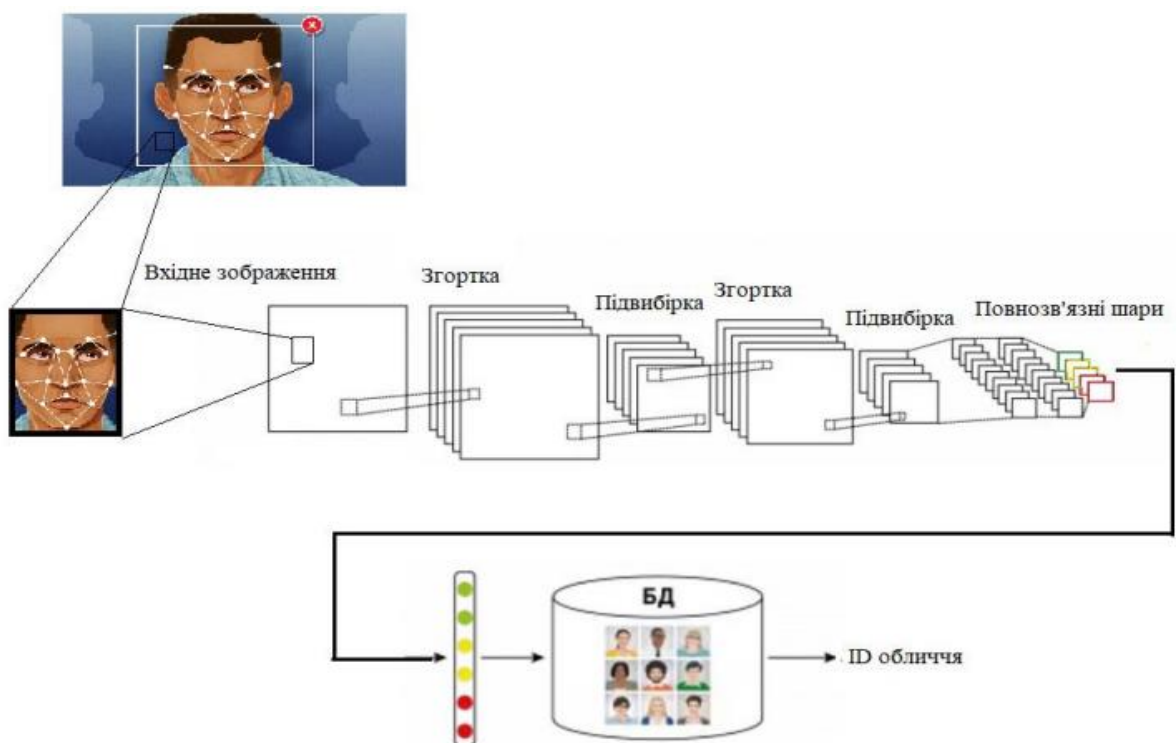


Рисунок 1.15 — Метод розпізнавання на основі загорткової нейронної мережі

Недоліки: додавання нових посилань в базу даних вимагає повного перенавчання мережі на всьому існуючому наборі даних, що є досить трудомістким процесом — від години до кількох днів, залежно від розміру вибірки. Існують також математичні проблеми, пов'язані з навчанням, такі як ризик досягнення локальних оптимумів, вибір оптимального кроку для оптимізації та перенавчання моделі.

1.4.3 Сховані Марківські моделі (СММ, НММ)

Одним із статистичних методів розпізнавання обличчя є прихована модель Маркова з дискретним часом (НММ), як показано на малюнку 1.16. СММ використовує статистичні властивості сигналу та враховує його просторові характеристики. Елементами моделі є: набір прихованих станів, набір спостережуваних станів, матриця ймовірностей переходу та ймовірність початкового стану. Кожному з цих елементів відповідає своя марковська модель.

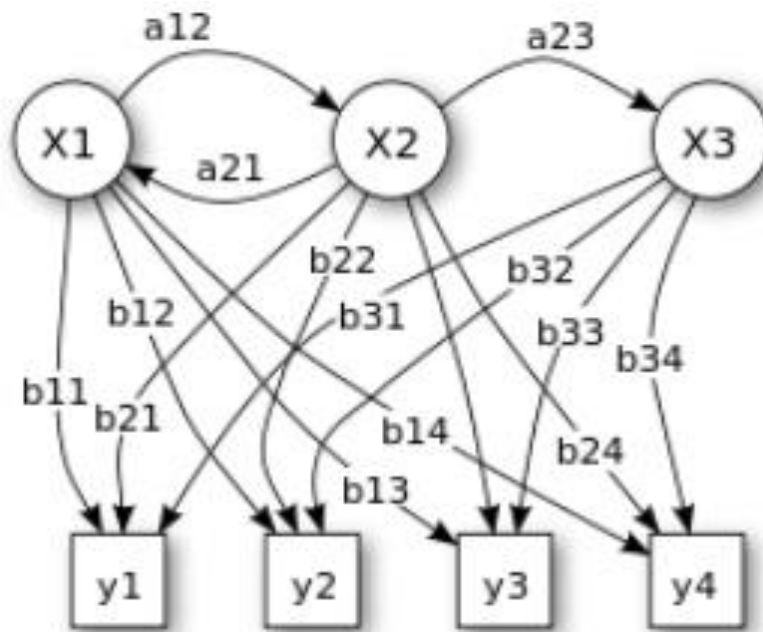


Рисунок 1.16 — Принцип роботи прихованої марківської моделі

Коли об'єкт розпізнається, досліджуються моделі Маркова, створені для даної бібліотеки об'єктів, а також визначається максимальна ймовірність того, що послідовність спостережень певного об'єкта буде згенерована відповідною моделлю.

Але є певні недоліки: по-перше, параметри моделі потрібно налаштувати для кожної бази даних. По-друге, SMM не має розрізнявальної сили, оскільки алгоритм навчання лише максимізує реакцію кожного зображення на власну модель і не мінімізує реакцію на інші моделі. На сьогодні не знайдено жодного комерційного застосування SMM для ідентифікації особистості.

1.4.4 Метод головних компонентів або principal component analysis (PCA)

Одним із найбільш відомих і добре зарекомендував себе методів є метод головних компонент (PCA), який базується на перетворенні Карунена-Лойова. Спочатку цей метод використовувався в статистиці для зменшення розмірності ознак без істотної втрати інформації. У задачах розпізнавання обличчя він в основному використовується для представлення зображень обличчя у вигляді векторів малої розмірності (головних компонентів), а потім їх порівняння з

опорними векторами, що зберігаються в базі даних. Основна мета методу головних компонентів (РСА) полягає в тому, щоб значно зменшити розмірність простору ознак, щоб точно представити «типові» зображення, що належать різним особам. Застосування цього методу дозволяє виявляти різні зміни в навчальному наборі зображень обличчя та представляти ці зміни за допомогою кількох ортогональних векторів, які називаються власними гранями. Принцип роботи цього методу показано на малюнку 1.17.

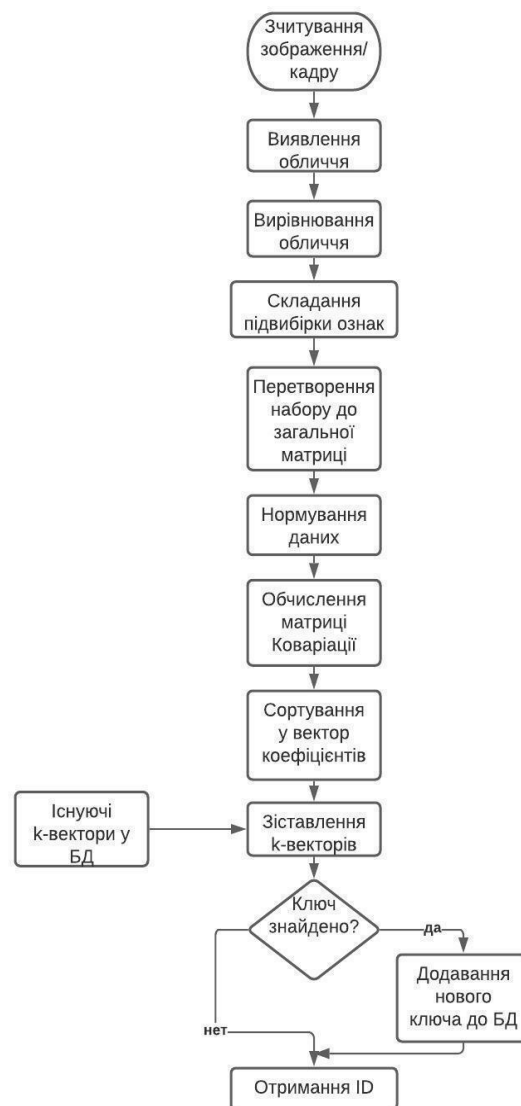


Рисунок 1.17 — Принцип розпізнавання облич методу головних компонентів

Метод головного компонента працює наступним чином: спочатку всі зображення в навчальному наборі об'єднуються в єдину матрицю даних. Кожен

рядок цієї матриці відповідає індивідуальному зображенню людини, представленому у вигляді одновимірного вектора. Усі зображення в навчальному наборі повинні бути масштабовані до однакових розмірів і мати нормалізовані гістограми.



Рисунок 1.18 — Перетворення навчального набору облич на одну загальну матрицю X

Після цього дані нормалізуються, зменшуючи рядки матриці до нульового середнього та одиничної дисперсії. Далі розраховуємо коваріаційну матрицю. Для цієї матриці вирішується задача визначення власних значень і відповідних їм власних векторів (власних граней). Потім вектори ознак сортуються в порядку спадання їхніх значень, і лише k верхніх векторів зберігаються відповідно до встановлених критеріїв.

Порівняйте отримані показники з контрольним показником у базі даних на діаграмі, як показано на малюнку 1.19.

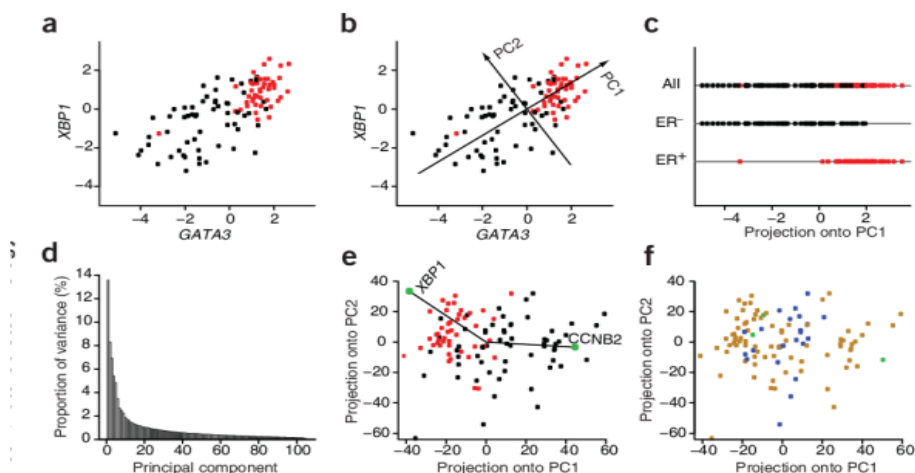


Рисунок 1.19 — Зіставлення отриманих результатів обчислень з еталонами БД

Недоліки: Метод головних компонентів (РСА) довів свою хорошу ефективність у практичних застосуваннях. Однак його ефективність значно знижується, якщо на зображеннях людей спостерігаються значні зміни в освітленні або виразі обличчя. Це пояснюється тим, що РСА вибирає підпростір, який найкраще наближає набір вхідних даних, а не розрізняє різні класи осіб..

1.5 Порівняння методів детектування обличчя

Вирівнювання зображення є обов'язковим попереднім етапом майже у всіх алгоритмах розпізнавання обличчя. Це означає переведення зображень обличчя у фронтальне положення відносно камери або переведення зразків цілих людей (наприклад, навчальних зразків) до єдиної системи координат. Для цього необхідно знайти на зображенні характерні антропометричні точки, загальні для всіх облич. Найчастіше це центр зіниці або куточки ока. Різні дослідники використовують різні набори таких точок, але для зменшення витрат на обчислення, особливо в системах реального часу, зазвичай призначається трохи більше 10 таких точок.

Щоб оцінити ефективність алгоритмів розпізнавання облич, DARPA та Дослідницька лабораторія армії США розробили програму FERET (Facial Recognition Technology). У тестах FERET брали участь алгоритми, засновані на гнучкому порівнянні графів і різні модифікації методу головних компонент (РСА). Виявляється, що всі алгоритми демонструють приблизно однакову ефективність, що ускладнює чітке розмежування між ними.

Для фронтальних зображень точність розпізнавання зазвичай становить 95%. Однак точність впала до 80% для зображень, зроблених іншою камерою або за інших умов освітлення. Для зображень, зроблених за рік, точність розпізнавання становить близько 50%, що є прийнятним результатом для такого типу системи.

2 ОБ'ЄКТИ ДОСЛІДЖЕННЯ ТА УДОСКОНАЛЕННЯ МЕТОДУ СПОСТЕРЕЖЕННЯ ЗА ОЧИМА ВОДІЯ

2.1 Цифрове зображення, методи їх обробки та основи комп'ютерного зору

Цифрове (електронне) зображення — це двовимірний масив пікселів розміром $M \times N$ (рис. 2.1).

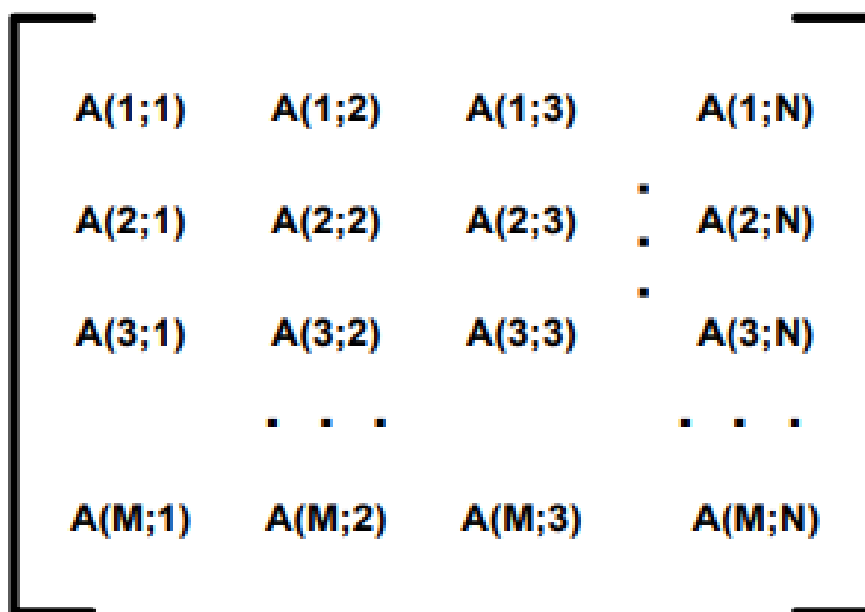


Рисунок 2.1 — Масив пікселів $M \times N$

До основних характеристик електронних зображень, крім розміру, відносяться кількість каналів і глибина кольору. Кількість каналів визначає, скільки кольорів використовується для створення пікселя. Наприклад, кольорове зображення має три канали, кожен з яких відповідає інтенсивності одного з трьох основних кольорів: червоного, зеленого та синього. Монохромні зображення, відтворені в градаціях сірого, є одноканальними. Глибина кольору визначає інтенсивність, характеризує кількість відтінків, доступних для кожного кольору, і визначається кількістю бітів. Кількість можливих відтінків розраховується так:

$$D=2^N-1 \quad (2.1)$$

де D — кількість відтінків кольору,

N — глибина кольору.

У даній роботі обробляються одно- та триканальні зображення з глибиною 8 біт. Отже, згідно з формулою (2.1.), кожен піксель зображення є комбінацією чисел у діапазоні від 0 до 255 для кожного каналу.

Основою алгоритмів обробки зображень є переважно інтегральне перетворення, таке як мішок, перетворення Фур'є тощо. Також широко використовуються статистичні методи. Методи обробки зображень часто класифікують на основі кількості пікселів, що використовуються на одному етапі трансформації. Дано деяке двовимірне зображення розміром $M \times N$, де $a(M; N)$ – точки вихідного зображення, а $a^*(M; N)$ – модифіковані точки вихідного масиву.

Існує в основному три методи обробки зображень:

- точковий метод: під час його застосування значення в точці $a(M; N)$ зміниться на $a^*(M; N)$ незалежно від значень суміжних точок;
- локальний метод: використовуйте значення сусідніх точок навколо $a(M; N)$, щоб обчислити значення $a^*(M; N)$;
- глобальний метод: значення $a^*(M; N)$ визначається на основі всіх значень вихідного зображення A [12].

Як наука ця дисципліна тісно пов'язана з проблемами штучного інтелекту та нейронних мереж, тобто фактично є інструментом, який дозволяє обчислювальним пристроям самостійно робити висновки про навколишнє середовище. Мета комп'ютерного зору полягає в тому, щоб робити корисні висновки про реальні об'єкти та сцени шляхом аналізу зображень, зібраних датчиками. Кожен піксель кольорового зображення — це набір із трьох цілих чисел, які визначають інтенсивність основного кольору. Тому зображення розглядаються комп'ютером як матриця цілих чисел (рис. 2.2)[13].

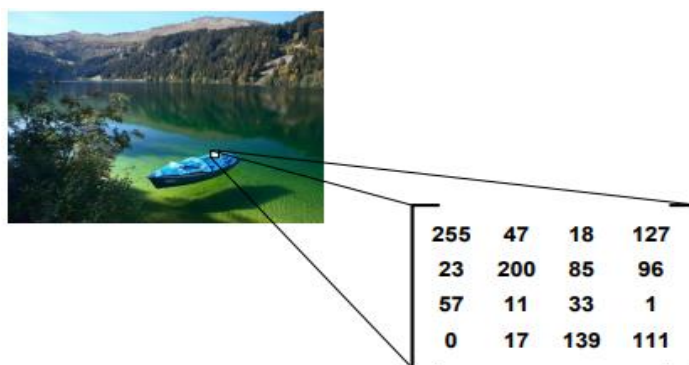


Рисунок 2.2 — Сприйняття зображення комп'ютером

2.2 Методи розпізнавання об'єктів у цифрових зображеннях

Основною функцією зазначеного пристрою є кадрове розпізнавання об'єктів у відеопотоці, зокрема облич і очей. На сьогодні основним інструментом виявлення облич та їх елементів на зображеннях у реальному часі є алгоритм Віоли-Джонса [14].

Цифрове зображення — це двовимірний масив пікселів розміром $M \times N$, де кожен піксель представлений значенням від 0 до 255 для кожного колірного каналу. Пошук облич і їх ознак виконується в активній області зображення за допомогою прямокутних ознак, що описують знайдене обличчя, і повертає межі прямокутної області, де може знаходитися об'єкт пошуку, у форматі:

$$R=\{x,y,w,h,a\}; \quad (2.2)$$

де x, y — координати центру прямокутника;

w, h — ширина та висота;

a — кут нахилу відносно осі ординат.

Процес розпізнавання об'єктів у реальному часі вимагає високої швидкості обчислень, тому вхідне зображення подається у вигляді цілісного представлення. Це представлення дозволяє швидко обчислити загальну інтенсивність пікселів у довільному прямокутнику, причому розмір

прямокутника не впливає на час обчислення. Інтегральне представлення можна уявити як матрицю, розмір якої узгоджується з розміром вихідного зображення, а обчислення кожного елемента відбувається наступним чином:

$$H(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(x, y) \quad (2.3)$$

де $I(x, y)$ — інтенсивність вихідного пікселя.

Кожен елемент обчисленого двовимірного масиву $H(x, y)$ показує загальну інтенсивність пікселів у прямокутній області, обмеженій координатами $(0, 0)$ і (x, y) . Узагальнення яскравості пікселів у певній області зазвичай вимагає часу, пропорційного розміру масиву, але в інтегральному представленні це вимагає лише 3 арифметичних операцій і 4 звернень до пам'яті, що значно покращує швидкість обчислення ознак Хаара під час розпізнавання. Припустимо, нам потрібно обчислити загальну інтенсивність прямокутної області ABCD, зображеної на малюнку 2.3.

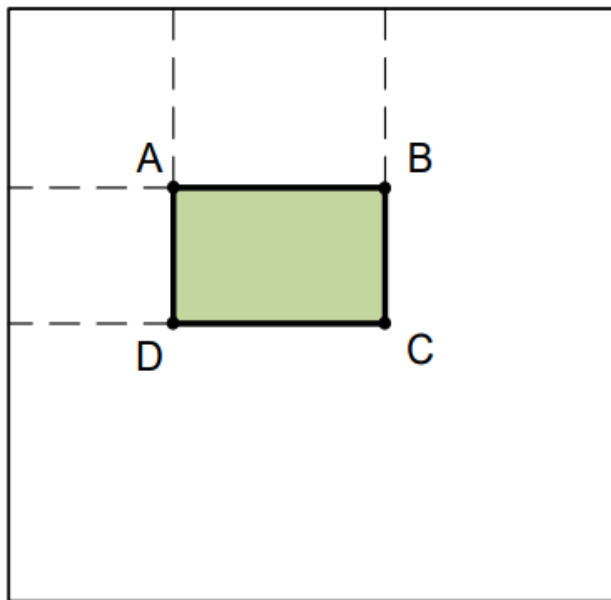


Рисунок 2.3 — Розрахунок сумарної інтенсивності області у інтегральному представленні

Суму можна виразити як суму та різницю суміжних прямокутних площ,

тому наступний вираз є дійсним[15]:

$$I_{ABCD}(x,y) = I_A(x,y) + I_C(x,y) - I_B(x,y) - I_D(x,y) \quad (2.4)$$

Завдання розпізнавання образів у цифрових зображеннях базуються на використанні спеціальних ознак, які називаються ознаками Хаара. Знак Хаара являє собою прямокутну структуру, що складається з двох або більше чорних і білих ділянок (рис. 2.4).

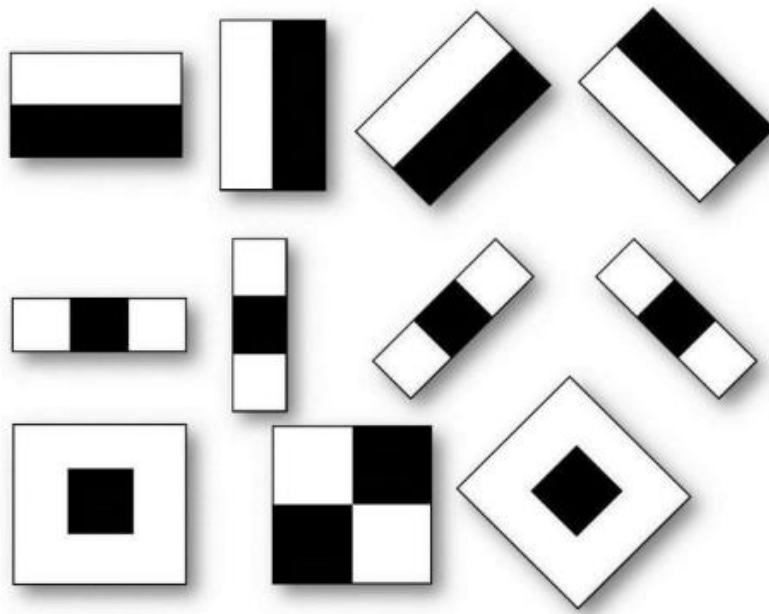


Рисунок 2.4 — Ознаки Хаара

Кожна відтворена функція розташована на вхідному зображенні, а інтенсивність пікселів у відповідних областях потім підсумовується. Ми виражаємо суму значень інтенсивності пікселів у вихідному зображенні, які накладаються на яскраву частину об'єкта Хаара, як Π_{LIGHT} , а суму значень інтенсивності пікселів, які накладаються на темну частину, як Π_{DARK} . Тоді різниця $F = \Pi_{\text{LIGHT}} - \Pi_{\text{DARK}}$ є розрахованим власним значенням [16, 17]. Спочатку метод Віюлі-Джонса використовував лише ортогональні примітиви, але в міру розвитку методу були запропоновані примітиви, нахилені на 45 градусів відносно координатних осей, а також елементи з асиметричною конфігурацією.

Крім того, під час обчислення власних значень замість простої різниці пропонується зважена сума пікселів темної та світлої областей:

$$f = \sum_{i=1..N} \omega_i \cdot \Pi_i \quad (2.5)$$

де ω — ваговий коефіцієнт,

Π — сумарна інтенсивність області [18].

2.3 Аналіз методів детектування стану обличчя бібліотеки dlib

Класифікація станів очей у запропонованому вдосконаленні існуючих алгоритмів виконана за допомогою методів комп'ютерного зору бібліотеки мови програмування Python «dlib».

Dlib — це кросплатформна універсальна бібліотека, написана мовою C++. Він був створений під впливом контрактного дизайну та ідей компонентно-орієнтованого програмування, тому є набором незалежних програмних компонентів. Dlib — це програмне забезпечення з відкритим кодом, яке розповсюджується за ліцензією Boost Software. Розробка бібліотеки почалася в 2002 році і містить великий набір інструментів. Станом на 2016 рік він включає такі компоненти, як комп'ютерні мережі, потокове передавання, графічні інтерфейси користувача, структури даних, лінійна алгебра, машинне навчання, обробка зображень, інтелектуальний аналіз даних, аналіз XML і тексту, чисельна оптимізація, мережі Байєса тощо. В останні роки вона в основному була зосереджена на розробці інструментів машинного навчання. У 2009 році бібліотека Dlib була опублікована в «Journal of Machine Learning Research».

DLib — це бібліотека C++ з відкритим вихідним кодом, яка реалізує різноманітні алгоритми машинного навчання, включаючи класифікацію, регресію, кластеризацію, перетворення даних і структуроване передбачення. Він також підтримує багатопотоковість і роботу в мережі. Бібліотека DLib-ml

реалізує багато алгоритмів машинного навчання. Метод виявлення очей у dlib показано на рисунку 2.5.

DLib містить розширені модульні тести та приклади використання бібліотеки, а також надає детальну документацію для всіх класів і функцій, яку можна знайти на офіційному веб-сайті бібліотеки. DLib надає потужну структуру для розробки програм машинного навчання на C++.

Бібліотека DLib подібна до DMTL тим, що вона також надає високопродуктивний інструментарій машинного навчання з різними алгоритмами. Однак DLib регулярно оновлюється, щоб включити більше прикладів і ширшу підтримку функціональності.

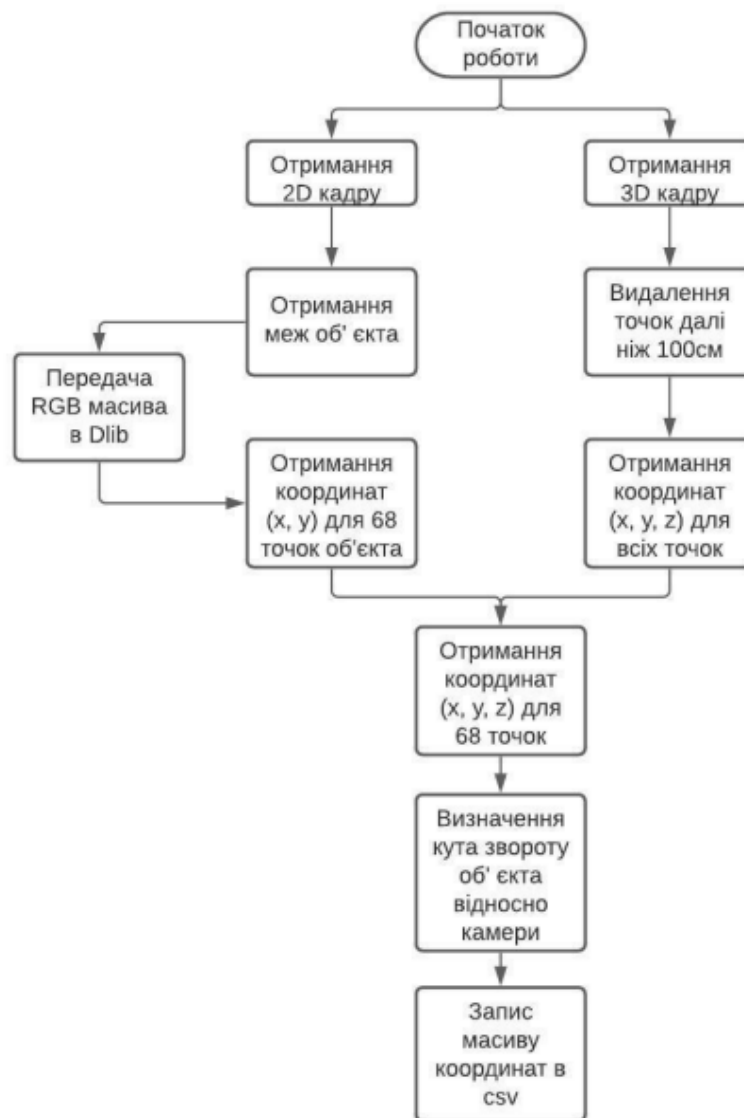


Рисунок 2.5 — Метод розпізнавання облич dlib

Особливістю DLib є те, що він підходить як для наукових досліджень, так і для створення програм машинного навчання на C++ і Python [20].

Порівняно з OpenCV, DLib більш точний у виявленні облич на зображеннях, оскільки він має більше моделей розпізнавання та може розпізнавати 68 або більше рис обличчя [21]. Особисті орієнтири, створені dlib, представлені у формі списку індексів, як показано на рисунку 2.6.

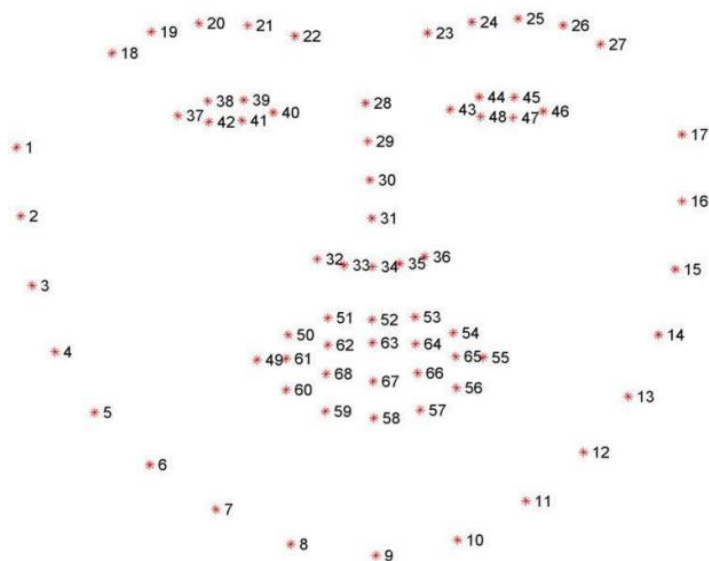


Рисунок 2.6 — Візуалізація ознак обличчя

Тому, щоб виділити область очей із набору рис обличчя, необхідно знати точний індекс зрізу в масиві. За допомогою цих індексів ви можете легко вибрати області очей за допомогою нарізки списку.

2.4 Побудова удосконаленого методу системи детектування очей водія

Спершу Спочатку створіть екземпляр VideoStream, який використовує індекс інфрачервоної камери для захоплення відео. Ви також повинні зробити паузу на секунду, щоб датчик камери увімкнувся. У програмному циклі кадри відеопотоку зчитуються один за одним і попередньо обробляються: розмір кожного кадру змінюється до 450 пікселів у ширину та перетворюється на відтінки сірого. Детектор обличчя dlib використовується для пошуку та визначення розташування облич на зображеннях.

Наступним кроком є застосування функції розпізнавання рис обличчя, щоб визначити місцезнаходження всіх ключових областей обличчя. Кожне виявлене обличчя обробляється в циклі (хоча припускається лише одне обличчя — водій, цикл дозволяє обробляти відео потенційно кількох людей). Для кожного виявленого обличчя застосуйте детектор обличчя `dlib` і перетворіть результат у масив `numpy`. Використовуючи фрагмент масиву, ви можете отримати координати (x, y) лівого та правого ока. Знаючи координати обох очей, легко розрахувати їх співвідношення сторін і для точності усереднити значення. Потім ви можете використати функцію `cv2.drawContours`, щоб відобразити кожную область очей у кадрі, що часто корисно для перевірки того, що очі виявлені та розташовані правильно.

На цьому етапі система готова перевірити, чи відображає водій симптоми сонливості у відеопотоці. Спочатку переконайтеся, що співвідношення сторін ока нижче порогу «моргання/закриття». Якщо це так, тоді збільште лічильник, підрахувавши кількість послідовних кадрів, у яких водій закрив очі. Якщо значення лічильника перевищує встановлену максимальну кількість кадрів, драйвер вважається сплячим.

Потім він перевіряє, чи увімкнено зумер — якщо ні, зумер увімкнено. Потім відтворіть звуковий сигнал, для цього створіть окремий потік, відповідальний за звук, щоб основна програма не блокувала під час відтворення звуку.

Після цього на екрані з'явиться текст "Sleepy Alert!" («Попередження про сон»).

Далі обробіть випадок, коли співвідношення сторін ока перевищує порогове значення, тобто око відкрите. Якщо очі відкриті, лічильник скидається, а звукове повідомлення не відтворюється.

Співвідношення сторін ока розраховується як відношення відстані між вертикальними рисами очей до відстані між горизонтальними рисами очей. Ці відстані визначаються шляхом обчислення евклідової відстані між двома наборами точок на оці.

Евклідова відстань — це геометрична відстань у багатовимірному просторі [22].

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.6)$$

Евклідова відстань обчислюється з необроблених даних (ненормалізованих даних). Це дозволяє використовувати незалежні змінні в рівнянні регресії як основні детермінанти багатовимірного простору, в якому можна помістити кожне спостереження.

Набір ізометрій в евклідовому просторі.

У математиці евклідова група — це група (ізометрія евклідового простору E), тобто перетворення цього простору, яке зберігає евклідову відстань між будь-якими двома точками. Ця група залежить від розмірності n простору, зазвичай виражається як $E(n)$ або $ISO(n)$ [23].

Евклідова група $E(n)$ включає всі можливі трансляції, повороти, відображення точок у просторі E та будь-яку кінцеву комбінацію цих перетворень. Ця група вважається групою симетрії простору, а також включає групу симетрії довільних фігур (підмножин) цього простору.

Евклідова група є не тільки топологічною групою, а й групою Лі, тому поняття, які використовуються в численні, можна легко адаптувати до цієї структури. Група $E(n)$ є підгрупою афінної групи міри n і зберігає напівпрямую товарну структуру обох груп. Тому елементи групи можна записати двома способами:

— за парами A, b , де A — ортогональна матриця розміру $n \times n$, a , a вектор-стовпчик розміру n ;

— квадратною матрицею розміру $n + 1$, як описано в афінній групі.

Відповідно до Ерлангенської програми Фелікса Кляйна, евклідова геометрія розглядається як геометрія симетрій евклідової групи, яка є

спеціалізацією афінної геометрії. Усі афінні теореми все ще застосовуються. Евклідова геометрія дозволяє визначити поняття відстані, а потім вивести з нього поняття кута. Ці групи є одними з найстаріших і найбільш вивчених груп, особливо у 2 і 3 вимірах, задовго до того, як було винайдено поняття групи.

Це означає, що кожен існуючий алгоритм використовує базу даних із великою кількістю інформації для порівняння рис обличчя, таким чином знижуючи продуктивність. Однак запропоноване вдосконалення алгоритму виявлення очей не використовує базу даних для порівняння ознак, а лише застосовує простий індексований список із 68 орієнтирів обличчя з бібліотеки DLib і евклідове обчислення, перетворене на матрицю (масив NumPy). Це дозволяє швидко розрахувати порогові значення, які визначають стан очей водія.

Алгоритм запропонованого методу стеження за очима водія показаний на рисунку 2.7.

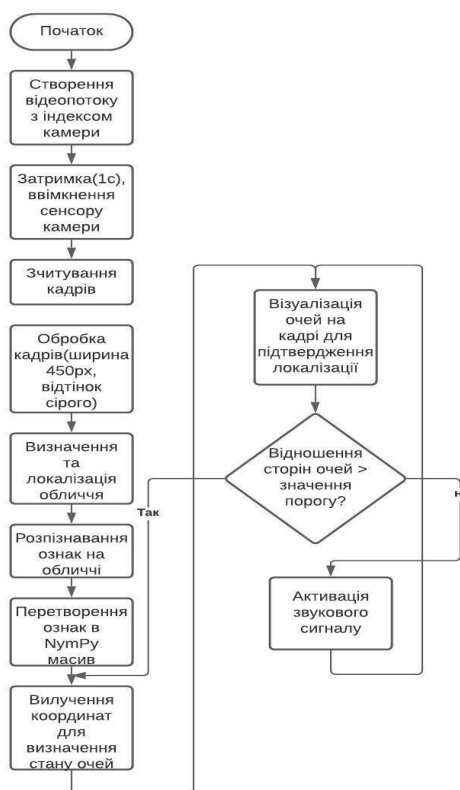


Рисунок 2.7 — Алгоритм роботи удосконаленого методу спостереження за очима

3 РЕАЛІЗАЦІЯ МЕТОДУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ КОНТРОЛЮ СТАНУ ВОДІЯ

3.1 Аналіз мікропроцесорних платформ

Пристрій Пристрій є прототипом, побудованим на загальній портативній апаратній платформі, що забезпечує мінімально необхідний функціонал. Для вибору апаратної платформи було проведено порівняльний аналіз доступних міні-комп'ютерів і апаратних обчислювальних платформ.

Метод комп'ютерного зору, який планується використовувати в цій роботі, передбачає своєчасну обробку двовимірних масивів, що є досить ресурсомістким процесом з точки зору обчислювальної потужності пристрою. Тому для вирішення цих завдань ключовими характеристиками платформи є: тактова частота процесора, обсяг оперативної та енергонезалежної пам'яті, наявність портів і пристроїв введення та виведення, підтримувані інтерфейси, вартість.

Виберіть найпопулярніші платформи для аналізу, зокрема BeagleBone Black, Intel Galileo та Raspberry Pi 2.

BeagleBone Black (рисунок 3.1) є останнім членом сімейства BeagleBoard, розробленим для спільноти розробників з відкритим кодом. Пристрій базується на процесорі ARM Cortex-A8 і має мінімально необхідний набір функцій, що дозволяє користувачам оцінити можливості процесора. У той же час BeagleBone Black не є зрілою платформою розробки через особливості вбудованих систем, деякі функції та інтерфейси процесора недоступні.

Це не готовий продукт для конкретного завдання, а платформа для експериментів і навчання програмуванню процесора та використання периферійних пристроїв. Пристрій забезпечує доступ до багатьох інтерфейсів і підтримує використання додаткових плат розширення, що дозволяє реалізувати різні комбінації функцій [24].

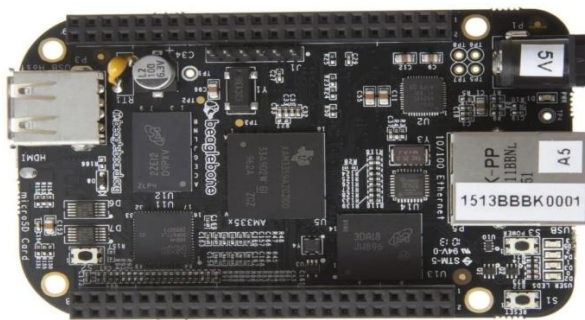


Рисунок 3.1 — Мікрокомп'ютер BeagleBone Black

Intel Galileo (рис. 3.2) — пристрій на базі мікропроцесора Intel Quark SoC X1000, 32-розрядна система на мікросхемі Intel Pentium класу. Це перший пристрій на архітектурі Intel, розроблений для повної апаратної та програмної сумісності з екранами Arduino.

Galileo повністю програмно сумісний із середовищем розробки Arduino IDE, що значно спрощує процес створення та налаштування програм. На додаток до сумісності з Arduino, пристрій підтримує низку стандартних інтерфейсів і портів введення/виведення, які зазвичай зустрічаються в ПК, включаючи: повнорозмірний слот mini-PCI Express, порт Ethernet (100 Мбіт/с), слот для роз'єму Micro-SD, послідовний порт RS-232, порти USB-хост і USB-клієнт, а також 8 МБ флеш-пам'яті NOR.

Така функціональність значно розширює сферу застосування Galileo порівняно зі стандартними платами Arduino, дозволяючи реалізувати більш складні та багатофункціональні проекти [25].

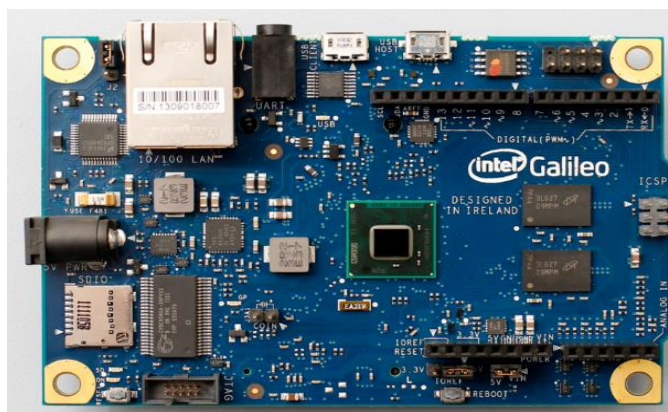


Рисунок 3.2 — Мікропроцесор Intel Galileo

Raspberry Pi 2 (рис. 3.3) — це одноплатний комп'ютер, який спочатку створювався як бюджетна система для навчання інформатики, але з тих пір використовується в багатьох галузях. Він заснований на процесорі Broadcom BCM2835 ARM11 з тактовою частотою 900 МГц і 1 ГБ оперативної пам'яті. Реалізовано за технологією «package packaging», яка інтегрує пам'ять безпосередньо в процесор.

У моделі «2» використовується процесор з чотирма ядрами Cortex-A7, що працюють на частоті 1 ГГц, і 1 ГБ оперативної пам'яті. Пристрій має чотири порти USB 2.0, порт Ethernet, графічне ядро з підтримкою OpenGL ES 2.0, апаратне прискорення, відео Full HD і ядро DSP для високопродуктивної обробки графіки та відео..



Рисунок 3.3 — Мікрокомп'ютер Raspberry Pi 2

За результатами порівняльного аналізу найкращим вибором платформи є Raspberry Pi 2. Основна відмінність Raspberry від багатьох більш просунутих одноплатних комп'ютерів полягає у високому рівні технічної підтримки, що відображається у високому рівні оптимізації дистрибутива програмного забезпечення, різноманітному наборі бібліотек із функціями та методами, які можна використовувати в оригінальному проекті, і Високий ступінь одноманітності апаратних компонентів.

Офіційною операційною системою для обраної апаратної платформи є варіант Debian — Raspbian. Окрім Raspbian, пристрій підтримує багато інших систем POSIX: Debian Wheezy, Ubuntu MATE, Fedora Remix [26].

3.2 Вибір електронних компонентів

Оскільки система повинна була забезпечити безперервний моніторинг обличчя людини, для виконання цієї функції була обрана відповідна камера, а також сенсорний дисплей, який дозволяв користувачеві налаштовувати параметри за потреби.

Для безперервного моніторингу облич і очей людини для виконання цієї функції була обрана камера Logitech C170 (рис. 3.4).



Рисунок 3.4 — Камера Logitech WebCam C170

Камера Logitech C170 підходить для систем відеозв'язку та відеоспостереження. Камера здатна знімати відео з роздільною здатністю до 1024×768 пікселів, що дозволяє отримувати якісні зображення для спілкування або запису.

Система кріплення дозволяє зручно закріпити камеру на моніторі або іншій поверхні. Підключення здійснюється через порт USB, що робить його сумісним з більшістю сучасних комп'ютерів і ноутбуків. Камера підтримує

автоматичне налаштування експозиції, що дозволяє адаптуватися до умов освітлення [27].

Крім того, для зручності користувача було обрано сенсорний РК-дисплей Raspberry Pi TFT від Waveshare для налаштування пристрою та зміни параметрів (рис. 3.5).

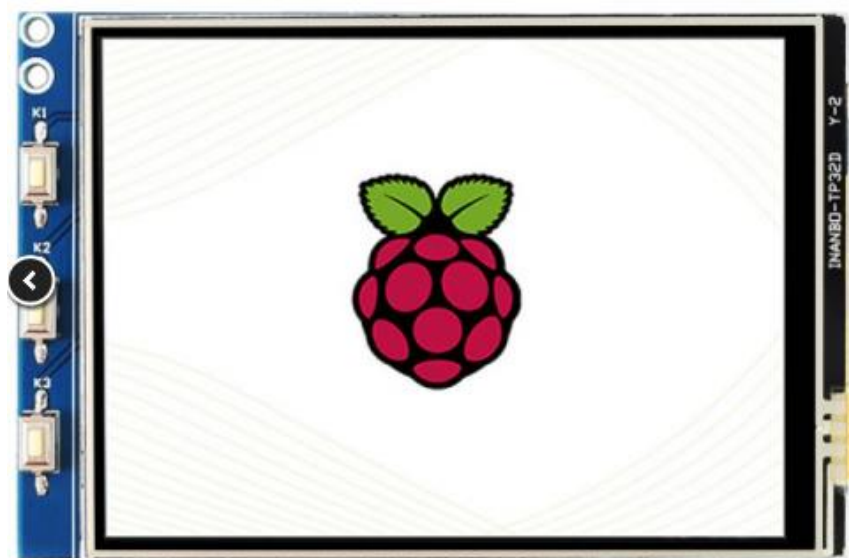


Рисунок 3.5 — Сенсорний дисплей TFT LCD для Raspberry Pi

Цей монітор призначений для підключення до міні-комп'ютера Raspberry Pi. Дисплей має роздільну здатність 320×240 пікселів і оснащений резистивним сенсорним екраном, який не вимагає додаткового обладнання для взаємодії з системою. Підтримуються всі версії Raspberry Pi, і доступні сумісні драйвери для операційних систем Raspbian і Ubuntu. Розмір дисплея ідеально підходить для Raspberry Pi, що забезпечує компактне та зручне розміщення. Крім того, на платі є три спеціальні кнопки, які можна запрограмувати відповідно до потреб користувача. Якісне покриття контактних рейок друкованої плати забезпечує надійність і довговічність пристрою[28].

3.3 Реалізація апаратного засобу

Зразок містить такі основні апаратні компоненти:

- мікрокомп'ютер Raspberry Pi 2;

- камера;
- сенсорний дисплей.

Камера, яка знімає зображення, підключена до наданої апаратної платформи за допомогою інтерфейсу USB 2.0. На етапі прототипу була обрана камера Logitech C170, здатна записувати відео з роздільною здатністю до 1024×768 пікселів.

Завдяки високому рівню підтримки та популярності інтерфейсу USB, для ініціалізації камери не потрібні додаткові операції чи налаштування програмного забезпечення.

Для демонстрації роботи пристрою 3,2-дюймовий сенсорний дисплей підключили до мікрокомп'ютера через інтерфейс SPI. Цей монітор потребує налаштування програмного забезпечення шляхом встановлення відповідних драйверів.

Принципова схема пристрою наведена на рисунку 3.6.

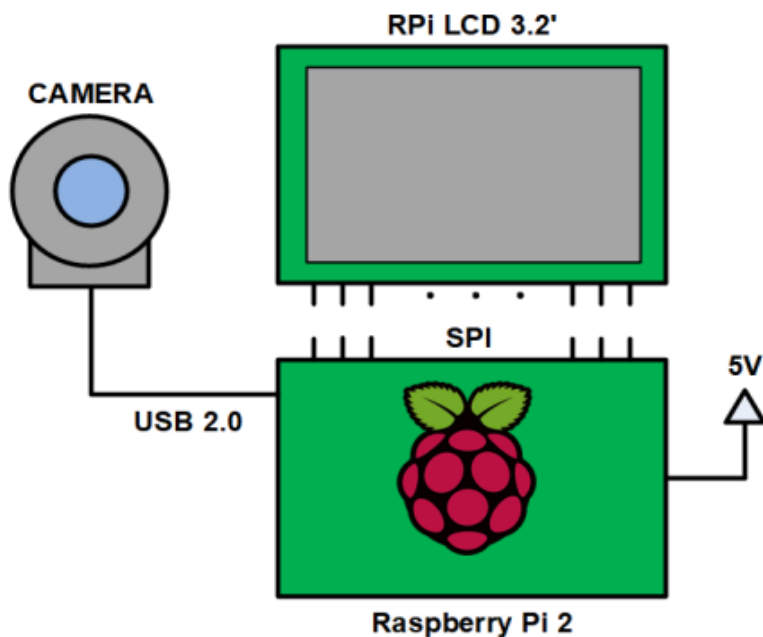


Рисунок 3.6 — Принципова схема пристрою

3.4 Мова програмування Python

Система Система реалізована за допомогою мови програмування Python (рисунок 3.7). Python — це високорівнева мова програмування загального

призначення з динамічною строгою типізацією та автоматичним керуванням пам'яттю. Він зосереджений на підвищенні продуктивності розробників, читабельності та якості коду, а також на забезпеченні переносимості створених програм. Python повністю об'єктно-орієнтований, усе є об'єктом. Однією з його унікальних особливостей є використання відступів для розділення блоків коду, що полегшує написання програм.

Синтаксис мови дуже мінімалістичний, тому розробники рідко звертаються до документації. Python — це інтерпретована мова, яка широко використовується для створення сценаріїв. Однак недоліки включають повільніше виконання програми та збільшене споживання пам'яті порівняно з програмами, написаними на скомпільованих мовах, таких як C або C++.[29].



Рисунок 3.7 — Мова програмування Python

Python — це багатопарадигмальна мова програмування, яка підтримує імперативне, процедурне, об'єктно-орієнтоване, метапрограмування, функціональне та структурне програмування. Завдяки динамічній типізації Python дозволяє ефективно вирішувати загальні проблеми програмування. Часткова підтримка аспектно-орієнтованого програмування реалізована за допомогою декораторів і може бути повністю реалізована за допомогою

додаткових фреймворків. Контрактне та логічне програмування можна реалізувати за допомогою спеціалізованих бібліотек або розширень.

Основними архітектурними особливостями Python є динамічний тип, автоматичне керування пам'яттю, самоспостереження, механізми обробки винятків, підтримка багатопотокових обчислень за допомогою глобального блокування інтерпретатора (GIL) і вдосконалені структури даних. Python підтримує модульність програми, що дозволяє організовувати код у вигляді модулів і об'єднувати їх у пакети.

Стандартна бібліотека Python надає велику кількість функцій, що охоплюють різноманітні завдання від обробки текстів до створення веб-додатків. Додаткові функціональні можливості, такі як математичне моделювання, апаратне забезпечення, розробка веб-додатків та ігор, можуть бути досягнуті за допомогою численних сторонніх бібліотек. Python також підтримує інтеграцію з кодом, написаним на C або C++, а сам інтерпретатор можна інтегрувати в проекти на цих мовах. Щоб полегшити пошук і інсталяцію бібліотек, існує офіційний репозиторій PyPI, який є стандартом де-факто і містить десятки тисяч готових до використання пакетів.

Python є однією з найпопулярніших мов програмування. Він широко використовується в аналізі даних, машинному навчанні, DevOps, веб-розробці та розробці ігор. Завдяки простому синтаксису, зручній читабельності та відсутності необхідності компіляції Python ідеально підходить для навчання програмуванню, оскільки дозволяє зосередитися на алгоритмах і концепціях. Інтерпретований характер мови спрощує процес налагодження та експериментування.

Python активно використовується багатьма провідними компаніями, такими як Google і Facebook, і досі очолює рейтинги популярності мов програмування. Станом на жовтень 2024 року Python залишається найпопулярнішою мовою програмування з популярністю 21,9%.

Python — це портативна мова програмування, яка підтримує більшість сучасних платформ від портативних пристроїв (ПК) до мейнфреймів. Він має

програмне забезпечення для Microsoft Windows, майже всіх варіантів UNIX (включно з FreeBSD і Linux), Android, macOS, iOS (починаючи з версії 2.0), iPadOS, Plan 9, Palm OS, OS/2, Amiga, HaikuOS, AS/400, Port /390 для ОС, Windows Mobile і Symbian.

Мова підтримує динамічну типізацію, де тип змінної визначається під час виконання. Замість традиційного «присвоєння значення змінній», Python «прив'язує значення до імені». Основні типи Python включають логічні значення, цілі числа довільної точності, числа з плаваючою комою та комплексні числа. Типи контейнерів включають рядки, списки, кортежі, словники та набори. Усі об'єкти в Python є екземплярами класів, включаючи функції, методи та модулі.

Бібліотека Playsound є кросплатформною та повністю написана на Python для відтворення звукових сигналів у системі. Він має мінімальну функціональність і залежності, використовує функцію «playsound» для відтворення звукових файлів. Ця функція приймає один обов'язковий параметр для шляху до звукового файлу, який може бути локальним шляхом або вираженням у вигляді URL-адреси. Це гарантує, що бібліотеку можна легко інтегрувати в різні проекти [30].

Бібліотека playsound має додатковий другий блок параметрів, який за замовчуванням має значення "True". Якщо встановлено значення «False», функція виконується асинхронно.

У Windows playsound використовує "windll.winmm". Формати WAVE та MP3 перевірено та відомо, що вони працюють. Інші формати файлів також можуть працювати. macOS використовує AppKit.NSSound. Загалом, усе, що підтримує QuickTime, має відтворюватися на macOS. У Linux playsound використовує GStreamer, який, як відомо, працює з Ubuntu 14.04 і ElementaryOS Loki.

Dlib — це сучасний набір інструментів C++, що містить алгоритми машинного навчання та інструменти для створення складного програмного забезпечення C++ для вирішення реальних проблем. Dlib широко

використовується в промисловості та наукових колах у таких сферах, як робототехніка, вбудовані пристрої, мобільні телефони та великомасштабні високопродуктивні обчислювальні середовища. Ліцензія з відкритим кодом Dlib дозволяє використовувати його безкоштовно [31].

На відміну від багатьох проектів з відкритим кодом, ця бібліотека надає повну та точну документацію для кожного класу та функції. Крім того, він підтримує режим налагодження для тестування записаних функціональних умов. Увімкнення цього параметра дозволяє виявляти більшість помилок, викликаних неправильними викликами функцій або неправильним використанням об'єктів.

Бібліотека має покриття одиничних тестів із співвідношенням приблизно від 1 до 4 рядків тестового коду до рядків коду в самій бібліотеці. Він регулярно тестується в середовищах MS Windows, Linux і macOS. Однак він також повинен працювати на будь-якій системі, сумісній з POSIX, і раніше використовувався на Solaris, HPUX і BSD.

Для використання цієї бібліотеки не потрібні додаткові пакети чи складне налаштування. Для цього потрібні лише стандартні API, надані операційною системою. Весь код, який залежить від операційної системи, ізольовано на рівні абстракції операційної системи. Решта бібліотеки базується на цих рівнях або є чистим стандартом C++ ISO.

Бібліотека NumPy використовується для обчислення евклідової відстані, перетворення отриманих результатів у матриці та обчислення необхідних метрик (рис. 3.8).



Рисунок 3.8 — Можливості бібліотеки NumPy

NumPy забезпечує:

- потужний N-вимірний масив об'єктів;
- підтримка комплексних (трансляційних) функцій;
- засоби інтеграції коду C/C++ і Fortran;
- функції для обробки лінійної алгебри, перетворення Фур'є,

випадкових чисел тощо.

Окрім використання в наукових обчисленнях, NumPy може служити ефективним багатовимірним контейнером для даних загального призначення. Він дозволяє визначати типи даних, тим самим спрощуючи інтеграцію з різними базами даних. Усі пакети NumPy, що розповсюджуються через PyPI, ліцензовані за ліцензією BSD [32].

PyCharm — інтегроване середовище розробки (IDE) для Python [33] було обрано для написання, компіляції та тестування системи програмного забезпечення. PyCharm автоматизує рутинні завдання, дозволяючи розробникам зосередитися на важливих аспектах свого коду. Завдяки механізму інтелектуального аналізу коду PyCharm забезпечує:

- точне автозаповнення;
- швидко знаходити помилки та виправляти їх;
- зручна кодова навігація;
- інструменти для забезпечення відповідності стандартам PEP8;
- рефакторинг коду та функція тестування.

- калібрування камери та 3D-реконструкція (calib3d) – цей модуль надає базові алгоритми для різних геометрій, калібрування одиночної та стереокамери, оцінку положення об’єкта, стереозбіг та 3D-реконструкцію;
- 2D Features Framework (features2d) — модуль для роботи з детекторами ключових точок, дескрипторами та інструментами порівняння;
- розпізнавання об’єктів (objdetect) – цей модуль забезпечує виявлення об’єктів і екземплярів визначених класів, таких як обличчя, очі, люди, автомобілі тощо;
- графічний інтерфейс високого рівня (highgui) — простий модуль для створення інтерфейсів користувача з базовим функціоналом;
- Video In/Out (videoio) — інтерфейс для захоплення відео та використання відеокодеків;
- додаткові модулі — включає інструменти підтримки для тестових оболонок FLANN і Google, прив’язки Python тощо.

Ці модулі забезпечують широкий спектр функціональних можливостей, що робить OpenCV універсальним інструментом для вирішення проблем комп’ютерного зору.

3.5 Побудова алгоритму програми та практична реалізація системи

Спочатку створіть екземпляр класу VideoStream, який використовує індекс із веб-камери вашого ноутбука для запису відео. Після цього настає односекундна пауза, щоб датчик камери увімкнувся та стабілізувався. У циклі відеопотік обробляється кадр за кадром. Кожен кадр попередньо обробляється: його розмір змінюється до 450 пікселів, а сам кадр перетворюється на відтінок сірого. Для визначення положення обличчя на кадрі використовується детектор обличчя бібліотеки dlib.

Потім використовується розпізнавання рис обличчя, щоб знайти кожну важливу область. Кожне розпізнане обличчя обробляється циклічно (хоча передбачається, що в кадрі буде лише одне обличчя – драйвер, алгоритм

підтримує обробку кількох облич). Для кожного виявленого обличчя застосуйте детектор обличчя dlib і конвертуйте результат у масив формату NumPy.

Використовуючи фрагмент масиву, ви можете отримати координати (x, y) лівого та правого ока. Знаючи ці координати, можна розрахувати співвідношення сторін кожного ока. Ці значення усереднені для підвищення точності. Далі функція cv2.drawContours може бути використана для візуалізації області очей на кадрі, що корисно для перевірки правильності визначення та позиціонування очей. Алгоритм цієї програми показаний у вигляді блок-схеми на рисунку 3.10.

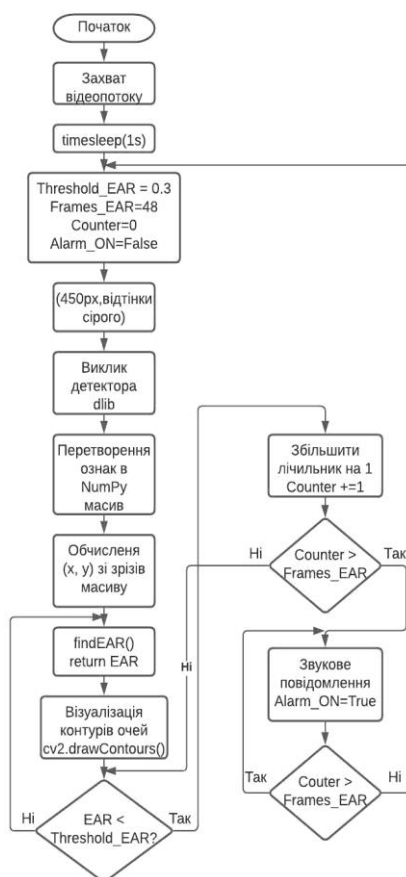


Рисунок 3.10 — Блок-схему алгоритму програми

Щоб розпочати впровадження системи, необхідно створити новий файл із розширенням .py, відкрити його в текстовому редакторі чи інтегрованому середовищі розробки Python. У цьому прикладі використовується професійне середовище розробки PyCharm. Після цього підключіть необхідні бібліотеки.

Повний список програм наведено в Додатку Б, а код для імпорту бібліотеки у Лістингу 1.

Лістинг 1. Імпорт бібліотек для реалізації системи:

```
import dlib
import numpy
from cv2 import videostream
from playsound import playsound
import time
```

Для відтворення звукового сигналу в системі необхідно створити звуковий потік, вказавши шлях до звукового файлу. Для цього використовується бібліотека Playsound. Функція програмного забезпечення для відтворення звуку показана в лістингу 2.

Лістинг 2. Функція відтворення звукового повідомлення водію:

```
def runAlarm(path):
    # відтворити повідомлення
    playsound.playsound(C:\Users\asus\Music\Alarm)
```

Щоб обчислити відношення евклідової відстані між двома наборами горизонтальних і вертикальних елементів ока (з координатами x і y), була створена функція findEAR(). Ця функція повертає метрику стану та спостереження для кожного ока за допомогою матриці евклідової відстані. Код функції програми для функціонування та обчислення евклідової відстані наведено в лістингу 3.

Лістинг 3. Функція обчислення евклідової відстані:

```
def findEAR(eye):
    # обчислити евклідові відстані між двома наборами вертикальних ознак
    # ока (x, y) координат
    A = dist.euclidean(eye[1], eye[5])
```

```

B = dist.euclidean(eye[2], eye[4])
# те саме для горизонтальних ознак
C = dist.euclidean(eye[0], eye[3])
# обчислити EAR
ear = (A + B) / (2.0 * c)
# повернути його
return ear

```

Коли очі відкриті, значення співвідношення сторін очей, яке повертає ця функція, залишається приблизно постійним і додатним. У момент миготіння це значення швидко падає до нуля [19].

Якщо очі закриті, співвідношення сторін очей знову залишається стабільним, але набагато меншим, ніж у відкритих. Візуалізація позначки ока показана на малюнку 3.11.



Рисунок 3.11 — Візуалізація орієнтирів очей водія

Верхній лівий кут показує стан відкритих очей, а верхній правий кут показує стан закритих очей. На рисунку 3.12 зображено графік залежності коефіцієнта сторони ока від часу. Зменшення значення співвідношення сторін вказує на момент миготіння.

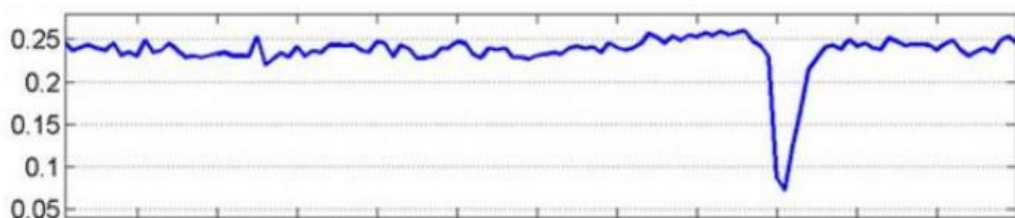


Рисунок 3.12 — Графік співвідношення сторін очей водія із часом

Під час проектування детектора необхідно контролювати співвідношення сторін очей, щоб визначити, чи значення зменшується, а більше не збільшується, вказуючи на те, що очі водія закриті. Після цього встановіть деякі важливі змінні, які будуть використовуватися в подальших розрахунках. Визначення цих змінних наведено в лістингу 4.

Лістинг 4. Створення ключових змінних для детектування очей водія (порогові значення):

```
# визначити дві константи:
```

```
# одну для співвідношення сторін ока, щоб визначити моргання, а потім
другу константу для кількості послідовних кадрів.
```

```
#стан ока має бути нижче порогового значення, щоб спрацював сигнал
тривоги
```

```
THRESHOLD_EAR = 0.3
```

```
FRAMES_EAR = 48
```

```
# ініціалізувати лічильник кадрів, а також логічне значення,
використовуване для індикації.Спрацьовування сигналізації
```

```
COUNTER = 0
```

```
ALARM_ON = False
```

Визначено Визначається змінна THRESHOLD_EAR, яка встановлює порогове значення для співвідношення сторін очей. Якщо значення нижче порогового значення, почніть підраховувати кількість кадрів, коли очі залишаються закритими.

Якщо кількість таких кадрів перевищує значення FRAMES_EAR, система видасть звуковий сигнал.

Експерименти виявили, що значення THRESHOLD_EAR 0,3 добре працює за різних умов. Для параметра FRAMES_EAR оптимальне значення 48: сигнал відтворюється, якщо очі залишаються закритими протягом 48 кадрів поспіль. Чутливість детектора можна регулювати, змінюючи значення

FRAMES_EAR — зменшення значення робить детектор більш чутливим, збільшення значення робить детектор менш чутливим.

Лістинг 3 також визначає змінну COUNTER, яка відстежує кількість послідовних кадрів, у яких співвідношення сторін ока менше ніж THRESHOLD_EAR. Якщо COUNTER перевищує FRAMES_EAR, змінна ALARM_ON встановлюється на логічне значення, що активує сигнал тривоги.

Бібліотека dlib включає детектор обличчя на основі гістограми орієнтованих градієнтів (HOG) і предиктор орієнтирів обличчя. Відповідний екземпляр створюється в наступному блоці коду, як показано в лістингу 5.

Лістинг 5. Створення екземплярів детектора обличчя і лицьових ознак водія:

```
detector=dlib.get_frontal_face_detector()
predictor=dlib.shape_predictor(args["shape_predictor"])
```

Риси обличчя (орієнтири), створені за допомогою бібліотеки dlib, є індексованим списком, як показано на малюнку 3.13. Щоб вибрати область ока з набору орієнтирів, необхідно знати відповідний індекс зрізу масиву. Лістинг 6 показує приклад.

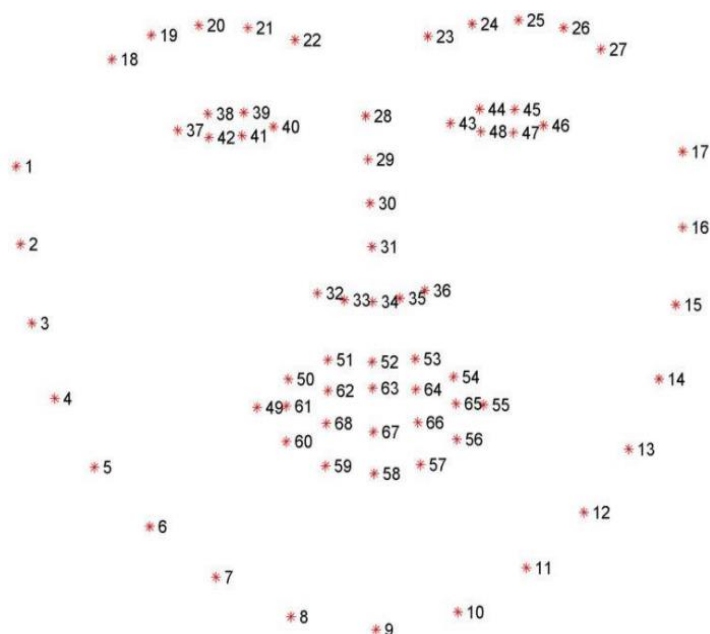


Рисунок 3.13 — Індексований список лицьових орієнтирів обличчя

Лістинг 6. Спискові зрізи індексу очей:

```
(leftEyest, leftEyeEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rightEyest, rightEyeEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
```

За допомогою цих індексів область очей можна легко виділити за допомогою нарізки списку. Базовий код детектора сонливості водія наведено в лістингу 7.

Лістинг 7. Ядро детектора сонливості водія: # запустити захоплення відеопотоку з веб-камери

```
vs = VideoStream(src=args["webcam"]).start()
time.sleep(1.0)
```

цикл з кадрів з відеопотоку

```
while True:
```

захопити кадр із потокового відеофайлу, змінити його розмір і перетворити на канали у відтінки сірого

```
    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    #Виявити обличчя за допомогою детектора
    rects=detector(gray, 0)
```

Нарешті все готово, щоб перевірити, чи не починає водій у відеопотоці проявляти симптоми сонливості.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Експериментальне дослідження системи

Перші програмні системи створювалися в рамках наукових досліджень або на замовлення. Тестування таких систем здійснюється в суворо формалізованому порядку з фіксацією всіх процедур тестування, вхідних даних і результатів. Цей процес вважається окремою фазою, яка починається після завершення фази кодування, але зазвичай виконується тим самим персоналом.

Зверніть особливу увагу на «вичерпне» тестування, яке передбачає перевірку всіх можливих шляхів у коді або всіх варіантів вхідних даних. Однак пізніше було зазначено, що повне тестування програмного забезпечення було практично неможливим через ряд причин: велика кількість можливих вхідних даних, велика кількість шляхів виконання коду та складність виявлення проблем в архітектурі та специфікаціях. Тому «вичерпне» тестування було визнано теоретично неможливим і відкинуто.

4.2 Опис основних етапів тестування

Тестування програмного забезпечення — це процес перевірки відповідності між фактичною та очікуваною поведінкою програми на основі набору особливим чином підібраних тестів. У більш широкому сенсі тестування — це один із методів контролю якості, який включає наступні етапи: планування роботи з управління тестуванням, дизайн тесту, проектування тесту, виконання тесту, виконання тесту та аналіз результатів аналізу тесту

Валідація — це процес оцінювання системи або її компонентів для визначення того, чи відповідають результати поточної фази розробки умовам, визначеним на початку цієї фази. Це перевірка на відповідність цілям, термінам і завданням, визначеним на поточному етапі розробки.

Перевірка — це процес оцінювання того, чи відповідає розроблене програмне забезпечення очікуванням і потребам користувача та системним вимогам.

План тестування — це документ, який описує всі аспекти процесу тестування. Він включає:

- опис об'єкта випробувань;
- стратегія тестування;
- графік роботи;
- критерії початку і закінчення контрольної роботи;
- необхідне обладнання;
- спеціальні знання;
- оцінити ризики та варіанти вирішення.

Тестовий дизайн — це етап процесу тестування програмного забезпечення, на якому тестові випадки проектуються та створюються відповідно до визначених стандартів якості та цілей тестування.

Тестовий приклад — це задокументований артефакт, який містить набір кроків, конкретних умов і параметрів, необхідних для тестування реалізації функції або її частини.

Звіт про помилку/дефект (звіт про помилку) — це документ, який описує ситуацію або послідовність операцій, що призвели до неправильної роботи об'єкта тестування, із зазначенням причини та очікуваних результатів.

Тестове покриття — це показник, який використовується для оцінки якості тесту та визначення щільності тестового покриття вимог або виконуваного коду.

Детальні тестові випадки (специфікації тестового прикладу) — це рівень деталізації в описі етапів тестування та очікуваних результатів, який забезпечує найкраще співвідношення між часом виконання тесту та якістю покриття тесту.

Час проходження тестового прикладу — це час, що минув від початку кроку тестового прикладу до отримання результатів тестування.

4.3 Покрокове тестування системи

Пропонована покращена система розпізнавання очей водія має камеру, яка буде розташована на верхньому краю лобового скла автомобіля або на торпеді перед водієм, як показано на малюнку 4.1.



Рисунок 4.1— Місце для розташування камери в машині

Це дозволить камері фокусуватися виключно на водієві, не відриваючи його від кадру. Завдяки функціональності інфрачервоної камери система працює навіть при поганому освітленні обличчя водія або вночі. Сонцезахисні окуляри, які носить водій, також не заважають роботі системи. Система реалізована на мові програмування Python.

Коли машина запускається, система запускається автоматично. Спочатку налаштуйте камеру для відстеження всіх людей у відеокадрах потоку, як показано на малюнку 4.2. Слід зазначити, що камера орієнтована виключно на водія, тому система не розпізнає додаткові обличчя.

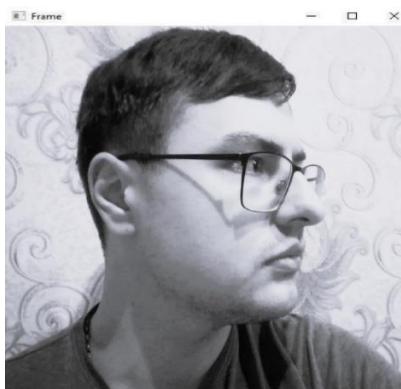


Рисунок 4.2 — Пошук облич у вхідному відеопотоці.

Коли обличчя знайдено, то виконується визначення ознак очей на обличчі та дістаються області очей, як зображено на рисунку 4.3, де EAR – поріг відношення сторін очей.

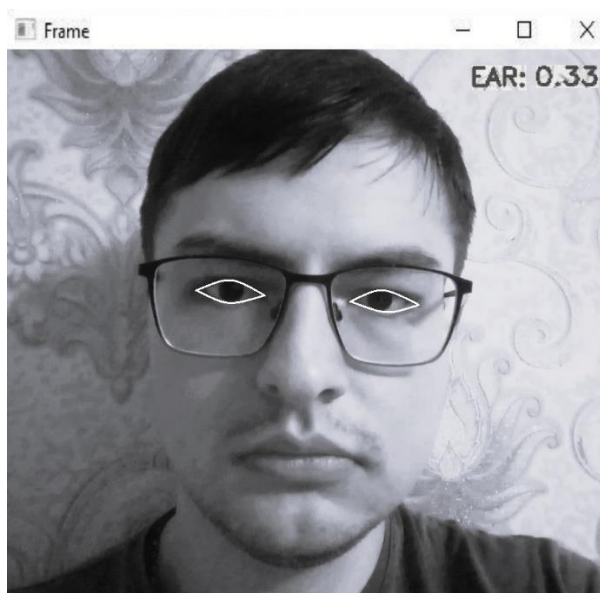


Рисунок 4.3 — Визначення ознак очей на обличчі

Написана програмна функція `cv2.drawContours()` для виділення контурів очей працює коректно. Тепер, коли на кадрі виділено області очей, можна обчислити співвідношення сторін ока (EAR – Eye Aspect Ratio), щоб визначити, чи є очі закритими, як показано на рисунку 4.4.

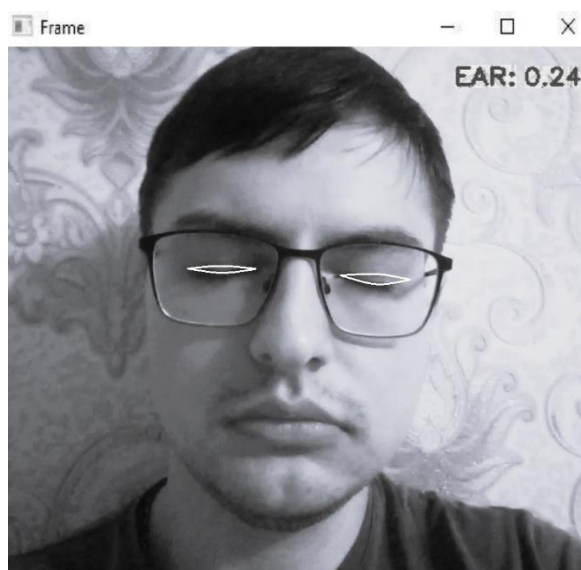


Рис. 4.4 — Визначення співвідношення сторін ока

Якщо співвідношення сторін ока вказує про те, що очі залишаються закритими протягом тривалого часу, активується звукове повідомлення, призначене для пробудження водія, який заснув, це зображено на рисунку 4.5.

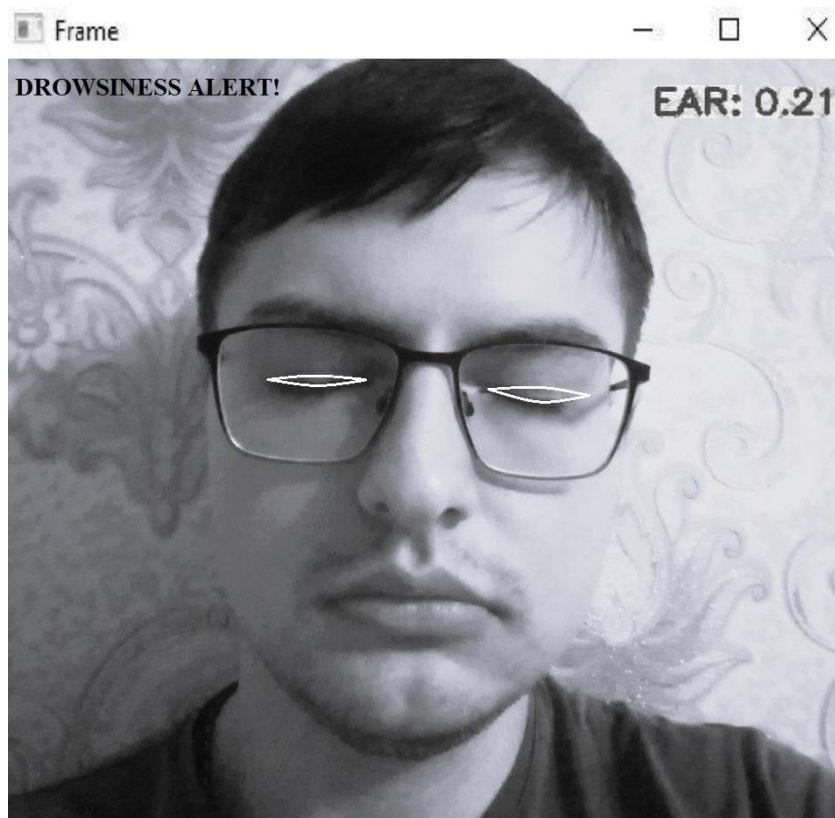


Рис. 4.5 — Виклик звукового повідомлення для пробудження водія

Після того як водій знову відкриває очі, ознаки обличчя перевіряються повторно, а програмні константи та змінні повертаються до значень за замовчуванням.

Якщо кількість кадрів, на яких очі залишаються закритими, перевищує `FRAMES_EAR`, активується звуковий сигнал.

Експериментально встановлено, що значення `THRESHOLD_EAR`, яке дорівнює 0,3, ефективно працює у різних умовах. Значення `FRAMES_EAR` було визначено як 48: якщо очі залишаються закритими протягом 48 послідовних кадрів, система видає звукове сповіщення. Для підвищення чутливості детектора значення `FRAMES_EAR` можна зменшити, а для зниження — збільшити.

Розрахунок евклідових відстаней виконується миттєво та не залежить від хмарних баз даних, що забезпечує перевагу запропонованої системи над існуючими аналогами. Швидкодія системи повністю відповідає поставленим вимогам.

Однак існує ризик того, що перевтомлений водій може ігнорувати звукові попередження системи. Тому пропонується додатково інформувати водія про закриті очі за допомогою вібрації водійського крісла та зменшення обертів двигуна.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нової системи контролю стану водія з безперервним скануванням обличчя водія з використанням методів розпізнавання обличчя і машинного зору. Особливістю розробки системи розпізнавання обличчя з машинним зором для моніторингу стану водія в реальному часі є те, що вона враховує індивідуальні особливості та забезпечує високу точність у виявленні ознак втоми і сонливості, а також може бути використана в автомобілях для особистого користування, пасажирському та вантажному транспорті, а також у логістичних перевезеннях.

Аналогом може бути Антисон для водіїв нового покоління з GPS контролем швидкості Dunobil — приблизна вартість 8400 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність в реальних умовах

Продовження табл. 5.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження табл. 5.1

9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	4	2

Продовження табл. 5.2

Наявність аналогів на ринку	2	2	2
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	3	3	3
Експлуатаційні витрати	4	2	3
Ринок збуту	2	3	2
Конкурентоспроможність	3	4	4
Фахівці з технічної і комерційної реалізації	2	3	2
Фінансування	2	2	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	4	3	3
Сума	36	37	35
Середньоарифметична сума балів	$(36+37+35) / 3 = 36$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового виробу є вище середнього, завдяки мікропроцесорному засобу оперативного контролю стану водія, яка дозволить ефективно аналізувати і миттєво повідомляти про нестабільний стан водія, що у свою чергу, дозволить зменшити відсоток дорожньо-транспортних пригод (ДТП) та збільшити безпеку на дорогах.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$З_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де М — місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p — число робочих днів в місяці, 23 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Таблиця 5.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	21000	913,04	28	25565,217
Інженер	20000	869,57	28	24347,826
Всього				49913,04

5.2.2 Витрати на основну заробітну плату робітників ($З_p$) розраховуються на основі норм часу, які необхідні для виконання даної роботи, розраховуються за формулою:

$$З_p = \sum_{i=1}^n t_i \cdot C_i \cdot K_c, \quad (5.2)$$

де t_i — норма часу (трудомісткість) на виконання конкретної роботи, годин;

n — число робіт по видах та розрядах;

K_c — коефіцієнт співвідношень, який установлений в даний час Генеральною тарифною угодою між Урядом України і профспілками;

C_i — погодинна тарифна ставка робітника відповідного розряду, який виконує відповідну роботу, грн./год.

C_i визначається за формулою:

$$C_i = \frac{M_m \cdot K_i}{T_p \cdot T_{зм}}, \quad (5.3)$$

де M_m — мінімальна місячна оплата праці, грн., 8000 грн. у 2024 р.

K_i — тарифний коефіцієнт робітника відповідного розряду;

T_p — число робочих днів в місяці, $T_p = 23$ дні;

$T_{зм}$ — тривалість зміни, $T_{зм} = 8$ годин.

Погодинна тарифна ставка згідно чинного законодавства у грудні 2024 року = 48 грн./год.

Розрахунки заносимо до табл. 5.5.

Таблиця 5.5 — Витрати на основну заробітну плату робітників

Найменування робіт	Трудовісткість, год.	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн.	Величина оплати на робітника грн.
Заготівельні	1	3	1,35	64,8	64,8
Слюсарно-збиральні	3	3	1,35	64,8	194,4
Налагоджувальні та контрольні	1	5	1,7	81,6	81,6
Всього					340,80

5.2.3 Додаткова заробітна плата прийнято розраховувати як 10% від основної заробітної плати розробників та робітників:

$$З_д = (З_{о.роз} + З_{о.роб}) \cdot 10\% / 100\% \quad (5.2)$$

$$З_д = (49913,04 + 340,80) \cdot 10\% / 100\% = 5025,38 \text{ (грн.)}$$

5.2.4 Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_{о.роз} + З_{о.роб} + З_д) \cdot 22\% / 100\% \quad (5.3)$$

$$Н_з = (49913,04 + 340,80 + 5025,38) \cdot 22\% / 100\% = 12086,45 \text{ (грн.)}$$

5.2.5 Оскільки для розроблювального програмного продукту не потрібно витрачати матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.6 Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{Т_в} \cdot \frac{t_{вик}}{12} \text{ [грн.]} \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;

Т — термін корисного використання обладнання згідно податкового законодавства, років;

$t_{вик}$ — термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 20000 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його фактичного використання — 1,217 міс.

$$A_{\text{обл}} = \frac{20000}{2} \times \frac{1,217}{12} = 1014,49 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.6.

Таблиця 5.6 — Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	20000	2	1,217	1014,493
Офісне обладнання (меблі)	22000	4	1,217	557,971
Приміщення	750000	20	1,217	3889,266
Всього				5461,73

Амортизація обладнання, що використовувалось робітниками, розраховується аналогічно, результати розрахунків зведено в таблицю 5.7 і враховуються при розрахунку виробничої собівартості виробу.

Таблиця 5.7 — Амортизаційні відрахування матеріальних і нематеріальних ресурсів для робітників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання		Амортизаційні відрахування, грн.
			год.	міс.	
Комп'ютер	20000	2	1	0,0054	4,5290
Спеціалізоване обладнання (меблі)	22000	4	1	0,0054	2,4909
Приміщення	750000	20	5	0,0272	84,9185
Всього					91,9384

Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів, а також спеціалізованого обладнання менше 20000

грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{\text{спец. обл.}} = 7500$ грн.

5.2.6. Витрати на комплектуючі

Витрати на комплектуючі, що були використані на виготовлення розраховуються за формулою

$$K = \sum_{i=1}^n H_i \cdot C_i \cdot K_i, \quad (5.5)$$

де H_i — кількість комплектуючих i -го виду, шт.,

C_i — роздрібна ціна комплектуючих i -го виду, грн.,

K_i — коефіцієнт транспортних витрат, $K_i=1,1$,

n — кількість видів матеріалів.

Проведені розрахунки зводимо до таблиці 5.8 без врахування транспортних витрат.

Таблиця 5.8 — Витрати на комплектуючі

Найменування комплектуючих	Кількість	Ціна за штуку, грн.
Raspberry Pi 2	1	2190
Logitech WebCam C170	1	850
Raspberry Pi 3.2inch LCD 320x240	1	850
Динамік мініатюрний 8 ом	1	15
Всього		3905,00

Витрати на комплектуючі, що були використані на розробку з врахуванням транспортних витрат:

$$H = 3905 \cdot 1,1 = 4295,50 \text{ (грн.)}$$

5.2.8 Тарифи на електроенергію для побутових споживачів

(промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$B_e = B \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (5.6)$$

де B — вартість 1 кВт-години електроенергії для 1 класу підприємства з ПДВ в 2024 році для Вінницької області за даними Енера-Вінниця, $B = 10,42$ грн./кВт;

Π — встановлена середня потужність обладнання, кВт. $\Pi = 0,4$ кВт;

Φ — фактична кількість годин роботи обладнання, годин.

K_{Π} — коефіцієнт використання потужності, $K_{\Pi} = 0,8$.

$$\begin{aligned} B_e &= 0,8 \cdot 0,4 \cdot 8 \cdot 28 \cdot 10,42 + 0,8 \cdot 0,4 \cdot 2,0 \cdot 10,42 = 746,9056 + 6,669 = \\ &= 753,57 \text{ (грн.)} \end{aligned}$$

5.2.9 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_{\varepsilon} = (3_o + 3_p) \cdot \frac{H_{ib}}{100\%}, \quad (5.6)$$

де H_{ib} — норма нарахування за статтею «Інші витрати».

$$I_e = (49913,04 + 187,44) \cdot 55\% / 100\% = 27639,61 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (3_o + 3_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.7)$$

де $H_{нзв}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = (49913,04 + 340,80) \cdot 110\% / 100\% = 55279 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 49913,04 + 340,80 + 5025,38 + 12086,45 + 5461,73 + 91,9384 + 7500 + \\ + 4295,50 + 753,57 + 27639,61 + 55279 = 168387,27 \text{ грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ZB , визначається за формулою:

$$ZB = \frac{B_{\text{заг}}}{\eta} \quad (\text{грн}), \quad (5.8)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ZB = 168387,27 / 0,5 = 336775 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);
- кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.9)$$

де $\pm\Delta\Pi_0$ — зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 — вартість програмного продукту у році до впровадження результатів розробки;

ΔN — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ — коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ — коефіцієнт, який враховує рентабельність продукту;

ϑ — ставка податку на прибуток, у 2024 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 6500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 100 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 1000 шт., протягом другого року – на 1500 шт., протягом третього року на 2000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0 \cdot 100 + (6500 + 100) \cdot 1000) \cdot 0,8333 \cdot 0,1 \cdot (1 - 0,18) = 444166,649 \text{ грн.}$$

$$\Delta\Pi_2 = (0 \cdot 100 + (6500 + 100) \cdot (1000 + 1500)) \cdot 0,8333 \cdot 0,1 \cdot (1 - 0,18) = 1127499,955 \text{ грн.}$$

$$\Delta\Pi_3 = (0 \cdot 100 + (6500 + 100) \cdot (1000 + 1500 + 2000)) \cdot 0,8333 \cdot 0,1 \cdot (1 - 0,18) = 2029499,919 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 3601166,52 грн.

Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.10)$$

де $\Delta\Pi_i$ — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T — період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t — період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\Pi\Pi = (444166,649 / (1 + 0,1)^1) + (1127499,955 / (1 + 0,1)^2) + (2029499,919 / (1 + 0,1)^3) = 403787,86 + 931818,145 + 1524793,33 = 2860399,335 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{inv} * 3B, \quad (5.11)$$

де k_{inv} — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{inv}=2...5$, але може бути і більшим;

$3B$ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 336775 = \text{грн.}$$

Тоді абсолютний економічний ефект E_{abc} або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV, \quad (5.12)$$

$$E_{abc} = 2860399,335 - 336775 = 2523624,80 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту

мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього використаємо формулу:

$$E_e = \sqrt[T_{ж}]{1 + \frac{E_{acc}}{PV}} - 1, \quad (5.13)$$

де $T_{ж}$ — життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{(1 + 2523624,80 / 336775) - 1} = 1,040$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,09...0,14)$;

f — показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_e > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_e}, \quad (5.15)$$

$$T_{ок} = 1 / 1,040 = 0,96 \text{ р.}$$

Оскільки $T_{ок}$ менше 3-х років, а саме термін окупності рівний 0,96 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку продукту, сума яких складає 336775 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений продукт за ціною дешевший за аналог і є високо конкурентоспроможним. Період окупності складе близько 0,96 роки.

ВИСНОВКИ

Результатом цієї роботи стала розробка прототипу пристрою на основі алгоритмів комп'ютерного зору, який не дає водіям заснути за кермом. Прототип реалізує базові функції, призначені для виявлення таких об'єктів, як «обличчя» та «очі», на основі портативної апаратної та обчислювальної платформи Raspberry Pi 2.

У ході роботи було проведено порівняльний аналіз популярних мікрокомп'ютерів і мікроконтролерів, які можуть бути придатними платформами для створення прототипів. Вибір платформи Raspberry Pi 2 ґрунтувався на порівнянні платформ за обсягом оперативної пам'яті, тактовою частотою ЦП, розміром і вартістю.

Крім того, розробка програмного забезпечення на основі бібліотек комп'ютерного зору виконується з використанням мови програмування Python. Ключовим методом виявлення обличчя та очей є алгоритм Віоли-Джонса, який реалізований на основі класифікаційного каскаду Хаара.

В якості основного методу оцінки стану людини був обраний метод оцінки стану очей, тобто як довго вони відкриті або закриті і як довго вони відкриті або закриті. Судити про стан очей в основному ґрунтуються на виявленні зіниць. Програмне забезпечення, реалізоване за допомогою бібліотеки Dlib, може визначати риси обличчя і стан очей водія.

В ході роботи отримав досвід використання мови програмування Python, комп'ютерного зору та бібліотек OpenCV в операційних системах Windows.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кількість ДТП від патрульної поліції України . URL: <https://patrolpolice.gov.ua/statystyka/>.
2. Національний центр досліджень охорони здоров'я: водіння в стані сонливості URL: <https://www.center4research.org/drowsy-driving-stop-falling-asleep-wheel/>.
3. Європейська Комісія питань мобільності й транспорту. URL: https://insat.org.ua/files/nav/law/3/vehicle_information_platform_uk.pdf.
4. Європейська Комісія питань мобільності й транспорту. URL: <https://www.abc.net.au/news/2021-05-12/how-to-tell-if-youre-suffering-fatigue-behind-the-wheel/100133940>.
5. Пристрій Stopslee. URL: <https://www.importitall.co.za/Anti-Sleep-Alarm-for-Drivers-Warns-up-to-5-Minutes-Before-Drowsiness-Beep-and-Vibration-Doze-Alert-Car-Truck-S-ap-B00A33APOU.html>.
6. Виявлення сонливості водія BOSCH. URL: <https://www.auto123.com/en/news/bosch-unveils-drive-drowsiness-detection/12153/>.
7. Виявлення сонливості водія BOSCH. URL: <https://www.auto123.com/en/news/bosch-unveils-drive-drowsiness-detection/12153/>.
8. Пристрій Anti sleep pilot. URL: <https://newatlas.com/anti-sleep-pilot-monitors-driver-fatigue/17439/>.
9. Система попередження про сходження зі смуги URL: <https://www.nissan-global.com/EN/INNOVATION/TECHNOLOGY/ARCHIVE/LDW/>.
10. Системи розпізнавання облич Facial recognition technology (FRT) URL: <https://maxnet.ua/blog/yak-kameri-videosposterezheniya-rozpiznayut-oblichchya/#:~:text=Принцип%20роботи%20системи%20розпізнавання%20облич,локалізації%20певної%20людини%20на%20знімку>.

11. Аналіз існуючих підходів до розпізнавання облич URL:
https://www.researchgate.net/figure/Face-Recognition-processing-flow_fig1_331646485
12. Навчальний посібник із курсу «Комп'ютерна обробка зображень» URL: http://aco.ifmo.ru/el_books/image_processing/1_01.html – Дата обращения: 20.04.2017 г.
13. Learning OpenCV / D. Bradsky, A. Kaehler. // O'Reily Media, Inc. – 2008. URL:
https://books.google.com.ua/books?id=seAgiOfu2EIC&printsec=frontcover&hl=uk&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
14. OpenCV URL: <http://opencv.org> . Дата обращения: 20.01.2017 г.
15. Digital image processing / R. Gonzalez, R. Woods URL:
<https://dl.ebooksworld.ir/motoman/Digital.Image.Processing.3rd.Edition.www.EBooksWorld.ir.pdf>
16. A general framework for object detection / O.Papageorgiou // International Conference on Computer Vision, 1998 URL:
https://static.aminer.org/pdf/PDF/000/293/432/a_general_framework_for_object_detection.pdf
17. Rapid object detection using a boosted cascade of simple features / P. Viola, M.J. Jones. // Computer Vision and Pattern Recognition, 2001 URL:
<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>
18. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. /R. Lienhart, E. Kuranov, V. Pisarevsky URL:
<http://www.staroceans.org/documents/MRL-TR-May02-revised-Dec02.pdf>
19. Виявлення сонливості водія URL:
https://www.researchgate.net/figure/Eye-aspect-ratio-VI-IMPLEMENTATION_fig2_370612914
20. DLib: бібліотека для машинного навчання URL: [DLib: Library for Machine Learning - KDnuggets](#)

21. Face detection with dlib (HOG and CNN) URL: <https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/>
22. Евклідова відстань URL: <https://mathworld.wolfram.com/Distance.html>
23. Euclidean group URL: https://link.springer.com/chapter/10.1007/978-1-4757-4034-9_24
24. Мікрокомп'ютер BeagleBone Black URL: <https://evo.net.ua/beaglebone-black/>
25. Intel Galileo URL: <https://doc.arduino.ua/ru/hardware/IntelGalileo>
26. Мікрокомп'ютер Raspberry Pi 2 URL: <https://evo.net.ua/raspberry-pi/2052/>
27. Камера Logitech WebCam C170 URL: <https://defis.ua/vebkamera-logitech-webcam-c170-960-001066>
28. Сенсорний дисплей TFT LCD для Raspberry Pi URL: <https://arduino.ua/prod2123-3-2-320x240-tft-lcd-displei-dlya-raspberry-pi>
29. Python [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/>
30. Playsound 1.3.0 URL: <https://pypi.org/project/playsound/>
31. Dlib C++ library URL: <http://dlib.net/>
32. Numpy 2.1.3 URL: <https://pypi.org/project/numpy/>
33. PyCharm як найкраща IDE для розробки ПЗ на Python URL: <https://foxminded.ua/pycharm-tse/>
34. OpenCV Комп'ютерний зір із відкритим вихідним кодом URL: <https://docs.opencv.org/4.x/d1/dfb/intro.html>

ДОДАТОК А

Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

д.т.н., проф.

_____ О. Д. Азаров

“29” ____ 09 ____ 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

«Метод та мікропроцесорні засоби оперативного контролю стану водія
автомобіля»

08-54.МКР.005.00.000 ТЗ

Науковий керівник

к.т.н., доц. каф. ОТ

_____ Богомолів С.В.

виконав:

магістрант 2 курсу,

_____ Давидун М.В.

Вінниця 2024

1 Підстава для присвоєння кваліфікаційної атестації магістра

1.1 Тема цієї теми пов'язана з необхідністю покращення безпеки дорожнього руху, особливо в контексті моніторингу стану водія. Основні напрямки досліджень зосереджені на вдосконаленні алгоритмів виявлення втоми та неуважності.

1.2 Аналіз існуючих систем показує, що більшість зосереджується на загальному моніторингу водіїв без урахування їх особистих характеристик. Новий підхід спрямований на вдосконалення таких систем шляхом впровадження методів комп'ютерного зору та алгоритмів машинного навчання.

2 Мета і призначенням МКР

2.1 Метою даної роботи є створення мікропроцесорного засобу моніторингу стану та роботи водія з використанням технологій комп'ютерного зору та алгоритмів машинного навчання для аналізу фізіологічних показників для виявлення ознак втоми, зниження концентрації та інших факторів, що впливають на безпеку водіння. По-друге, забезпечити своєчасне попередження з метою запобігання ДТП.

2.2 Метою розробки є підвищення безпеки дорожнього руху та зменшення аварійності.

3 Вихідні дані для впровадження МКР

Вихідні дані для виконання МКР: мікрокомп'ютерна платформа Raspberry Pi 2, технічна документація на електронні компоненти, середовище розробки Python, бібліотека обробки відеопотоку OpenCV. технічна документація на електронні компоненти.

4 Вимоги до реалізації МКР

Система повинна працювати в режимі реального часу, мати можливість

з високою точністю виявляти ознаки втоми та змінювати параметри для підвищення адаптивності до різних умов середовища.

5 Етапи МКР та очікувані результати

Етапи роботи перераховані в таблиці А.1.

Таблиця А.1 — Етапи МКР

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Прим.
1	Постановка мети та задач роботи	20.10.24	
2	Огляд та аналіз методів та мікропроцесорних засобів оперативного контролю стану водія	30.10.24	
3	Вибір моделі мікроконтролера	04.11.24	
4	Проектування апаратної частини	08.11.24	
5	Розробка електричної принципової схеми	19.11.24	
6	Проектування програмної частини	24.11.24	
7	Розрахунок економічної частини роботи	01.12.24	
8	Оформлення пояснювальної записки та ілюстративного матеріалу	10.12.24	
9	Аналіз виконання роботи, висновки		
10	Перевірка якості виконання магістерської роботи та усунення недоліків		

6 Матеріали, подані на захист МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами.

Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення МКР

8.1 При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2023.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ–03.02.02 П.001.01:21.

ДОДАТОК Б

Лістинг програми

```
Aimport cv2
import dlib
import numpy as np
from scipy.spatial import distance
from playsound import playsound
import threading

# Функція для розрахунку співвідношення сторін очей (EAR)
def eye_aspect_ratio(eye):
    A = distance.euclidean(eye[1], eye[5])
    B = distance.euclidean(eye[2], eye[4])
    C = distance.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return ear

# Пороги та змінні для моніторингу стану очей
EYE_AR_THRESH = 0.25 # Поріг для визначення закритих очей
EYE_AR_CONSEC_FRAMES = 48 # Кількість кадрів для тригера
попередження

COUNTER = 0 # Лічильник для закритих очей
ALARM_ON = False # Статус сигналу

# Функція для звукового сигналу
def sound_alarm(path):
    playsound(path)
```

```
# Завантаження моделі для визначення обличчя та ознак обличчя
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

# Індекси для визначення очей
(lStart, lEnd) = (42, 48) # Ліве око
(rStart, rEnd) = (36, 42) # Праве око

# Захоплення відео
video_stream = cv2.VideoCapture(0)

while True:
    ret, frame = video_stream.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Визначення обличчя на кадрі
    rects = detector(gray, 0)

    for rect in rects:
        shape = predictor(gray, rect)
        shape = np.array([[p.x, p.y] for p in shape.parts()])

        # Визначення координат очей
        leftEye = shape[lStart:lEnd]
        rightEye = shape[rStart:rEnd]

        leftEAR = eye_aspect_ratio(leftEye)
```

```

rightEAR = eye_aspect_ratio(rightEye)

# Середнє співвідношення сторін очей
ear = (leftEAR + rightEAR) / 2.0

# Контури для візуалізації
leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

# Моніторинг стану очей
if ear < EYE_AR_THRESH:
    COUNTER += 1

    if COUNTER >= EYE_AR_CONSEC_FRAMES:
        if not ALARM_ON:
            ALARM_ON = True
            threading.Thread(target=sound_alarm, args=("alarm.wav",),
daemon=True).start()

            cv2.putText(frame, "ALERT! Drowsiness Detected", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        else:
            COUNTER = 0
            ALARM_ON = False

# Відображення кадру
cv2.imshow("Driver Monitoring System", frame)

```

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

```
# Завершення роботи  
video_stream.release()  
cv2.destroyAllWindows()
```

ДОДАТОК В

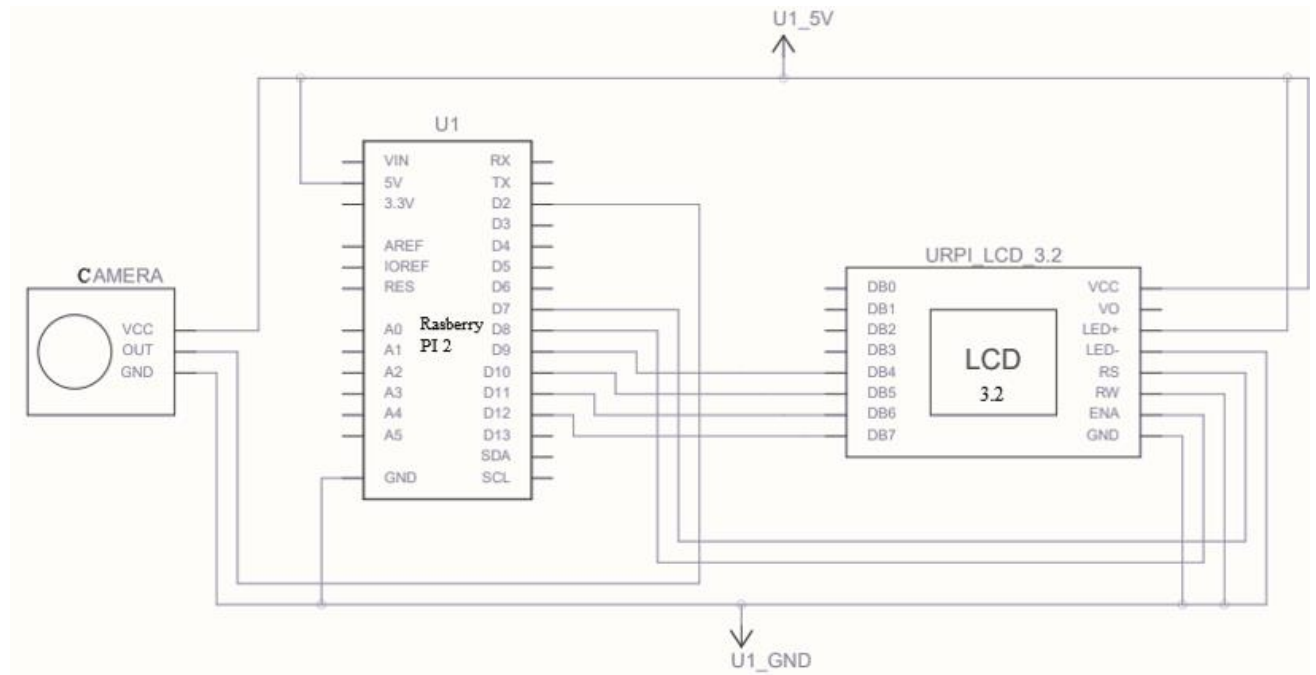
Схема електрична структурна



Рисунок В.1 — Алгоритм роботи удосконаленого методу спостереження за очима

ДОДАТОК Г

Схема електрична принципова



					08-54.МКР.005.00.000 ЕЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Давидун М.В.				Метод та мікропроцесорні засоби оперативного контролю стану водія Схема електрична принципова	Лім.	Арк.	Акрушів
Перевір.	Богомолов С.В.						106	108
Опонент.	Кательніков Д.І.					ВНТУ, 1КІ-23м		
Н. Контр.	Швец С.І.							
Затверд.	Азаров О.Д.							

Перелік елементів

[illegible]

ДОДАТОК Е

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Метод та мікропроцесорні засоби оперативного контролю стану водія

Тип роботи: Магістерська кваліфікаційна робота

(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (зазначити))

Підрозділ кафедра обчислювальної техніки

(кафедра, факультет (інститут), навчальна група)

Керівник Богомолов С.В., доц. каф. ОТ

(прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	82%
Загальна схожість	18%

Аналіз звіту подібності (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор Давидун М.В.

(підпис)

(прізвище, ініціали)

Опис прийнятого рішення

Особа, відповідальна за перевірку Захарченко С.М.

(підпис)

(прізвище, ініціали)

Експерт

(за потреби)

(підпис)

(прізвище, ініціали, посада)