

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних наук

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Методи машинного навчання в задачах розпізнавання міміки
обличчя

Виконала:

студентка

спеціальності

гру
VI курсу, пи СНмз-61

122 Комп'ютерні науки

(шифр і назва спеціальності)

Керівник

Нормоконтроль

Завідувач кафедри

Рецензент

Голубко К.А.

(прізвище та ініціали)

Литвиненко

Я.В.

(прізвище та ініціали)

Дуда О.М.

(прізвище та ініціали)

Боднарчук І.О.

(прізвище та ініціали)

Мудрик І.Я.

(прізвище та ініціали)

Тернопіль, 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 22 » грудня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки

(шифр і назва спеціальності)

студенту

Голубко Катерині Анатоліївні

(прізвище, ім'я, по батькові)

1. Тема роботи Методи машинного навчання в задачах розпізнавання міміки обличчя

Керівник роботи Литвиненко Ярослав Володимирович доктор технічних наук, професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » листопада 2025 року № 4-7-1139

2. Термін подання студентом завершеної роботи 22 грудня 2025 р.

3. Вихідні дані до роботи наукові публікації та аналітичні матеріали з питань комп'ютерного зору, аналізу міміки обличчя та мікровиразів, технічна документація сучасних бібліотек і фреймворків комп'ютерного зору, нормативні матеріали щодо захисту персональних і біометричних даних

4. Зміст роботи (перелік питань, які потрібно розробити)

1) Дослідити проблему мімічного перенапруження обличчя під час тривалої роботи за комп'ютером та проаналізувати сучасні підходи до його автоматизованого виявлення.

2) Проаналізувати методи комп'ютерного зору та машинного навчання, що застосовуються для детекції мімічних напружень, і обґрунтувати вибір відповідної моделі.

3) Спроекувати архітектуру десктопної програмної системи для моніторингу мімічної активності користувача.

4) Реалізувати, протестувати та оцінити ефективність програмного прототипу системи детекції мімічного перенапруження.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Високорівнева діаграма архітектури програмного рішення

Слайди що демонструють інтерфейс рішення

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Гурик О. Я., доцент	25.11.2025	
Безпека в надзвичайних ситуаціях	Теслюк В. М., доцент	25.11.2025	

7. Дата видачі завдання 17.11.2025

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Голубко К.А.

(прізвище та ініціали)

Керівник роботи

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка десктопної системи для детекції мімічних напружень на основі комп'ютерного зору // Кваліфікаційна робота освітнього ступеня «Магістр» // Голубко Катерина Григорівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2025 // С. 63, рис. – 14, табл. – 17, кресл. – 17, додат. – 2, бібліогр. – 50.

Ключові слова: комп'ютерний зір, мімічне перенапруження, детекція, глибинне навчання, інформаційна система, об'єктна детекція, аналіз мікрОВИРАЗІВ.

Кваліфікаційна робота присвячена розробленню десктопної інформаційної системи для автоматичної детекції мімічного перенапруження із використанням методів комп'ютерного зору. У першому розділі наведено аналіз предметної області, досліджено природу мімічних напружень, оглянуто існуючі підходи до аналізу мікрОВИРАЗІВ та локальних змін текстури. У другому розділі проведено огляд і порівняння сучасних моделей глибинного навчання, обґрунтовано вибір алгоритму об'єктної детекції та розроблено архітектуру системи Frownbully. У третьому розділі здійснено реалізацію програмного прототипу, подано результати тестування та обчислювального експерименту.

Об'єкт дослідження: процеси комп'ютерного аналізу мімічних напружень. Предмет дослідження: методи детекції локальних мімічних заломів та інструменти побудови інформаційних систем моніторингу.

ANNOTATION

Development of a Desktop System for Facial Strain Detection Based on Computer Vision // Master's Degree Qualification Work // Kateryna Holubko // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNm-61 group // Ternopil, 2025 // P. 63, fig. – 14, tables – 17, posters – 17, annexes – 2, ref. – 50.

Key words: computer vision, facial strain, detection, deep learning, information system, object detection, micro-expression analysis.

The thesis is devoted to the development of a desktop information system for automatic detection of facial muscular strain using computer vision techniques. The first chapter presents the analysis of the subject area, the nature of facial tension, and existing approaches to micro-expression and texture-change analysis. The second chapter investigates and compares modern deep learning models, substantiates the choice of an object detection algorithm, and presents the architecture of the Frownbully system. The third chapter describes the implementation of the software prototype and provides the results of testing and computational experiments.

Object of research: processes of computer-based analysis of facial strain.
Subject of research: methods for detecting local facial wrinkles and tools for building monitoring information systems.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

База даних, БД (англ. Database) – сховище системи Frownbully для результатів детекції, шляхів до зображень і статистики.

Бандлінг (англ. Bundling) – процес об'єднання модулів та залежностей у дистрибутив.

Бекенд (англ. Backend) – серверна логіка системи, що обробляє дані та виконує інференс.

Баундінг бокс (англ. Bounding box) – прямокутна рамка, що обмежує область знайденого об'єкта.

Детекція (англ. Detection) – виявлення об'єктів на зображенні.

Фронтенд (англ. Frontend) – клієнтська частина застосунку, інтерфейс користувача.

Інференс (англ. Inference) – процес виконання моделі для отримання прогнозу.

Згорткові блоки (англ. Convolutional Blocks) – структури CNN, що поєднують згортки, активації та нормалізації.

Комп'ютерний зір (англ. Computer vision) – галузь, що вивчає методи автоматичного аналізу зображень.

Механізми Уваги (англ. Attention mechanisms) – механізми нейронних мереж, що динамічно виділяють релевантні просторові або каналові ознаки.

Поріг чутливості (англ. Sensitivity threshold) – параметр, що визначає момент сповіщення про перенапруження.

Прунінг (англ. Pruning) – оптимізація моделі шляхом видалення зайвих параметрів.

Сповіщення, Нотифікація (англ. Notification) – повідомлення користувачу про виявлене перенапруження.

API-gateway pattern (англ. API-gateway pattern) – архітектурний шаблон, що забезпечує єдину точку входу для клієнтів у багатокомпонентній системі.

ASGI-сервер (англ. ASGI server) – сервер, що реалізує протокол Asynchronous Server Gateway Interface для асинхронних Python-застосунків.

Browser-first (англ. Browser-first) – підхід до розроблення з пріоритетом браузерного середовища.

CNN, Згорткова нейронна мережа (англ. Convolutional Neural Network) – модель глибинного навчання, заснована на згорткових операціях.

CPU, Центральний процесор (англ. Central Processing Unit) – процесор, що виконує інференс у десктопній версії системи.

Desktop-first (англ. Desktop-first) – підхід до розробки інтерфейсів із пріоритетом настільних систем.

Edge-inference (англ. Edge inference) – виконання інференсу безпосередньо на локальному пристрої.

Facial analysis (англ. Facial analysis) – комплекс алгоритмів для аналізу обличчя та мімічних змін.

FastAPI (англ. FastAPI) – високопродуктивний Python-фреймворк для побудови API.

Few-shot learning (англ. Few-shot learning) – метод навчання моделей за мінімальної кількості прикладів.

Fine-tuning (англ. Fine-tuning) – перенавчання попередньо натренованої моделі на нових даних.

FPS, Кадри за секунду (англ. Frames Per Second) – кількість кадрів, які система здатна обробляти за секунду.

GPU (англ. Graphics Processing Unit) – процесор для прискорення тренування та інференсу.

IoU, Коефіцієнт перетину (англ. Intersection over Union) – метрика збігу передбаченої та справжньої рамки.

JSON (англ. JavaScript Object Notation) – формат обміну структурованими даними.

ML, Машинне навчання (англ. Machine Learning) – підгалузь штучного інтелекту, на якій базується модель.

ONNX Runtime (англ. ONNX Runtime) – рушій для швидкого виконання моделей у форматі ONNX.

OpenCV (англ. Open Source Computer Vision Library) – бібліотека інструментів комп'ютерного зору.

PyTorch із CUDA (англ. PyTorch with CUDA) – середовище глибинного навчання з GPU-прискоренням.

PyWebView (англ. PyWebView) – бібліотека для створення десктопних додатків із вебінтерфейсом.

Single source of truth (англ. Single source of truth) – принцип зберігання даних в одному достовірному джерелі.

SSD (англ. Single Shot Detector) – однопрохідна модель для детекції об'єктів.

TensorRT (англ. TensorRT) – фреймворк NVIDIA для прискорення інференсу.

Tension-Type Headache, Головний біль напруження (англ. Tension-Type Headache) – первинний головний біль, пов'язаний із м'язовим перенапруженням.

Transfer learning (англ. Transfer learning) – перенесення знань із попередньо натренованої моделі на нову задачу.

YOLO (англ. You Only Look Once) – сімейство моделей детекції, що працюють у режимі реального часу.

ЗМІСТ

ВСТУП.....	12
РОЗДІЛ 1. ТЕОРЕТИКО-АНАЛІТИЧНИЙ ОГЛЯД ПРОБЛЕМИ ТА ПІДХОДІВ ДО ДЕТЕКЦІЇ МІМІЧНИХ НАПРУЖЕНЬ.....	15
1.1. Аналіз проблеми неконтрольованого напруження м'язів під час роботи за комп'ютером та її наслідків.....	15
1.1.1 Існуючі нетехнологічні способи розв'язання проблеми зморшок.....	18
1.2. Огляд технологій моніторингу обличчя.....	19
1.2.2. Facial landmarks (dlib, MediaPipe) як підхід для визначення активності м'язів	20
1.2.3. Аналіз micro-expressions (мікровиразів обличчя)	23
1.2.4. Комерційні та наукові системи моніторингу обличчя	25
1.2.5 Моделі комп'ютерного бачення на основі детекції об'єктів	28
1.3. Огляд сучасних рішень wrinkle/frown detection	31
РОЗДІЛ 2 — ТЕОРЕТИЧНІ ЗАСАДИ ПРОЄКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ МІМІЧНИХ НАПРУЖЕНЬ.....	35
2.1 — Вимоги до системи моніторингу зморшок.....	35
2.1.1 Короткий опис продукту і постановка проблеми.....	35
2.1.3 Цілі та не-цілі.....	36
2.1.4 User Stories / Сценарії використання.....	37
2.1.5 Нефункціональні вимоги	39
2.1.6 Вимоги до інтерфейсу.....	41
2.1.7. Ризики та обмеження, а також стратегії їх мінімізації	43
2.2 Методи та моделі, використані в системі	45
2.3 Об'єктна детекція як підхід.....	45
2.4. Теоретичне обґрунтування архітектурних рішень.....	50
2.5 Забезпечення приватності.....	55
2.5. Архітектура системи	58

РОЗДІЛ 3 — ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ	63
3.1 Архітектура та основні компоненти розробленої системи FrownBully.....	63
3.2 Потік даних та взаємодія компонентів системи.....	66
3.3. Обмеження системи та перспективи розвитку.....	68
3.3 Рекомендації для майбутніх версій.....	73
3.3.1 Покращення моделі та ML-складової	73
3.3.2 Інференс та архітектура: напрями розвитку	76
Висновки до розділу 3.....	78
РОЗДІЛ 4 — ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	80
4.1. Фактори виробничого середовища і їх вплив на життєдіяльність людини	80
4.2 Психоемоційна безпека та профілактика професійного стресу при роботі з інтелектуальними системами	83
4.3. Безпека користувача під час експлуатації програмної системи ...	85
ВИСНОВКИ.....	88
ПЕРЕЛІК ДЖЕРЕЛ	90
Додаток А	99
Додаток Б	100
Додаток В.....	102

Актуальність теми. У сучасних умовах активного використання комп'ютерів та цифрових робочих середовищ значно зростає навантаження на зорову систему та мимічні м'язи обличчя. Мимовільне перенапруження лобових і міжбрівних м'язів — насуплення міжбрів'я, мікрозалом шкіри — часто відбувається непомітно для користувача, але сприяє формуванню стійких мимічних звичок, появі зморшок, підвищенню рівня стресу та виникненню головного болю напруження. Водночас люди не здатні самотійно контролювати частоту та інтенсивність таких проявів, що обумовлює потребу у технологічних інструментах для їх автоматичного виявлення.

Останні досягнення в галузі комп'ютерного зору та глибинного навчання відкривають можливості для точного аналізу мимічної активності в реальному часі. Однак більшість комерційних рішень орієнтовані на розпізнавання емоцій або використовують закриті алгоритми, що ускладнює розуміння принципів роботи та адаптацію таких систем під специфічні завдання. Тому створення інформаційної системи, здатної детектувати мимічне перенапруження на основі локальних змін текстури обличчя, є актуальним та перспективним напрямом наукових досліджень у галузі комп'ютерного зору та цифрового здоров'я.

Мета і задачі дослідження.

Метою кваліфікаційної роботи є підвищення ефективності та інформативності аналізу мимічного перенапруження шляхом розроблення десктопної системи для його автоматичної детекції із використанням методів комп'ютерного зору. Для досягнення мети необхідно виконати такі завдання:

- проаналізувати сучасні наукові дослідження у сфері комп'ютерного аналізу миміки та мікровиразів;
- дослідити актуальні моделі глибинного навчання та підходи до локальної детекції зморшок і мимічних заломів;
- проаналізувати алгоритми об'єктної детекції та визначити модель, оптимальну для виявлення мимічного перенапруження;
- розробити архітектуру програмної системи Frownbully;

- реалізувати прототип, що виконує обробку зображення, прийняття рішення щодо сповіщення та статистичний аналіз мімічної активності;

Об'єкт дослідження — процеси аналізу та комп'ютерної інтерпретації мімічних напружень обличчя.

Предмет дослідження — методи детекції локальних мімічних заломів та технології побудови інформаційних систем для моніторингу мімічної активності.

Наукова новизна одержаних результатів полягає в тому, що розроблено підхід до детекції мімічного перенапруження на основі використання моделі об'єктної детекції, адаптованої до аналізу текстурних змін у зоні міжбрів'я. Запропонована система забезпечує можливість інтерпретації мікрозаломів як ознак м'язового гіпертонусу в реальному часі.

Практичне значення одержаних результатів полягає у створенні працюючого програмного прототипу Frownbully, який здійснює моніторинг мімічної активності, формує статистичний профіль користувача та забезпечує механізм превентивного інформування про надмірну напругу. Прототип може бути використаний у контекстах цифрової ергономіки, профілактики перенапруження та формування здорових звичок роботи за комп'ютером.

Апробація результатів магістерської роботи.

Основні положення дослідження були представлені та обговорені на XIV Міжнародній науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій» Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 2025 р.) а також на XIII науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 2025 р.)

Публікації.

Основні результати дослідження викладено у двох публікаціях у матеріалах конференцій (див. додаток В).

Структура й обсяг кваліфікаційної роботи.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 55 найменувань і трьох додатків. Загальний обсяг

становить 99 сторінок, з них 70 сторінок основного тексту, що містить 6 рисунків, 2 таблиці і 3 додатки.

РОЗДІЛ 1. ТЕОРЕТИКО-АНАЛІТИЧНИЙ ОГЛЯД ПРОБЛЕМИ ТА ПІДХОДІВ ДО ДЕТЕКЦІЇ МІМІЧНИХ НАПРУЖЕНЬ

Станом на 2025 рік середня людина проводить перед екраном 6 годин 45 хвилин на день, що становить приблизно 47 годин 15 хвилин на тиждень. Порівняно з 2013 роком, середній екранний час дорослих зріс на 8,4 %. Під час сконцентрованої роботи людина часто перенапружує м'язи обличчя не помічаючи цього, в цьому розділі ми розглянемо наслідки такого напруження та існуючі методи його виявлення.

1.1. Аналіз проблеми неконтрольованого напруження м'язів під час роботи за комп'ютером та її наслідків

Проблема неконтрольованого напруження м'язів лиця є досить обширною адже викликає як медичні так і соціо-економічні наслідки. Така активність може виражатись у нахмуренні міжбрівних м'язів або піднятті брів також можуть виражатись як стискання щелеп, примруження очей. Ця поведінка найчастіше пов'язана із психофізіологічною напругою.

Поглиблює проблему також м'язова пам'ять — здатність м'язів автоматично повторювати вже знайомі рухи без свідомого контролю. У контексті обличчя це означає що людина може несвідомо насуплювати лоб або хмурити брови навіть якщо не має емоційної причини бо її мімічні м'язи звикли до певного патерну.

Дослідження показують, що частіше за все людина хмурить брови при стресі, концентрації, роботі за комп'ютером, ці рухи відбуваються автоматично та непомітно для самої людини. Комплекс факторів призводить до хронічного насуплення під час концентрації і спричиняє як естетичні наслідки так і певні проблеми із здоров'ям.

Надмірне та постійне напруження м'язів обличчя, зокрема лобового м'яза, м'язів міжбрів'я (*procerus*, *corrugator supercilii*) та *orbicularis oculi* (Рис 1.1) —

може спричиняти хронічну м'язову напругу у зоні чола та навколо очей, головні болі напруження [1].

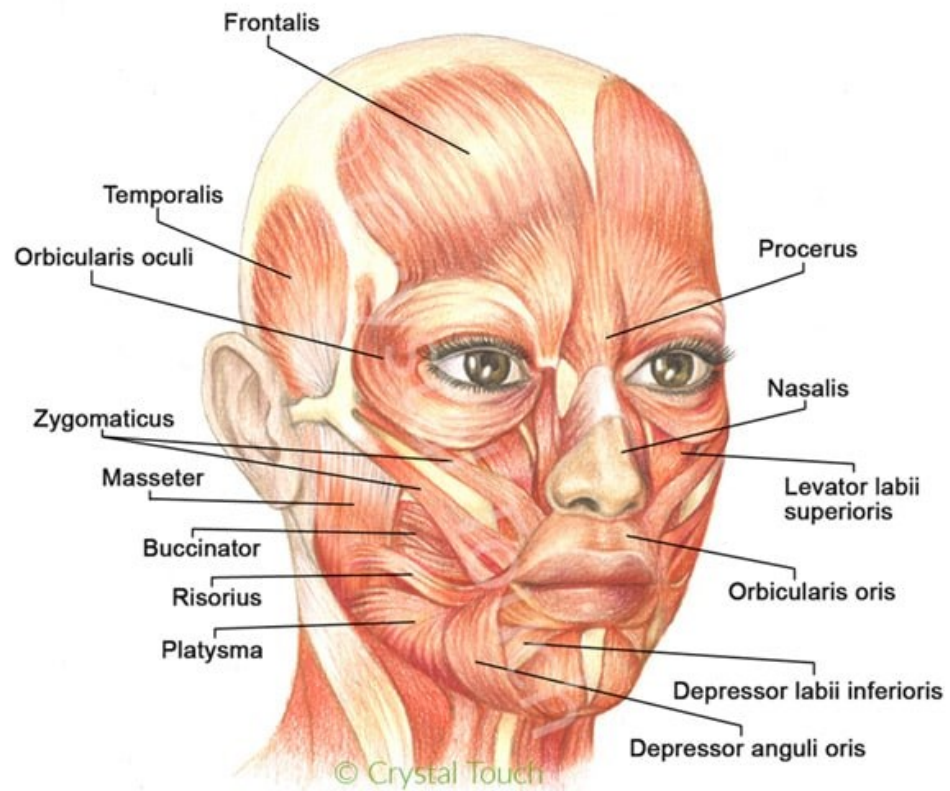


Рисунок 1.1 - Анатомія м'язів обличчя людини

За даними наукової літератури, головний біль напруження (Tension-Type Headache, ТТН) спричиняє більшу втрату працездатності та більше пропущених робочих днів, ніж мігрень, попри те, що часто вважається менш тяжким станом. Це зумовлено його поширеністю та хронічним перебігом. Особливістю ТТН є те, що він має багатофакторну етіологію та може співіснувати з іншими типами головного болю, що ускладнює діагностику та лікування.

ТТН тісно пов'язаний зі стресом, неправильною поставою та підвищеним м'язовим напруженням у ділянці голови та шиї. Його виникнення пояснюється як периферичними, так і центральними механізмами. У патогенезі розглядають:

- підвищену чутливість центральної нервової системи, насамперед структур трійчастої нервової системи;

- хронічне скорочення м'язів голови, шиї та обличчя, що призводить до подразнення периферичних рецепторів і формування болю.

Таким чином, напруження м'язів обличчя — зокрема лобового м'яза, *m. corrugator supercilii* та *m. procerus* — може виступати одним із тригерів ТНН. Тривале м'язове напруження сприяє переходу гострого болю в хронічний, утворюючи “порочне коло” стрес → м'язовий спазм → біль → подальше напруження. Це безпосередньо впливає на якість життя, рівень працездатності та психологічний стан.

Додатково до проблем фізичного самопочуття людина що має звичку хмуритись стикається із проблемою мімічних зморшок, що з першого погляду має здаватись менш значущою, але має суттєвий вплив на життя людей тож на нашу думку вартує уваги. Проблема появи мімічних зморшок була важливою для людства упродовж усієї історії й залишається актуальною сьогодні. Це зумовлено тим, що зовнішність є одним із ключових факторів соціального сприйняття. Дослідження у сфері соціальної психології підтверджують, що привабливі, доглянуті люди отримують більше довіри, їм приписують вищий інтелект, доброзичливість та компетентність. Це явище отримало назву “ефект ореолу” (halo effect) [2].

Наукові дані показують що людей, яких вважають фізично привабливими, оцінюють як більш успішних, доброзичливих і соціально адаптованих. Візуальна привабливість впливає на кар'єрні результати, співбесіди та соціальні контакти. “Ефект ореолу” систематично змінює судження людей незалежно від реальних здібностей

Таким чином, зменшення мімічних зморшок та підтримання доглянутого вигляду мають не лише естетичне, а й соціально-психологічне значення. Люди прагнуть виглядати молодшими та привабливішими, адже це безпосередньо впливає на те, як їх сприймає оточення і покращує їх рівень життя.

Другим важливим чинником є зростання тривалості життя. За даними Всесвітньої організації охорони здоров'я, середня тривалість життя продовжує збільшуватися, і люди старшого віку значно довше залишаються активними у професійній та соціальній сферах [3].

Це формує природне бажання підтримувати молодий та здоровий зовнішній вигляд навіть у старшому віці. Відповідно, попит на засоби боротьби зі зморшками постійно зростає, а ринок косметологічних та дерматологічних послуг стабільно розширюється.

1.1.1 Існуючі нетехнологічні способи розв'язання проблеми зморшок

Для боротьби з мімічними та віковими зморшками тривалий час використовувалися різні методи, які можна умовно поділити на косметологічні, медичні та фізичні. Попри поширення, усі ці підходи мають суттєві обмеження або ризики.

Ін'єкційний метод знерухомлення м'язів використовує ботулотоксин (відомий під назвою бренду «Ботокс») та філери. Токсин блокує активність м'язів, тим самим вирівнює зморшки і запобігає головним болям пов'язаним із перенапруженням заблокованого м'язу, але існують певні ризики, такі як Опущення повік (птоз): Найпоширеніший серйозний косметичний ризик. Виникає, якщо токсин потрапляє у м'яз-підіймач повіки. Опущення брів: Брови можуть опуститися, якщо препарат введено занадто низько в м'язи чола. токсин може поширитися за межі зони ін'єкції, викликаючи: Ускладнене дихання або ковтання (дисфагія), сильну м'язову слабкість, що поширюється на інші частини тіла, порушення зору (двоїння, розмитість) [4].

Філери (гіалуронова кислота) заповнюють зморшки, але ризики включають набряки, фіброз, алергічні реакції. При цьому філери не вирішують проблему перенапруження м'язів і відповідно не вирішують проблему головного болю [5].

Існують також фізичні та альтернативні методи, такі як тейпування обличчя, анти зморшкові наліпки, масаж.

Тейпування використовується для розслаблення м'язів і зменшення напруги. Недоліки такого методу полягають в тому що ефективність цього методу не підтверджена науковими дослідженнями, результат короточасний і також його неможливо використовувати і професійному середовищі адже тейп видимий на обличчі [6].

Анти зморшкові наклейки (face patches) фіксують шкіру, перешкоджаючи створенню зморшок уночі. Але ефективність таких наліпок також не є дослідженою і вони навіть теоретично попереджають насуплення тільки під час сну а не під час роботи за комп'ютером.

Масажі та самомасажі допомагають тимчасово покращити кровообіг і тонус шкіри, допомагають розслабити перенапружені м'язи лица, але також не впливають на звичку насуплювати лоб.

Головний недолік усіх нетехнологічних методів - вони усувають, або намагаються усунути, наслідок, але не працюють із причиною — неконтрольованою активністю лицьових м'язів та мімічними звичками.

Саме тут з'являється потреба у технологічних рішеннях, які:

- відстежують міміку в реальному часі,
- допомагають формувати нові корисні звички,
- попереджають перенапруження,
- оптимізують зовнішній вигляд неінвазійно.

Це й обґрунтовує актуальність створення системи для розпізнавання мімічних зморшок та попередження їх формування.

1.2. Огляд технологій моніторингу обличчя

Сучасні технології моніторингу мімічної активності ґрунтуються на різних підходах комп'ютерного зору, кожен з яких по-різному відображає структуру, рух і текстуру обличчя. Перші методи орієнтувалися на класичні алгоритмічні принципи обробки зображень, такі як детекція контурів та аналіз текстур. Алгоритм Canny, градієнтні оператори на кшталт Sobel чи Prewitt, а також фільтри Габора дозволяли виділяти локальні елементи шкіри — дрібні лінії, заломы, різкі переходи тону, що часто виникають у місцях регулярної мімічної активності [7][8]. Ці підходи забезпечували швидкість роботи та простоту реалізації, однак залежали від зовнішнього освітлення та не давали можливості оцінювати динаміку м'язових рухів. Вони дозволяли зафіксувати лише наслідок

перенапруження — сформовану лінію чи зморшку — не даючи інформації про сам процес формування мімічного патерну.

Переваги цих методів полягають в: 1) дуже високій швидкості роботи; низьких вимогах до ресурсів; чутливості до дрібних текстур (зморшок, ліній, мікро заломів).

Класичні методи комп'ютерного зору, такі як Canny, Sobel, Prewitt чи фільтри Габора, оперують виключно низькорівневими ознаками зображення — градієнтами, різкими переходами яскравості або текстурними патернами. Вони здатні виділити лінії, контури або локальні неоднорідності шкіри, але не надають жодної інформації про те, що саме ці лінії означають. З точки зору таких алгоритмів, вертикальна міжбрівна зморшка, тінь від волосся, відблиск монітора або нерівність шкіри — це однакові градієнтні структури [9].

Саме тому класичні фільтри не можуть бути використані як основа для моніторингу мімічного перенапруження. Вони не розрізняють, чи утворилась лінія внаслідок скорочення *m. corrugator supercilii*, чи вона пов'язана з освітленням, оптикою камери чи особливостями шкіри. У результаті система, побудована на таких методах, генерувала б значну кількість хибних спрацьовувань і не мала б здатності інтерпретувати побачені структури у контексті мімічної активності.

1.2.2. Facial landmarks (dlib, MediaPipe) як підхід для визначення активності м'язів

Подальший розвиток комп'ютерного зору привів до широкого застосування моделей, що аналізують ключові точки обличчя — так званих *facial landmarks*. Підходи, реалізовані у бібліотеках *dlib* та *MediaPipe* [10], будують високоточну карту орієнтирів обличчя і дають змогу відстежувати рухи брів, очей, рота та чола з високою частотою. На відміну від класичних текстурних методів, *landmarks* дозволяють працювати з динамічними змінами та фіксують моментальний стан м'язової активності. (Рис 1.2)

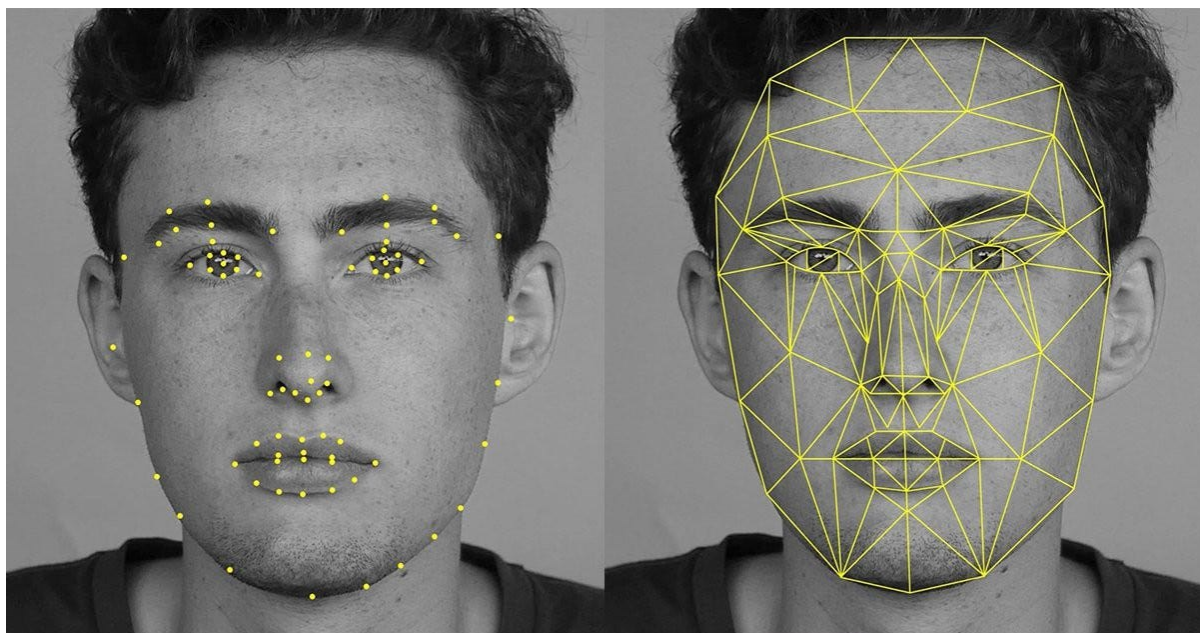


Рисунок 1.2 – візуалізація роботи Facial landmark

Найпоширеніші технологіями є dlib (68 landmark model) — класичний підхід, добре працює на статичних фото та MediaPipe Face Mesh (468 точок) — висока точність, робота в реальному часі, низька затримка, відмінно підходить для UI-додатків [11].

Перевагами такого методу є те що він ідеально підходить для реального часу, дозволяє відслідковувати динаміку м'язів, а не лише зміну текстури, незалежний від віку та стану шкіри та працює навіть у середньому освітленні.

Але існує і ряд недоліків. Першою ключовою проблемою є той факт, що алгоритми facial landmarks працюють виключно з геометричною конфігурацією обличчя. Тобто їхнім предметом аналізу є переміщення та положення спеціально визначених контрольних точок. Зморшки, дрібні заломы та локальні структурні зміни поверхні шкіри не впливають на положення цих точок, а відтак залишаються невидимими для такого підходу. Іншими словами, моделі landmark-детекції не оперують текстурними характеристиками і не аналізують стан шкіри між орієнтирами, що є критично важливим для виявлення мимічного перенапруження [12].

Другою суттєвою особливістю є відсутність у landmark-методів можливості семантичної класифікації локальних мимічних патернів. Якщо моделі об'єктної детекції здатні розпізнавати конкретні типи зморшок (наприклад, лобна

зморшка, міжбрівна зморшка, гусячі лапки), то landmarks можуть лише фіксувати зміщення точок, не додаючи інтерпретації, що саме це зміщення означає. Відтак вони не визначають ані наявності зморшки, ані її типу, ані пов'язаності з активністю конкретних м'язових груп.

Окрім того, рух мімічних м'язів не завжди супроводжується значущим переміщенням ключових точок. Наприклад, незначне скорочення *m. corrugator supercilii* може спричинити вертикальний залом у міжбрів'ї без помітних змін у позиції брів. У такому випадку landmarks не фіксують жодної мімічної активності, хоча фактично відбувається локальне перенапруження м'язу, що має практичне значення для даної роботи. Об'єктна детекція, навпаки, ідентифікує саму структуру залому як семантичний об'єкт.

Важливим є також те, що facial landmarks орієнтовані на макроструктури обличчя, тоді як дрібні зморшки є мікроструктурними змінами текстури, які не впливають на глобальну геометрію. Через це модель, побудована лише на landmarks, не може забезпечити ані точності, ані інформативності, необхідної для розпізнавання саме результатів мімічного перенапруження.

Нарешті, у контексті побудови системи моніторингу перенапруження важливою є здатність алгоритму генерувати структуровані дані про наявність конкретних ознак. Об'єктні детектори повертають координати, клас та ймовірність виявлення, що дозволяє аналізувати повторюваність певних типів зморшок у часі. Натомість landmarks повертають лише набір координат, які самі по собі не містять семантичної інформації про зморшки чи локальні прояви м'язової активності.

У сукупності ці характеристики роблять методи facial landmarks непридатними для задачі детекції мімічного перенапруження, коли його ключовим проявом є формування локальних заломів або зморшок. Саме тому в межах цього дослідження обрано модель об'єктної детекції, здатну працювати з текстурними та семантичними ознаками шкіри, що забезпечує значно вищу точність і практичну корисність у порівнянні з підходами, заснованими на геометричному аналізі обличчя.

1.2.3. Аналіз micro-expressions (мікроекспресій обличчя)

Методи розпізнавання мікро виразів обличчя (micro-expressions) становлять важливий напрям афективних обчислень і спрямовані на виявлення надшвидких, часто несвідомих мімічних реакцій, що тривають менш ніж 0,2 секунди. Такі реакції зазвичай відображають внутрішній емоційний стан людини та класифікуються на основі системи FACS (Facial Action Coding System), яка описує активність окремих м'язових одиниць (Action Units).

Одним із методів аналізу мікро виразів обличчя є Facial Action Coding System (FACS) — система AU (action units), що пов'язує мімічні рухи із конкретними м'язами (Рис 1.3)

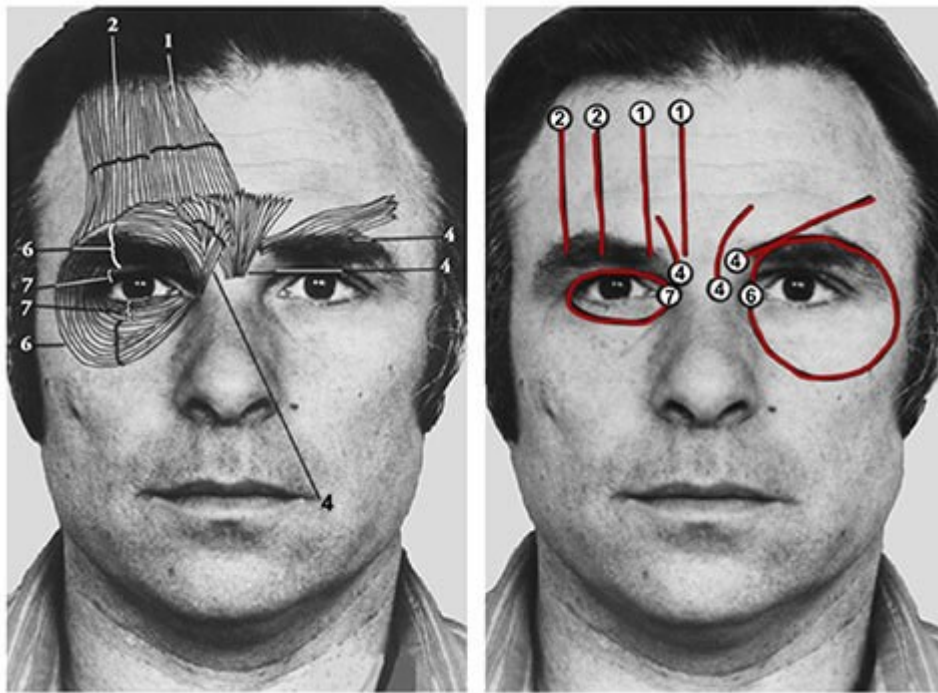


Рис 1.3 – візуалізація роботи FACS

Також використовуються CNN/LSTM моделі (для глибокого аналізу) та Temporal interpolation для прискорених мімічних реакцій.

Цей підхід дозволяє виявляти несвідомі мімічні реакції безпосередньо пов'язані з активацією певних м'язових груп і дають змогу оцінювати стрес і внутрішню напругу [13].

Але, незважаючи на високу наукову цінність цього підходу, його застосування для задачі моніторингу мімічного перенапруження, що проявляється у вигляді локальних зморшок, має низку суттєвих обмежень.

Першою фундаментальною проблемою є те, що мікровирази — це динамічне явище, яке існує лише в дуже короткі часові проміжки. Для його коректного аналізу необхідне відео високої частоти (60–200 кадрів на секунду), а також спеціальні алгоритми, здатні аналізувати мімічні зміни в часовій площині. Натомість у цьому дослідженні використовується покадровий аналіз зображень, отриманих кожні дві секунди, що не дозволяє зафіксувати мікро вираз як такий. Мікро вирази не залишають стійкої текстурної ознаки на шкірі, а тому не можуть бути виявлені у статичному зображенні. [14]

Другою очевидною проблемою є те, що методи розпізнавання мікровиразів спрямовані на аналіз емоційних реакцій, а не структурної зміни шкіри. Вони визначають активність м'язів крізь рух та деформацію обличчя, а не через текстурні прояви, які є результатом тривалого або повторюваного перенапруження. Отже, детекція мікровиразів дозволяє виявити емоційну реакцію, але цей метод не призначений для визначення наявності зморшки чи залому, що є ключовим індикатором мімічного гіпертонусу у даному проєкті [15].

Третє обмеження полягає у високому рівні шуму та складності інтерпретації. Мікро вирази ґрунтуються на аналізі дуже тонких рухів, які часто не залишають видимих змін на поверхні шкіри. Вони можуть мати мінімальний амплітудний вплив на текстуру чола або міжбрів'я, а тому не дають інформації про наявність або відсутність локального перенапруження м'язів. У випадках, коли перенапруження вже проявилось у вигляді видимого залому, методи детекції мікрівирізів не дають додаткової інформації, оскільки не спрямовані на аналіз структури зморшок.

Важливою особливістю є також те, що моделі детекції мікровиразів не оперують семантичними класами локальних текстур. Вони класифікують мімічні рухи як емоційні стани (наприклад, гнів, здивування, страх), але не визначають наявності конкретних типів мімічних ліній. Таким чином, навіть у разі наявності мімічної активності такі підходи не можуть ідентифікувати структурні ознаки

перенапруження, необхідні для побудови статистики чи аналізу патернів поведінки користувача.

Окремо варто зазначити, що методи розпізнавання мікроекспресій мають надзвичайно високі вимоги до якості даних. Вони чутливі до шумів, варіацій освітлення, часткових оклюзій та низької роздільної здатності. Дані з веб камери, які використовуються в цьому проєкті, не відповідають типовим вимогам для якісного аналізу, що суттєво знижує практичну можливість застосування такого підходу.

1.2.4. Комерційні та наукові системи моніторингу обличчя

Комерційні та промислові системи автоматичного аналізу обличчя, такі як FaceReader (Noldus), Affectiva (SmartEye), Microsoft Face API / Face SDK та Google MediaPipe Vision Tasks, є потужними інструментами, широко застосовуваними у психологічних дослідженнях, UX-тестуванні, емоційній аналітиці, вивченні поведінки споживачів та афективних обчисленнях,

FaceReader (Noldus, Нідерланди) є алгоритмом для аналізу виразів обличчя у реальному часі, визначення емоцій, AU, рухів брів, очей, губ. Це рішення використовується у психології, UX-дослідженнях. [16] Affectiva (SmartEye) – це одна з найпоширеніших систем емоційного розпізнавання. Вона виявляє мікроекспресії, напруження, напруження лобових м'язів, моргання. [17] Microsoft Face API / Face SDK детектує ключові точки, вирази обличчя та положення голови (Рис 1.4). [18] Google MediaPipe + Vision Tasks дуже точно детектує facial landmarks. А також є безкоштовною технологією реального часу. [19]

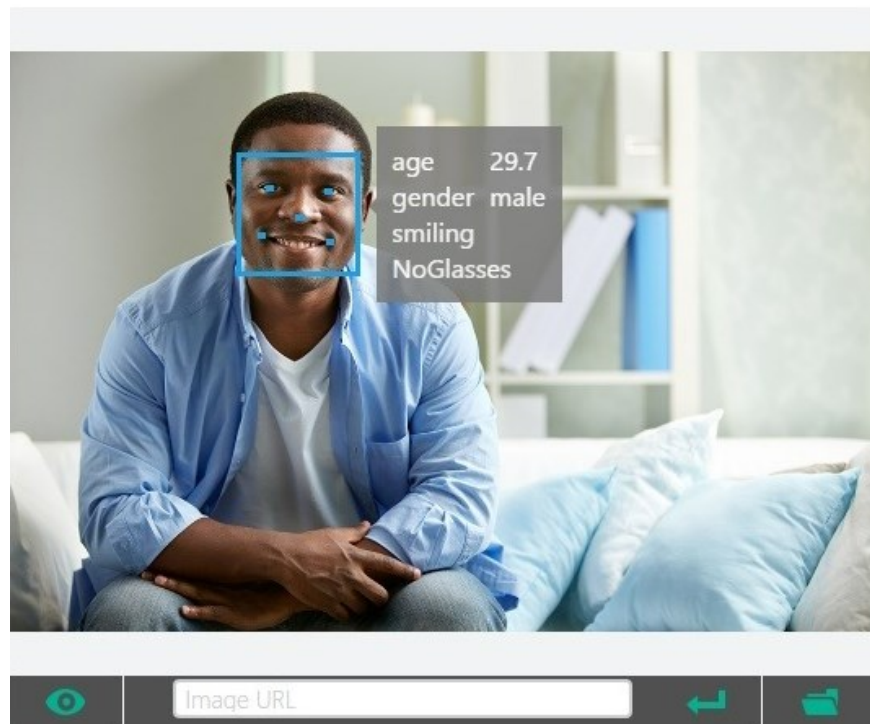


Рисунок 1.4 Приклад роботи Microsoft Face API

Ці рішення використовують власні моделі для оцінки інтенсивності скорочення м'язів, аналізу емоцій, визначення мікро виразів і є стандартом у дослідженнях поведінки, UX-аналізі та психології. Існування таких систем свідчить про актуальність і практичну значущість моніторингу міміки,

Попри їхню технологічну зрілість, вони не є придатними для задачі моніторингу мімічного перенапруження, яке проявляється у вигляді локальних зморшок та тонких текстурних змін шкіри, і мають низку фундаментальних обмежень у контексті даного проєкту.

Перш за все, ці системи орієнтовані на класифікацію емоційних станів та активності м'язів на основі їх руху, а не на аналіз поверхневих шкірних структур. FaceReader та Affectiva визначають інтенсивність емоцій через Action Units (AU) відповідно до FACS-моделі, але не аналізують і не локалізують зморшки як окремі об'єкти. Їх алгоритми оптимізовані під динаміку м'язових скорочень та деформацію обличчя, а не під текстурні прояви хронічного або повторюваного перенапруження. Таким чином, навіть за умови виявлення активності певних м'язів, ці системи не можуть встановити факт присутності структурного залому на шкірі — основного індикатора мімічного гіпертонусу у цьому дослідженні.

Другою суттєвою проблемою є закритість моделей і неможливість кастомізації. Ані FaceReader, ані Affectiva не дозволяють досліднику модифікувати модель або навчити її на спеціалізованому наборі даних, зокрема на зображеннях зі зморшками різного типу. Вони працюють виключно в межах заздалегідь визначеного функціоналу. Це унеможливорює додавання класів на кшталт “міжбрівна зморшка” чи “лобна зморшка” та інтеграцію таких моделей у систему, що потребує аналізу індивідуальних текстурних патернів шкіри.

Третім критичним обмеженням є висока вартість ліцензій та закритість API. FaceReader є дорогим професійним програмним забезпеченням, призначеним переважно для лабораторного використання, та не дозволяє запускати інференс локально в межах кастомного застосунку без повноцінної комерційної ліцензії. Affectiva, після інтеграції у SmartEye, також змінила ліцензійну політику і орієнтується на корпоративних клієнтів. Навіть якщо API доступний, модель платного використання часто робить використання таких сервісів економічно необґрунтованим у контексті персонального проєкту.

Microsoft Face API та Google MediaPipe Vision Tasks мають значно ширший спектр доступності, однак вони орієнтовані на розв’язання інших категорій задач: виявлення облич, визначення ключових точок, відстеження погляду, класифікацію емоцій або визначення віку та статі. Жоден із цих сервісів не підтримує детекцію зморшок як окремих об’єктів. Крім того, Microsoft Face API не надає доступу до локальних структурних ознак шкіри через політику приватності та етичні обмеження, оскільки дрібні текстурні характеристики потенційно можуть використовуватися для біометричної ідентифікації. MediaPipe Vision, своєю чергою, пропонує інструменти для сегментації, landmarks та загальної класифікації, але не містить моделей для аналізу мікротекстур шкіри.

Ще одним важливим фактором є непридатність цих систем для аналізу високочастотних дрібних текстурних елементів, які характерні саме для зморшок. FaceReader та Affectiva фокусуються на макроструктурних деформаціях (русі брів, змінах контуру рота), тоді як дрібні заломы, тонкі вертикальні чи горизонтальні лінії та локальні асиметрії структури шкіри не є

частиною їхнього аналітичного поля. Системи такого типу часто застосовують згладжувальні чи нормалізаційні алгоритми, які зменшують видимість дрібних деталей, що додатково ускладнює детекцію зморшок.

Нарешті, для інтеграції в реальну систему моніторингу, де необхідний доступ до конкретних координат локальних об'єктів, ці інструменти стають непрактичними. Вони повертають лише метадані про емоції, рівні активації AU або загальні параметри обличчя, але не надають структурованої інформації про локальні артефакти шкіри. У контексті цього проекту саме така інформація — bounding boxes та ймовірність виявлення конкретного типу зморшки — є ключовою.

У сукупності ці характеристики свідчать, що комерційні системи аналізу обличчя, попри свою зрілість та популярність, не відповідають специфічним цілям дослідження, яке спрямоване на моніторинг мімічного перенапруження через аналіз локальних змін текстури шкіри. Натомість використання моделі об'єктної детекції, натренованої на спеціалізованому датасеті зморшок, дає можливість отримати саме ті дані, які потрібні для побудови ефективної системи попередження мімічної гіперактивності та її наслідків.

1.2.5 Моделі комп'ютерного бачення на основі детекції об'єктів

Паралельно з цими напрямками розвивалися моделі об'єктної детекції, які аналізують зображення цілісно й шукають на них локальні об'єкти — у тому числі дрібні патерни шкіри. Архітектури родини YOLO, Faster R-CNN, RetinaNet або сегментаційні моделі, як-от U-Net, дозволяють точно знаходити локальні зміни текстури: мімічні лінії, заломы, міжбрівні зморшки, «гусячі лапки» чи лінії усмішки (Рис 1.5) [20]. На відміну від landmark-методів, які оцінюють положення м'язів, моделі об'єктної детекції працюють безпосередньо з візуальними проявами перенапруження, даючи змогу аналізувати не тільки рух, а й структурні зміни шкіри.

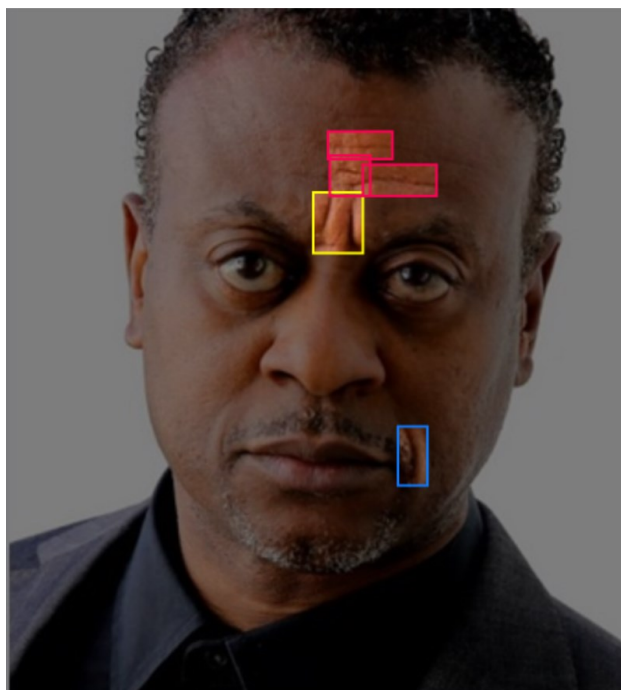


Рисунок 1.5 – Приклад візуалізації моделі детекції об’єктів

Вони поєднують здатність працювати з окремими статичними зображеннями та можливість інтерпретувати локальні структури за семантичними категоріями. Це робить їх придатними для побудови системи, що має на меті автоматичне виявлення мімічних патернів, пов’язаних із перенапруженням м’язів, навіть без аналізу руху у часовому вимірі.

На відміну від класичних фільтрів обробки зображень, таких як Canny, Sobel, Prewitt або фільтри Габора, моделі об’єктної детекції не обмежуються аналізом низькорівневих градієнтів або текстурних ознак. Об’єктна детекція, навпаки, формує семантичний опис цих структур: кожна зморшка визначається як окремий об’єкт із певним типом, що має змістовне значення для подальшого аналізу.

На противагу всім зазначеним методам, моделі об’єктної детекції демонструють збалансовану перевагу між точністю, гнучкістю та практичністю використання. Навіть при покадровому аналізі вони здатні виявляти та класифікувати локальні патерни текстури шкіри, надаючи структуровану інформацію у вигляді координат об’єкта, його класу та рівня ймовірності. [21] Це дозволяє формувати статистичні показники повторюваності певних типів зморшок, ідентифікувати мімічні звички користувача та встановлювати

індикатори перенапруження відповідних м'язових груп. Детектори об'єктів також демонструють більшу стійкість до варіативності освітлення, ракурсу та камерних артефактів, оскільки навчаються на великих та різномірних датасетах.

Важливою перевагою об'єктної детекції є її інваріантність до реальних умов експлуатації. Моделі, що пройшли навчання на великих різномірних датасетах, демонструють стійкість до змін освітлення, ракурсу, кольорового тону шкіри та технічних характеристик камери [22]. Для системи, яка повинна працювати на звичайному споживчому обладнанні без додаткових сенсорів, така властивість є критично важливою. Вона дозволяє забезпечити відтворюваність результатів у домашніх чи офісних умовах, де користувач постійно змінює положення голови та навколишнє освітлення.

Ще одним ключовим аргументом є можливість отримувати структуровані дані, необхідні для подальшого аналізу. Об'єктні детектори повертають координати виявленої зони, клас об'єкта та рівень впевненості моделі. На основі таких даних легко будувати часові ряди, визначати частоту появи конкретних типів зморшок та формувати статистичні профілі мімічної активності. Це забезпечує можливість довгострокового моніторингу та формування персоналізованих рекомендацій — функціональність, яку неможливо реалізувати без семантично структурованого виводу.

Важливо підкреслити й масштабованість підходу. Модель, що працює з об'єктами, може бути донавчена або розширена новими класами без необхідності радикальної зміни архітектури системи. Збільшення кількості типів зморшок або додавання нових зон обличчя потребуватиме лише розширення датасету і повторного навчання, що робить систему гнучкою та адаптивною. У контексті досліджень, де з часом можуть виникати нові гіпотези про патерни мімічної активності, це забезпечує методологічну стійкість [23].

Водночас об'єктна детекція має певні недоліки, які необхідно враховувати. Вона потребує значного обсягу навчальних даних, особливо якщо йдеться про тонкі та мало контрастні структури, якими є зморшки. Такі моделі також є більш ресурсоемними порівняно з класичними фільтрами. Однак у контексті цього дослідження використання попередньо навченої моделі дає можливість

нівелювати ці обмеження і орієнтуватися на практичну реалізацію системи моніторингу, а не на побудову фундаментального ML-компонента.

1.3. Огляд сучасних рішень wrinkle/frown detection

Сучасний ринок програмних рішень для аналізу обличчя представлений значною кількістю застосунків та сервісів, орієнтованих переважно на косметичний, естетичний або розважальний сегмент. До таких належать YouCam, FaceApp, SkinAI, BeautyCam, а також численні wellness-додатки, спрямовані на формування корисних звичок та загальний моніторинг стану здоров'я. Попри широке використання методів комп'ютерного зору та глибинного навчання, ці інструменти мають обмеження, що не дозволяють застосовувати їх у задачі аналізу мимічного перенапруження [24][25]. Розглянемо їхній функціонал за допомогою таблиці 1.1

Таблиця 1.1 – Порівняльний аналіз сучасних рішень детекції зморшок

Додаток	Детекція зморшок	Класифікація зморшок	Нотифікації	Реальний час	Призначення	Кастомізація
YouCam Makeup	Частково (поверхневий аналіз)	Ні	Ні	Так	Косметика, фільтри	Немає
FaceApp	Ні (ефекти на основі генерації)	Ні	Ні	Так	Розважальний функціонал	Немає
SkinAI	Так (загальний аналіз шкіри)	Ні	Ні	Так	Дерматологія, косметологія	Немає

Додаток	Детекція зморшок	Класифікація зморшок	Нотифікації	Реальний час	Призначення	Кастомізація
BeautyCam	Частково (згладжування дефектів)	Ні	Ні	Так	Камера з фільтрами	Немає
Wellness apps (Calm, FaceYoga, etc.)	Ні	Ні	Часто	Так	Здоров'я, медитація	Немає
FaceReader (Noldus)	Ні	Ні	Ні	Так	Психологія, UX	Закритий інструмент
Affectiva (SmartEye)	Ні	Ні	Ні	Так	Automotive + UX	Закрите рішення
Microsoft Face API	Ні	Ні	Ні	Так	Біометрія, аналітика	Немає зморшок
Google Vision + MediaPipe	Ні	Ні	Ні	Так	Загальний CV	Немає аналізу шкіри
AWS Rekognition	Ні	Ні	Ні	Так	Безпека, ідентифікація	Немає аналізу шкіри

Продовження таблиці 1.1

Додаток	Детекція зморшок	Класифікація зморшок	Нотифікації	Реальний час	Призначення	Кастомізація
Готові моделі (Huggingface/Roboflow API)	Так (залежить від датасету)	Так	Ні	Так	Детекція об'єктів	Так — можна створити свою модель

Більшість косметичних застосунків використовують алгоритми для покращення зображення, віртуального макіяжу або фоторетуші, зосереджуючись на модифікації зовнішності, а не на аналізі реальних структурних характеристик шкіри. Такі системи часто застосовують згладжування текстур, усуваючи дрібні недоліки, що, навпаки, призводить до втрати інформації про зморшки. Їхня мета — естетичне покращення та створення візуальних ефектів, а не моніторинг фізіологічних проявів мімічної активності. Додатки типу FaceApp використовують генеративні моделі для трансформації зовнішності, але вони не виконують достовірного аналізу мікроструктур шкіри; зморшки, які вони “визначають”, є результатом генеративних фільтрів, а не реальної інтерпретації вхідних зображень.

Додатки на кшталт SkinAI або BeautyCam застосовують алгоритми оцінки стану шкіри для косметологічних рекомендацій, але їхня точність орієнтована на загальні категорії, такі як рівень гладкості, жирності або рівномірність тону. Вони не розпізнають зморшки як окремі об’єкти конкретного типу, і тим більше не співвідносять їх із мімічними патернами або активністю конкретних м’язових груп. Wellness-додатки, спрямовані на контроль емоційності, стресу чи постави, також не здійснюють аналіз локальних змін поверхні шкіри і працюють у зовсім іншому домені даних.

Окремим напрямом є готові API для аналізу зображень, серед яких Roboflow, Microsoft FaceAPI, AWS Rekognition та Google Vision. Ці сервіси застосовуються в системах безпеки, аналітиці поведінки користувачів, біометричній ідентифікації або класифікації зображень. Проте жоден із них, не пропонує можливості детектити зморшки як окремі семантичні класи. Microsoft FaceAPI фокусується на емоційних станах, віці та основних рисах обличчя, але не здійснює локальний аналіз текстур шкіри. AWS Rekognition орієнтується на розпізнавання облич, ідентифікацію людей та аналіз загальних атрибутів. Google Vision має широкий спектр моделей, однак серед них відсутні інструменти, здатні розрізняти конкретні типи мімічних заломів [26].

У цьому контексті моделі об'єктної детекції займають окреме місце, оскільки дають можливість використовувати або створювати вузькоспеціалізовані моделі об'єктної детекції, що працюють зі зморшками як із повноцінними об'єктами. Використання платформ для тренування і хостингу моделей як Roboflow підтримує кастомні датасети, анотовані bounding-box, дозволяє тренувати моделі на дрібних локальних структурах та забезпечує API для їх подальшої інтеграції. Саме доступність моделей, орієнтованих на wrinkle/frown detection, робить її єдиним готовим інструментом, який може бути адаптований до задачі моніторингу мімічного перенапруження без необхідності створення повної ML-інфраструктури.

Проте навіть використання готової моделі не знімає потреби у власній логіці системи, оскільки готовий inference API не охоплює аспектів збереження статистики, інтерпретації патернів поведінки, обробки повторюваних виявлень або візуалізації детекцій у режимі реального часу. Тому сучасні рішення можуть бути використані лише частково — як компонент глибинного аналізу зображень — але не як завершена система для моніторингу мімічної активності.

Узагальнюючи, сучасні застосунки та API демонструють значні можливості в аналізі обличчя, проте їх функціональний профіль спрямований на косметичні трансформації, емоційний аналіз або загальні характеристики шкіри, а не на виявлення локальних структурних ознак перенапруження. Це створює об'єктивну потребу у розробці окремої системи, яка поєднує точність спеціалізованої моделі об'єктної детекції з можливістю статистичного моніторингу та інтерфейсного подання інформації користувачеві.

РОЗДІЛ 2 — ТЕОРЕТИЧНІ ЗАСАДИ ПРОЄКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ МІМІЧНИХ НАПРУЖЕНЬ

2.1 — Вимоги до системи моніторингу зморшок

На основі проведеного аналізу специфіки мімичного перенапруження, його фізіологічних механізмів та доступних технологічних підходів виникає необхідність визначити вимоги до майбутньої системи автоматичного моніторингу. Вимоги буде задокументовано у даному розділі у формі PRD (Product Requirements Document)

2.1.1 Короткий опис продукту і постановка проблеми

Десктоп аплікація яка нотифікує користувача про те що він нахмурився і нагадує розслабити лице. Також користувач може побачити статистику зморщень які ідентифікував додаток для того щоб бачити користь від використання додатку.

Проблема неконтрольованого мімичного перенапруження є поширеним явищем серед користувачів комп'ютерів, офісних працівників, студентів та людей, що тривалий час працюють у стані підвищеної концентрації. Значна частина міміки, яка супроводжує інтелектуальну або емоційну діяльність, виникає автоматично та не усвідомлюється людиною. Зокрема, мимовільне напруження лобного та міжбрівного м'язів, відоме як frowning, є одним із найбільш типових прикладів. Навіть короткочасне, але часте повторення такого м'язового скорочення створює підвищене навантаження, що з часом може спричиняти головні болі напруги (Tension-Type Headache), локальну втомлюваність м'язів, погіршення емоційного стану та формування стійких мімичних заломів.

Попри поширеність явища, більшість людей не здатні контролювати ці мікроповедінкові патерни без зовнішньої допомоги. Мімичні реакції формуються у моторній корі та лімбічній системі, часто не проходячи крізь механізми свідомого контролю. У результаті людина може протягом багатьох годин

перебувати у стані непомітного перенапруження м'язів, що ускладнює своєчасне коригування і призводить до кумулятивних ефектів.

Існуючі програмні рішення не надають ефективного інструменту для автоматичного моніторингу таких проявів. Естетичні та косметологічні додатки концентруються на візуальних ефектах або на аналізі стану шкіри без прив'язки до мимічної активності. Емоційні аналітичні системи визначають Action Units, але не розпізнають локальні текстурні зміни, які є наслідком мимічного перенапруження. Готові API для аналізу зображень не підтримують роботу зі зморшками як з окремими семантичними об'єктами, а більшість комерційних систем не дозволяє кастомізацію моделей під конкретну задачу.

Таким чином, користувачі не мають засобів, які б дозволяли їм об'єктивно відстежувати власну мимічну активність під час роботи, розуміти частоту появи перенапруження та отримувати своєчасний зворотний сигнал. Відсутність таких інструментів обмежує можливість формування здорових поведінкових звичок та профілактики м'язового дискомфорту і появи мимічних ліній.

Вказане формує чітку потребу у створенні спеціалізованої системи, яка здатна автоматично виявляти мимічні заломы у реальному часі, локалізувати їх у межах обличчя, накопичувати статистику повторюваності та повідомляти користувача про появу небажаних патернів. Саме така система здатна компенсувати недоліки наявних рішень і забезпечити ефективний інструмент моніторингу мимічного перенапруження в умовах повсякденного використання.

2.1.3 Цілі та не-цілі

Визначення цілей і не-цілей дозволяє окреслити межі відповідальності системи та забезпечує коректне розуміння її функціонального призначення. Така структуризація спрощує подальше проєктування архітектури, визначення вимог і вибір технологій, формуючи чітке бачення того, які задачі система повинна вирішувати, а які залишаються поза сферою її впливу.

Мета розроблюваної системи полягає у створенні інструменту, який забезпечує автоматичний моніторинг мимічного перенапруження та допомагає

користувачу усвідомлювати власні мімічні реакції під час роботи за комп'ютером. Система має виявляти появу локальних мімічних заломів, що пов'язані з активністю лобного та міжбрівного м'язів, інтерпретувати ці ознаки як потенційний маркер напруження і формувати зворотний сигнал у вигляді статистики або нотифікацій. Таким чином, ціль продукту полягає у своєчасному виявленні повторюваних мікропатернів, які користувач зазвичай не помічає, і створенні передумов для формування більш здорової мімічної поведінки.

Водночас розроблювана система свідомо не орієнтована на функції, характерні для медичних діагностичних або косметологічних інструментів. Її призначення не включає постановку клінічних діагнозів, оцінку дерматологічних станів, визначення віку шкіри чи надання лікувальних рекомендацій. Система також не аналізує емоційний стан користувача та не класифікує емоційні реакції, оскільки ці завдання належать до іншої області комп'ютерного зору та нейронауки. Не ставиться за мету здійснювати біометричну ідентифікацію або зберігати зображення у довгостроковому вигляді, оскільки це суперечить принципам приватності, важливим у контексті схожих рішень.

До не-цілей також належить виконання високочастотного відстеження рухів обличчя, характерного для аналізу micro-expressions або Action Units. Завдання системи — фіксація текстурних проявів напруження на статичних зображеннях, а не аналіз динаміки м'язових скорочень. Крім того, система не має на меті повністю усунути мімічні реакції або змінити природну міміку користувача, оскільки її призначення — допомога у контролі перенапруження, а не регуляція емоційної поведінки.

2.1.4 User Stories / Сценарії використання

Проектована система передбачає взаємодію з користувачем у декількох типових сценаріях, що відображають реальні потреби людей, які мають схильність до несвідомого мімічного перенапруження під час роботи. Основа таких сценаріїв полягає у тому, що користувач прагне отримувати об'єктивні дані про власну міміку, але не бажає витратити час на налаштування чи ручний аналіз.

Тому перший і найважливіший сценарій полягає у можливості одним натисканням активувати процес моніторингу. Після натискання кнопки запуску система переходить у фоновий режим роботи, а користувач може продовжувати свої щоденні завдання, не відволікаючись від робочих процесів.

У ході звичайної роботи за комп'ютером користувач очікує, що система надасть зрозумілий та ненав'язливий сигнал у разі повторюваного мімічного перенапруження. Такий сценарій описує ситуацію, у якій людина, зосереджена на завданні, мимоволі напружує м'язи лоба або насуплюється, не усвідомлюючи цього. У відповідь система фіксує часті появи характерного мімічного залому і надсилає коротку нотифікацію, яка не перериває робочий процес, але дозволяє скоригувати поведінку. Таким чином, користувач отримує можливість у режимі реального часу впливати на власну міміку, що є важливим у профілактиці перенапруження.

Інший сценарій передбачає ознайомлення зі статистикою мімічної активності. Користувач відкриває інтерфейс програми, щоб переглянути кількість подій за день, тиждень або місяць, а також простежити їхній розподіл у часі. На цьому етапі він може оцінити, у які часові проміжки або у зв'язку з якими видами діяльності виникають найчастіші епізоди перенапруження. Такий сценарій важливий для формування усвідомлених мімічних звичок, а також для оцінки впливу стресу, робочого навантаження чи зміни способу життя на стан м'язів обличчя.

Список User stories з пріоритетами:

MUST HAVE — критично необхідні функції

1. Як користувач, я хочу запустити детекцію одним натисканням, щоб система почала моніторинг без складних налаштувань.
2. Як користувач, я хочу мати можливість зупинити детекцію, щоб повністю контролювати роботу програми.
3. Як користувач, я хочу бачити індикатор активності, щоб розуміти, чи система зараз аналізує мою міміку.
4. Як користувач, я хочу отримувати нотифікації, щоб система нагадувала мені розслабити лоб або міжбрів'я.

5. Як користувач, я хочу, щоб система зберігала статистику локально, щоб мої дані залишалися приватними.
6. Як користувач, я хочу, щоб система працювала у фоновому режимі, щоб я могла продовжувати працювати на комп'ютері без взаємодії з додатком.
SHOULD HAVE — важливі, але не критичні функції
7. Як користувач, я хочу мати доступ до статистики за день, щоб розуміти, скільки разів я напружував м'язи.
8. Як користувач, я хочу переглядати статистику за тиждень і місяць, щоб бачити динаміку своєї мімічної поведінки.
9. Як користувач, я хочу, щоб інтерфейс був простим та зрозумілим, щоб легко взаємодіяти з системою.
COULD HAVE — бажані, але другорядні функції
10. Як користувач, я хочу могли налаштувати чутливість детекції для різних типів зморшок, щоб могли виключити моніторинг тих зморшок які мене не цікавлять або є постійними на моєму лиці не залежно від міміки
11. Як користувач я хочу могли кастомізувати текст нотифікації для того щоб краще мотивувати себе розслабити лице.

1.4.5 Нефункціональні вимоги

Ефективність функціонування системи моніторингу мімічного перенапруження визначається не лише набором її функціональних можливостей, але й сукупністю нефункціональних характеристик, які забезпечують стабільність, продуктивність, безпеку та зручність використання. Оскільки система працює у фоновому режимі і взаємодіє з користувачем протягом тривалих робочих сесій, саме нефункціональні вимоги визначають реальну якість взаємодії та здатність продукту відповідати потребам щоденного застосування. Наявність нефункціональних вимог є невід'ємною складовою процесу проектування програмного забезпечення, оскільки саме вони визначають якісні характеристики системи, такі як надійність та керованість. У наукових працях з управління та розвитку програмних систем підкреслюється,

що ігнорування цих вимог на етапі дизайну призводить до зниження ефективності та ускладнення подальшої експлуатації програмного продукту [27].

Передусім особливу увагу слід приділити вимогам до продуктивності. Система має забезпечувати обробку зображень у режимі, близькому до реального часу, із затримкою, яка не перевищує кількох сотень мілісекунд від моменту захоплення кадру до отримання результатів інференсу. Це необхідно для своєчасної реакції на мимічні зміни та забезпечення точності фіксації повторюваних заломів. Крім того, важливо, щоб система не створювала надмірного навантаження на центральний процесор та інші ресурси комп'ютера, оскільки вона має працювати паралельно з основними задачами користувача, зокрема офісними, графічними або дослідницькими програмами.

Стабільність і надійність роботи є ще одним важливим нефункціональним аспектом. Система повинна коректно функціонувати впродовж тривалих періодів часу, не допускаючи збоїв, зависань і витоків пам'яті. Особливу складність становлять варіативні умови освітлення, зміна положення голови або наявність шумів у зображенні веб камери. Тому алгоритми мають бути стійкими до подібних факторів і зберігати стабільний рівень точності навіть за субоптимальних умов. Також необхідно гарантувати коректну роботу при взаємодії з іншими додатками, які можуть використовувати веб камеру, зберігаючи при цьому передбачуваність поведінки системи.

Зручність використання (UX) формує ще одну важливу групу нефункціональних вимог. Оскільки цільова аудиторія включає користувачів без технічних навичок, інтерфейс має бути інтуїтивно зрозумілим і мінімалістичним. Важливо забезпечити чітку індикацію того, чи система наразі здійснює аналіз, а також доступ до статистики у простому та наочному форматі. Нотифікації повинні бути ненав'язливими, з'являтися у відповідний момент та не перекривати активні робочі вікна. Процедура встановлення програми має бути простою, без необхідності встановлення сторонніх бібліотек чи налаштування середовищ, що забезпечує комфортний старт користувача.

Окремою категорією нефункціональних вимог є безпека та конфіденційність даних. Система має працювати виключно з локальною

інформацією, забезпечуючи зберігання статистики у локальній базі даних без передачі зображень на сторонні сервери. Це критично важливо в контексті чутливості обличчя як біометричної інформації та відповідності сучасним етичним стандартам. Користувач повинен мати можливість у будь-який момент переглянути, очистити або повністю видалити всі збережені дані.

Портативність та сумісність також відіграють важливу роль. Система повинна коректно працювати на різних конфігураціях обладнання, переважно в межах операційної системи Windows, що відповідає типовим умовам використання. Додатково важливо забезпечити автономність, тобто можливість роботи без постійного доступу до інтернету, що особливо актуально у випадках, коли модель детекції працює локально або коли користувач бажає зберегти максимальну приватність.

Узагальнюючи, нефункціональні вимоги до розроблюваної системи формують комплекс характеристик, які визначають її надійність, зручність, безпечність та придатність до щоденного використання. Виконання цих вимог є обов'язковою умовою для створення продукту, здатного забезпечити стабільний моніторинг мімічної активності в реальних робочих сценаріях.

2.1.6 Вимоги до інтерфейсу

Інтерфейс системи має бути побудований таким чином, щоб забезпечувати інтуїтивну, швидку та комфортну взаємодію користувача з основними функціями продукту. Оскільки система належить до класу фонових інструментів поведінкового моніторингу, інтерфейс повинен мінімально відволікати користувача від основної діяльності, залишаючись при цьому достатньо інформативним. Дизайн застосунку передбачає чітку структурованість, поділ на логічні блоки та використання візуальних елементів, які полегшують сприйняття інформації.

На головному екрані має розміщуватися блок керування процесом детекції, який відображає актуальний стан системи (Active/Stopped) та містить великі, чітко позначені кнопки для запуску та зупинки аналізу. Важливо, щоб ці елементи

були візуально контрастними та легко розпізнаваними, оскільки вони є основними точками взаємодії користувача з програмою. Статус детекції має оновлюватися в реальному часі, забезпечуючи користувача впевненістю в тому, що система функціонує коректно.

Особливе значення має розділ статистики, у якому користувач може переглядати динаміку мімічних напружень за останні дні та тижні. Інтерфейс повинен включати візуалізацію у вигляді графіка “Last 7 Days”, що дозволяє інтуїтивно оцінити частоту подій. Графік має бути достатньо простим, з м’якою кольоровою палітрою, лаконічними підписами осей та акуратною інтеграцією числових значень. Додатково передбачено блок з агрегованою статистикою “Today / Yesterday / This Week / Last Week”, оформлений у вигляді окремих карток. Таке рішення полегшує порівняння та прискорює отримання ключових показників.

У системі передбачено також окремий блок налаштувань, який дозволяє користувачу адаптувати поведінку програми під власні потреби. Важливо, щоб цей блок був відокремлений візуально, але водночас зберігав стилістичну цілісність інтерфейсу. Користувач має змогу редагувати текст нотифікацій у спеціальному полі, яке повинно мати чіткий контур, легку тінь та достатню висоту для підтримки довших фраз. Додатково в цьому блоці передбачено елементи управління чутливістю детекції (sensitivity sliders), оформлені у вигляді горизонтальних повзунків із плавними градієнтними кольорами. Кожен повзунок супроводжується числовим позначенням відсотку, що забезпечує точність налаштування та зрозумілість зворотного зв’язку.

Стильова палітра інтерфейсу має відповідати загальним принципам м’якого, ненав’язливого UI: використання пастельних тонів, плавних тіней, округлених форм та градієнтів підсилює відчуття легкості й водночас підкреслює сучасний характер продукту. Важливо, щоб усі кольори відповідали критеріям контрастності, що забезпечує доступність для користувачів із різними візуальними особливостями.

Інтерфейс також повинен забезпечувати наочність у відображенні станів. Елементи “Start” і “Stop” мають чітко змінювати свій стиль залежно від

активності системи, а статус “Active/Stopped” повинен оновлюватися миттєво при зміні режиму. Додатково передбачено, що користувачу має бути зрозуміло, чи система використовує камеру в даний момент, аби уникнути конфліктів із іншими програмами.

Таким чином, вимоги до інтерфейсу включають забезпечення прозорості, простоти та естетичної цілісності, що дозволяє користувачу легко керувати системою, отримувати ключову інформацію та налаштовувати поведінку програми відповідно до власних потреб. Усі елементи інтерфейсу мають залишатися функціонально зрозумілими та візуально узгодженими, що сприяє позитивному користувацькому досвіду й підсилює практичну цінність продукту.

2.1.7. Ризики та обмеження, а також стратегії їх мінімізації

Під час розробки системи моніторингу мімічного перенапруження необхідно враховувати ризики, що виникають як унаслідок технічних обмежень програмної частини, так і через неконтрольовані умови середовища. Частина з них може бути суттєво зменшена за рахунок відповідних архітектурних рішень або корекції алгоритмів тоді як інші залишаються прийнятними ризиками, що впливають із самої природи роботи з відеопотоком у реальному часі.

Одним із найбільш вагомих чинників ризику є неконтрольоване освітлення. Зміни яскравості, тіньові ділянки, сильні відблиски або нерівномірне освітлення можуть призводити до спотворення текстур обличчя та, відповідно, до некоректної детекції мімічних заломів. Оскільки контроль умов освітлення неможливий у реальному використанні, цей ризик є частково прийнятим. Однак система частково його мітигує завдяки використанню порогової чутливості, яка дозволяє відфільтровувати слабкі артефакти освітлення, а також за рахунок того, що нотифікації генеруються лише за повторюваних патернів, а не за одиничних спрацьовувань.

Другим обмеженням є якість веб камери. Багато пристроїв мають низьку роздільну здатність, автоматичну корекцію різкості чи шумозаглушення, що може маскувати дрібні мімічні лінії. Цей ризик є переважно прийнятим, оскільки

система має працювати з обладнанням, яке вже доступне користувачеві. Проте його наслідки мінімізуються через використання детекції у часовому розрізі: навіть якщо окремий кадр нечіткий, система орієнтується на повторюваність ознак і може відфільтровувати нестабільні результати.

Помилкові спрацьовування — ще один ризик, пов'язаний як із моделлю детекції, так і з особливостями освітлення та якості зображення. Це може проявлятися у випадках, коли тіні від волосся, окуляри чи інші об'єкти на зображенні сприймаються моделлю як зморшки. Цей ризик частково мітигується завдяки механізму порогової чутливості, який дозволяє адаптувати систему під конкретного користувача, а також завдяки тому, що нотифікації генеруються лише за умов повторюваності або високої впевненості моделі. Таким чином, випадкові поодинокі помилки не призводять до надмірної кількості попереджень.

Варіативність облич — природна особливість, що охоплює різні форми черепа, структуру м'язів, вікові особливості, наявність статичних зморшок і зміни пози голови. Цей ризик є фундаментально прийнятим і не може бути повністю усунутий у рамках системи, яка працює на одному загальному датасеті. Однак він частково зменшується через можливість налаштовувати чутливість детекції, що дозволяє користувачеві адаптувати систему під власні анатомічні особливості.

Окремо необхідно відзначити залежність від стабільного інтернет-з'єднання, оскільки модель детекції виконується через Roboflow API. Переривання зв'язку може призводити до пропусків кадрів або затримок при обробці. Цей ризик є обмежено прийнятим: система попереджає про проблеми в логах, а цикл детекції автоматично продовжує роботу при відновленні з'єднання. Проте повністю усунути залежність від зовнішнього ресурсу можливо лише шляхом майбутнього переходу на повністю локальний інференс.

Ще одним технічним ризиком є потенційне збільшення навантаження на процесор під час тривалої роботи, особливо на слабких ноутбуках. Це обмеження частково мітигується використанням фонових потоків, оптимізацією циклу обробки, а також адаптивним інтервалом захоплення кадрів, який не перевищує можливостей системи.

Додатковим ризиком при проєктуванні та експлуатації системи може бути невідповідність її якісних характеристик очікуваним вимогам, зокрема щодо продуктивності, стабільності та надійності роботи. Застосування стандартизованого підходу до оцінювання якості програмного забезпечення на основі формалізованих метрик дозволяє зменшити ризики проєктування шляхом об'єктивного аналізу та порівняння альтернативних рішень на ранніх етапах розробки. Ефективність такого підходу підтверджується дослідженнями, у яких запропоновано технології оцінювання якості та продуктивності програмних систем із використанням міжнародних стандартів якості [28].

Таким чином, ризики та обмеження системи можна розділити на ті, що є прийнятними та впливають з умов реального застосування (варіативність облич, якість веб камери, зовнішні умови освітлення), і ті, що частково компенсуються архітектурними та алгоритмічними рішеннями (фільтрація помилкових спрацьовувань, механізми чутливості, обробка повторюваності патернів, багатопотокова обробка). Такий аналіз дозволяє правильно оцінити очікувану поведінку системи та формує підґрунтя для подальшого вдосконалення її робочих механізмів.

2.2 Методи та моделі, використані в системі

За результатами аналітичного огляду, проведеного у попередньому розділі, для реалізації системи було обрано низку технологічних рішень, які найповніше відповідають специфіці задачі моніторингу мімічного перенапруження. У цьому підрозділі розглядається теоретична основа цих методів, а також наводиться обґрунтування вибору відповідних моделей і технологічних інструментів. Метою є демонстрація того, яким чином застосовані алгоритми, архітектури та платформи забезпечують необхідний рівень точності, стабільності та продуктивності системи, а також чому саме ці рішення є оптимальними для поставленої проблеми.

2.3 Об'єктна детекція як підхід

Об'єктна детекція є одним із ключових напрямів сучасного комп'ютерного зору, що поєднує класифікацію та локалізацію об'єктів на зображенні. На відміну від методів, спрямованих на аналіз глобальних характеристик кадру, підхід детекції дає можливість ідентифікувати окремі структури разом із точними координатами їх розташування. У контексті задачі моніторингу мімічного перенапруження це дозволяє розглядати мімічні лінії та локальні заломы як окремі об'єкти, присутність і впевненість яких можуть бути формалізовані у вигляді структурованих даних.

Основна перевага об'єктної детекції полягає у здатності виділяти малі та слабо виражені елементи зображення, що робить цей підхід особливо доречним для задачі виявлення мімічних патернів. На відміну від традиційних алгоритмів виявлення контурів або текстур, детекційна модель не просто фіксує наявність лінійних структур, а інтерпретує їх у семантичному контексті, визначаючи, чи відповідають ці структури певному класу зморшок або мімічних заломів. Це забезпечує значно вищу специфічність і дозволяє відокремити реальні прояви м'язового напруження від сторонніх артефактів, таких як тіні чи відблиски.

З технічної точки зору об'єктна детекція відрізняється високою гнучкістю у застосуванні. Вона дозволяє адаптувати модель під конкретні типи об'єктів шляхом відповідної підготовки датасету та анотацій. Саме ця властивість лягла в основу обраного рішення: модель, використана у системі, була навчена на даних, що включають декілька типів зморшок та ліній, характерних для перенапруження мімічних м'язів. Отже, алгоритм може аналізувати не абстрактне зображення обличчя, а конкретні мімічні зони, що потребують контролю.

У практичному аспекті використання об'єктної детекції дозволяє не тільки визначати момент виникнення напруження, але й переводити ці дані в числовий формат для подальшого статистичного аналізу. Система отримує для кожної детекції набір параметрів — координати прямокутника, клас та рівень впевненості — що дозволяє ефективно агрегувати інформацію, відстежувати повторювані патерни і формувати аналітику у вигляді щоденних та тижневих показників.

Таким чином, об'єктна детекція є методологічним ядром системи, оскільки забезпечує одночасно точність, узагальненість і структурованість даних, необхідних для розуміння поведінки мімічних м'язів у динаміці. Це робить підхід оптимальним для реалізації практичного завдання моніторингу мімічного перенапруження та формування корисних для користувача рекомендацій.

Сучасні підходи до об'єктної детекції базуються на використанні глибинних нейронних мереж, що дозволяють поєднувати класифікацію та локалізацію об'єктів у єдиній моделі. Існує два основні класи детекторів: одноетапні (one-stage) та двоетапні (two-stage). У контексті системи моніторингу мімічного напруження важливо розуміти їхні принципові відмінності, оскільки саме вони визначають баланс між точністю, швидкістю та здатністю моделі працювати з дрібними об'єктами, до яких належать мімічні лінії.

Одноетапні моделі, такі як YOLO (You Only Look Once) та SSD (Single Shot Detector), виконують прогноз класу й координат об'єкта за один прохід через мережу. Це забезпечує високу швидкість інференсу та робить їх придатними для систем, що працюють у режимі, близькому до реального часу. Архітектура YOLO використовує поділ зображення на сітку й одночасне передбачення декількох bounding-box-кандидатів для кожної комірки, що дає змогу ефективно виявляти навіть невеликі об'єкти [29]. У сучасних версіях YOLO точність розпізнавання тонких структур зросла завдяки застосуванню багаторівневих ознак, механізмів уваги і вдосконалених згорткових блоків (convolutional blocks).

SSD, при збереженні одноетапного принципу, пропонує альтернативну архітектуру, у якій детекція виконується на кількох рівнях фіч-карт. Це дозволяє моделі краще працювати з об'єктами різного масштабу, включно з дрібними текстурними елементами, такими як зморшки чи тонкі мімічні заломы [30]. Проте SSD зазвичай демонструє нижчу точність у порівнянні з новішими представниками сімейства YOLO.

Двоетапні моделі, зокрема Faster R-CNN, виконують процес детекції у два кроки: спочатку генеруються регіони-кандидати (Region Proposal Network), а потім кожен кандидат класифікується й уточнюються його межі. Подібний підхід забезпечує високу точність, особливо для складних або слабконтрастних

об'єктів [31]. Проте ці моделі є обчислювально складнішими й мають меншу швидкість інференсу, що робить їх менш придатними для фонові роботи в реальному часі на звичайних користувацьких пристроях.

Для задачі виявлення мімічних заломів важливим є поєднання високої швидкості обробки кадрів та здатності моделі працювати із дрібними об'єктами, що можуть мати низький контраст або бути частково затемненими. Саме одноетапні моделі, зокрема сучасні модифікації YOLO, найкраще відповідають цим вимогам, оскільки забезпечують баланс між продуктивністю та точністю, необхідними для моніторингу в умовах повсякденної експлуатації. Багато готових моделей, доступних через спеціалізовані платформи, також будуються саме на базі цих архітектур, що спрощує інтеграцію та забезпечує стандартизований формат результатів.

Таким чином, архітектури YOLO, SSD та Faster R-CNN репрезентують теоретичну основу, на якій базується сучасна детекція об'єктів. Обраний у системі підхід спирається на переваги одноетапних моделей, що дозволяє досягти високої швидкості інференсу при достатній точності, необхідній для аналізу мімічних патернів у реальному часі.

Платформа Roboflow надає можливість отримувати результати детекції через стандартизований HTTP-запит, у якому зображення передається у закодованому вигляді (Base64). На серверній стороні запиту відбувається обробка кадру за допомогою моделі, яка навчена на спеціалізованому датасеті, що включає різні типи мімічних зморшок та інші локальні текстурні патерни. Наявність декількох класів дозволяє системі виконувати аналіз окремих мімічних зон, що підвищує інформативність зібраних даних [32].

Результати інференсу повертаються у форматі JSON і містять набори координат, клас об'єкта та рівень впевненості моделі. Такий формат є придатним для подальшого опрацювання і дозволяє системі зберігати інформацію у базі даних, аналізувати повторюваність патернів та формувати статистичні показники у часовій динаміці. Оскільки структура відповіді є уніфікованою, модуль інтерпретації результатів може працювати незалежно від конкретної конфігурації

моделі, що підвищує гнучкість архітектури та спрощує можливе майбутнє оновлення моделі або зміни її версії.

Технічно важливою перевагою використання Roboflow є можливість виконання інференсу на стороні хмарної інфраструктури, що зменшує навантаження на локальний пристрій. Це особливо актуально для сценаріїв фонові роботи, коли застосунок повинен залишатися легким та непомітним для користувача, не знижуючи продуктивності його робочої системи. Крім того, така реалізація мінімізує вимоги до встановлення: кінцевий користувач не потребує додаткових бібліотек або середовищ виконання, що зазвичай необхідні для локального запуску нейронних мереж.

З точки зору інтеграції у загальну архітектуру програми, взаємодія з Roboflow є асинхронною та проводиться як окремий етап у циклі детекції. Після захоплення кадру вебкамерою вбудований модуль інференсу формує запит, отримує відповідь та оновлює відповідний запис у базі даних. Такий підхід забезпечує чітке розділення функціональних компонентів і дозволяє легше контролювати помилки, наприклад у випадку перевищення часу очікування або нестабільності мережевого з'єднання.

У підсумку використання Roboflow Inference API у поєднанні з моделлю wrinkle detection дає змогу забезпечити точну локалізацію та класифікацію мімічних заломів без необхідності складного розгортання ML-інфраструктури на стороні користувача. Це робить технологічне рішення оптимальним для системи, що орієнтована на простоту інсталяції, стабільність роботи та мінімальне ресурсне навантаження.

Вибір підходу об'єктної детекції у поєднанні з використанням попередньо навченої моделі wrinkle detection забезпечує низку технологічних переваг, які роблять це рішення оптимальним для задачі моніторингу мімічного перенапруження в умовах повсякденної експлуатації. Однією з ключових переваг є здатність моделі працювати з дрібномасштабними й слабовираженими структурами, характерними для мімічних ліній. Об'єктна детекція дозволяє виділити такі структури як окремі семантичні об'єкти, що значно спрощує подальший аналіз та формування статистики.

Важливою перевагою є і висока швидкість одноетапних архітектур, що лежать в основі обраної моделі. Це дає змогу запускати детекцію у фоновому режимі без відчутного навантаження на обчислювальні ресурси користувача. Для настільного застосунку, який має працювати непомітно і стабільно протягом тривалих проміжків часу, цей аспект є критично важливим.

Використання зовнішнього inference-сервісу через Roboflow API додатково спрощує технічну інфраструктуру програми. Такий підхід усуває потребу у локальному розгортанні моделі та її залежностей, що покращує переносимість системи та зменшує вимоги до апаратного забезпечення. Крім того, обчислювально складні операції виконуються на стороні хмарного сервісу, що підвищує загальну стабільність та точність роботи.

Не менш важливим є те, що модель повертає структуровані результати, придатні для формування кількісної аналітики. Кожен об'єкт характеризується координатами та рівнем впевненості, що дозволяє системі агрегувати детекції у часових інтервалах, будувати історичні тренди та визначати патерни мімічної активності. Це робить підхід не лише технічно ефективним, але й функціонально релевантним, оскільки система спрямована на поведінковий аналіз, а не лише на класифікацію окремих кадрів.

Таким чином, обраний метод поєднує швидкість, точність, простоту інтеграції та здатність створювати аналітичну цінність на основі результатів детекції. Сукупність цих характеристик підтверджує доцільність використання саме такого технологічного рішення у системі моніторингу мімічного перенапруження.

2.4. Теоретичне обґрунтування архітектурних рішень

Архітектура програмної системи є ключовим фактором, що визначає її функціональні можливості, стабільність роботи та зручність використання. Після визначення методів аналізу обличчя та вибору моделі детекції постає завдання побудувати таку інфраструктуру, яка дозволить забезпечити безперервний моніторинг, мінімальне навантаження на обчислювальні ресурси та високий

рівень приватності для користувача. У цьому підрозділі розглядаються теоретичні засади архітектурних рішень, обраних для реалізації системи: від концепції desktop-first до організації обчислювальних процесів, інтеграції локальних і хмарних ресурсів, вибору технологічного стеку та забезпечення конфіденційності даних. Метою є обґрунтувати, чому саме така архітектура є оптимальною для поставленої задачі та яким чином вона відповідає вимогам, визначеним у попередніх підрозділах.

Вибір між desktop-first та browser-first архітектурою є принципово важливим рішенням для систем, що взаємодіють із апаратними ресурсами користувача та мають працювати у фоновому режимі. У контексті поставленої задачі — безперервного моніторингу мімічного перенапруження на основі періодичного захоплення зображення з вебкамери — саме desktop-first підхід є оптимальним з огляду на обмеження вебсередовища, вимоги до продуктивності та потребу в забезпеченні приватності. [33]

Браузерні технології обмежують доступ до апаратних ресурсів як із міркувань безпеки, так і через архітектурні особливості. Робота з вебкамерою можлива лише за активної взаємодії користувача та за наявності відкритої вкладки браузера, що суперечить вимогам фоновій роботі, визначеним у функціональних вимогах системи. Крім того, браузер не забезпечує гарантовану стабільність довготривалих процесів: вкладки можуть бути призупинені системою, перезавантажені користувачем, закриті або переведені у стан сну для економії ресурсів. Це робить browser-first підхід непридатним для систем, які повинні працювати неперервно протягом багатьох годин.

У свою чергу, desktop-first підхід відкриває доступ до широкого набору системних можливостей: стабільної роботи з веб камерою через бібліотеки OpenCV, контролю за власним життєвим циклом процесу та можливості взаємодії з локальною файловою системою. Така архітектура дозволяє організувати окремий фоновий потік, в якому виконується цикл захоплення й обробки зображень, не залежачи від активності користувача чи від стану інтерфейсу. Це забезпечує передбачуваність інтервалів детекції та стабільність системи під час довготривалого використання [34].

Ще одним важливим аргументом на користь desktop-first є необхідність локального зберігання даних. Система працює з персональними зображеннями обличчя, що належать до чутливих даних, і тому вимога приватності є критичною. У браузерному середовищі дані зберігаються у нестабільних або обмежених сховищах (IndexedDB, LocalStorage), які не підходять для структурованих записів або великих обсягів. У desktop-реалізації дані можуть бути збережені у файлах та базі SQLite з повним контролем доступу та можливістю шифрування.

Desktop-first також дозволяє сформувати єдиний, цілісний користувацький досвід. Завдяки PyWebView, інтерфейс на основі веб технологій може бути упакований у нативне вікно, що усуває потребу в окремому браузері та робить взаємодію з системою більш природною для кінцевого користувача. Таким чином досягається поєднання гнучкості веб інтерфейсу та надійності десктопного застосунку.

У підсумку desktop-first підхід забезпечує стабільність фонових процесів, доступ до системних ресурсів, контроль над приватністю та можливість реалізації повноцінного життєвого циклу застосунку. Саме ці характеристики роблять його теоретично та практично обґрунтованим вибором для системи моніторингу мімічного перенапруження.

Проектування системи, що здійснює аналіз зображень у режимі, наближеному до реального часу, вимагає чіткого визначення меж між обробкою, виконуваною на пристрої користувача, та обробкою, перенесеною на хмарні інфраструктури. Це рішення впливає на продуктивність, стабільність, вимоги до апаратних ресурсів, приватність та масштабованість системи. У контексті задачі моніторингу мімічного перенапруження було обрано комбінований підхід: локальна частина відповідає за організацію циклу обробки зображень, а інференс моделі детекції виконується у хмарному середовищі Roboflow. Таке розмежування має чітке теоретичне обґрунтування.

Локальна обробка є доцільною для тих компонентів, які повинні працювати стабільно та з мінімальною затримкою. Захоплення зображення з веб камери, попередня обробка та збереження даних у базу виконуються на стороні клієнта,

що гарантує передбачувану затримку та можливість контрольованого запуску незалежно від зовнішніх умов. Оскільки ці операції не є обчислювально складними, їх виконання локально зменшує загальний час проходження кадру через систему й виключає непотрібні мережеві затримки.

Інференс глибинних моделей, навпаки, є ресурсоємним і потребує наявності потужного графічного або спеціалізованого обчислювального обладнання. Виконання нейронної мережі локально вимагало б наявності відповідних бібліотек, значного обсягу пам'яті та сучасного графічного процесора, що суперечить вимозі простоти інсталяції та універсальності застосунку. У разі використання Roboflow inference обчислення переносяться у хмарне середовище, де модель працює на оптимізованих серверних конфігураціях. Це забезпечує стабільну швидкість роботи й дозволяє гарантувати точність, незалежно від характеристик пристрою користувача.

Розмежування локальних і хмарних функцій також сприяє гнучкості системи. Оскільки взаємодія з моделлю здійснюється через API, заміна або оновлення версії моделі може бути виконана без модифікації клієнтської частини програми. Це відповідає принципам модульності та ізоляції залежностей, які є важливими у проєктуванні масштабованих систем. Водночас локальна частина залишається стабільною та передбачуваною, оскільки її функціональність не залежить від змін у моделі або серверній інфраструктурі.

Окремим аспектом такого розподілу є приватність даних. Хоча зображення передаються у хмару для інференсу, система не зберігає їх на стороні сервера, а передає виключно тимчасові копії для разової обробки. Водночас персональні записи, статистика та історичні дані зберігаються локально у базі SQLite, що мінімізує ризик доступу до особистої інформації з боку сторонніх сервісів.

Таким чином, обраний підхід до розмежування локальної та хмарної обробки дозволяє досягти балансу між продуктивністю, простотою інсталяції та забезпеченням приватності. Локально виконуються операції, що потребують низької затримки та стабільного доступу до апаратних ресурсів, тоді як інференс моделі, що є складною задачею для обчислення, перенесено на відповідну інфраструктуру, здатну забезпечити необхідну якість та швидкість обробки. Таке

рішення є оптимальним для системи, орієнтованої на тривалу роботу у фоновому режимі та широке коло користувачів.

Вибір технологічного стеку для серверної частини та інтерфейсного шару системи зумовлений вимогами до продуктивності, простоти розгортання, крос-платформенності та інтеграції з настільним середовищем. У розробленій системі в якості основних технологій було використано FastAPI як вебфреймворк для реалізації API, SQLite як локальну систему керування базами даних та PyWebView як засіб вбудовування вебінтерфейсу у нативне десктопне вікно.

FastAPI обрано як серверний фреймворк завдяки поєднанню високої продуктивності, асинхронної моделі обробки запитів та зручної декларативної роботи з маршрутами й моделями даних. Оскільки система потребує обробки відносно невеликої, але регулярної кількості запитів від інтерфейсу (отримання статистики, зміна налаштувань, запити на запуск/зупинку детекції), надлишкова складність важких фреймворків є недоцільною. FastAPI базується на ASGI-стеку та добре поєднується з легковаговим сервером Uvicorn, що дозволяє швидко обробляти запити при невеликому обчислювальному навантаженні. Крім того, типовий для FastAPI підхід із використанням типізованих моделей полегшує документування й тестування API, що важливо для підтримки прозорості архітектури [35].

SQLite було обрано як система керування базами даних через її простоту, автономність та відсутність потреби у додатковому серверному оточенні. Оскільки система орієнтована на одного користувача і не вимагає багатокористувацького доступу чи високонавантажених транзакцій, використання повноцінного серверного СУБД (таких як PostgreSQL чи MySQL) є надлишковим. SQLite зберігає дані у вигляді одного файлу в локальній файловій системі, що спрощує розгортання, резервне копіювання та перенесення системи. Такий підхід повністю узгоджується з вимогами до приватності, оскільки всі дані залишаються на пристрої користувача, а доступ до них контролюється самим застосунком [36].

Використання PyWebView обумовлено бажанням поєднати переваги веб технологій у побудові інтерфейсу з нативним користувацьким досвідом

настільної програми. Замість того, щоб запускати інтерфейс у браузері, PyWebView створює нативне вікно операційної системи та вбудовує в нього веб сторінку, яка відображає фронтенд, побудований на React. Це дозволяє сприймати програму як повноцінний десктопний застосунок, із власною іконкою, вікном та поведінкою, звичною для користувача. Водночас розробник зберігає можливість використовувати сучасні веб технології (HTML, CSS, JavaScript, React) для швидкого створення гнучкого і візуально привабливого інтерфейсу [37].

Спільне використання FastAPI, SQLite та PyWebView формує узгоджену архітектуру, у якій серверна логіка, зберігання даних та інтерфейс чітко розділені, але взаємодіють через стандартизовані механізми: HTTP-запити та локальний файловий доступ. Це спрощує підтримку й розвиток системи, дозволяє замінювати або оновлювати окремі компоненти (наприклад, змінювати структуру бази даних або оновлювати фронтенд) без потреби у радикальній перебудові всієї архітектури. Таким чином, обраний технологічний стек відповідає як функціональним, так і нефункціональним вимогам, визначеним на етапі проєктування.

2.5 Забезпечення приватності

Оскільки система працює з чутливими персональними даними — зображеннями обличчя користувача — питання забезпечення приватності є визначальним при побудові архітектури. Приватність у даному контексті включає контроль за обробкою зображень, обмеження зберігання інформації, прозорість доступу до даних та мінімізацію ризиків, пов'язаних із передаванням даних третім сторонам що регулюється, зокрема, GDPR [38].

Першим принципом, на якому базується архітектура системи, є локальність основних операцій. Усі кадри, отримані з веб камери, обробляються у локальному середовищі користувача: вони зберігаються у внутрішній директорії програми, недоступній стороннім додаткам, та використовуються виключно у межах детекційного циклу. Запис у базу SQLite містить лише числові показники

активності моделі, а не самі зображення, що мінімізує обсяг персональних даних, які зберігаються тривалий час. Це відповідає принципу «мінімізації даних», рекомендованому сучасними стандартами обробки персональної інформації.

Другим елементом забезпечення приватності є контрольована і обмежена взаємодія з хмарною інфраструктурою. Передавання зображень у Roboflow API відбувається виключно для виконання одноразового інференсу, без збереження даних на стороні сервісу та без передачі метаданих, що дають можливість ідентифікації користувача. Оскільки зображення надсилається у вигляді тимчасового буфера та не використовується повторно, ризик витоку персональних даних є мінімальним. Крім того, API-сервіс не виконує навчання на отриманих даних, що додатково виключає ризики неконтрольованого використання зображень.

Хоча локальне виконання інференсу глибинної моделі є теоретично привабливим варіантом з точки зору приватності, його застосування у даній системі виявилось недоцільним через низку технічних, обчислювальних та архітектурних обмежень. Розглянемо ключові причини такого рішення.

Перш за все, локальний інференс вимагає наявності відповідного середовища виконання та значних апаратних ресурсів. Сучасні моделі об'єктної детекції, навіть оптимізовані для мобільних або edge-пристроїв, потребують або GPU-прискорення, або використання високопродуктивних бібліотек (TensorRT, ONNX Runtime, PyTorch із CUDA), які є складними для інсталяції та несумісними з широким спектром користувацьких конфігурацій. Оскільки цільова аудиторія програми включає користувачів звичайних ноутбуків, часто оснащених базовими інтегрованими відеоядрами, гарантувати стабільну роботу моделі локально практично неможливо. Таким чином, локальний інференс суперечить вимозі універсальності та доступності системи для некваліфікованого користувача.

Другим критичним аспектом є складність розгортання. Локальне виконання нейронної мережі потребує встановлення ML-фреймворків, вагових файлів моделі, залежностей та специфічних бібліотек для обробки зображень. Це значно ускладнило б процес інсталяції застосунку, збільшило б його обсяг та

поставило б додаткові вимоги до операційної системи. У той час як застосунок позиціонується як легковаговий, автономний та простий у встановленні, використання локального інференсу фактично виключало б можливість розповсюдження його у вигляді готового .exe-паketу без складних інструкцій.

Третім чинником стало питання продуктивності. Детекція зморшок належить до категорії завдань розпізнавання дрібних об'єктів із низьким контрастом, що є складним навіть для потужних моделей. На слабших ПК локальне виконання призводило б до значних затримок між кадрами, що робило б неможливим роботу у режимі, наближеному до реального часу. У хмарному ж середовищі моделі запускаються на спеціалізованому обладнанні (CPU/GPU-кластери), що гарантує стабільну та передбачувану продуктивність, незалежно від можливостей пристрою користувача [39].

Важливо зазначити, що використання Roboflow інференс API дозволяє досягти необхідного компромісу: зображення передаються на сервер без їх збереження, а локально зберігається лише агрегована статистика. Таким чином, ризик витоку персональних даних мінімізується, а система водночас залишається легковаговою та стабільною. Це відповідає принципам «privacy-aware design», коли приватність забезпечується не лише абсолютною локальністю обробки, а й мінімізацією обсягу й часу існування чутливих даних.

У підсумку, локальний інференс хоч і забезпечує максимальну приватність, однак є практично невідповідним вимогам до стабільності, продуктивності та простоти розгортання. Використання хмарного інференсу у поєднанні з локальним зберіганням статистики дозволяє зберегти приватність на високому рівні та водночас гарантувати доступність системи для широкого кола користувачів.

Третім важливим аспектом є централізований контроль за зберіганням та доступом до даних у локальній файлової системі. Оскільки SQLite-файл і конфігураційні дані зберігаються у внутрішніх директоріях користувача, доступ до них можливий лише в межах операційної системи та за участі самого застосунку. Це усуває ризики небажаного доступу через браузер або сторонні вебсервіси, що є типовими проблемами для browser-first архітектур.

Нарешті, застосунок не передає статистику, історію детекцій або будь-які особисті дані в мережу чи сторонні сервіси, що забезпечує повну автономність системи. Уся аналітика та історичні дані формуються локально, а користувач має повний контроль над своїми даними, включно з можливістю їх очищення.

Таким чином, забезпечення приватності у системі реалізовано через сукупність архітектурних рішень: локальну обробку даних, мінімізацію їхнього зберігання, контрольовану взаємодію з хмарними сервісами та повну автономність роботи програми. Це дозволяє гарантувати, що персональна інформація користувача залишається захищеною та не виходить за межі його пристрою.

2.5. Архітектура системи

Розроблена система моніторингу мімічного перенапруження має багатoshарову архітектуру, у якій чітко розмежовано відповідальність між інтерфейсом користувача, серверною логікою, модулем обробки зображень, модулем інференсу моделі та рівнем зберігання даних. Така структура забезпечує масштабованість, прозорість взаємодії між компонентами та спрощує подальший розвиток системи.

На верхньому рівні архітектури розташований інтерфейс користувача, реалізований як веб застосунок на базі React, що завантажується в нативне вікно настільного застосунку за допомогою PyWebView. Інтерфейс містить елементи керування процесом детекції (кнопки запуску й зупинки, індикатор стану), блоки відображення статистики (щоденні та тижневі показники, короткострокові графіки) та модуль налаштувань, де користувач може змінювати текст нотифікацій та рівень чутливості. Вся взаємодія між інтерфейсом та серверною частиною здійснюється через HTTP-запити до локального API.

Середнім шаром виступає бекенд, реалізований на основі FastAPI. Він виконує роль координаційного центру системи: приймає запити від інтерфейсу, викликає відповідні функції модулів обробки зображень і статистики, повертає агреговані дані для відображення в UI. До типових операцій бекенду належать

запуск і зупинка циклу детекції, отримання поточного статусу системи, формування зведеної статистики за вибрані інтервали часу, а також читання і запис конфігураційних параметрів. Бекенд одночасно відповідає за подання зібраної статистики в структурованому форматі, зручному для фронтенда.

На наступному рівні розташований модуль обробки зображень, що працює у фоновому потоці. Після ініціалізації детекції з боку бекенду цей модуль періодично захоплює кадри з веб камери, зберігає їх у локальну файлову систему та ініціює процес інференсу. Він також виконує роль першого етапу фільтрації результатів: на основі відповідей від моделі визначає, чи слід вважати конкретний кадр проявом мімічного перенапруження, та за потреби формує запит на показ нотифікації. Цей модуль є критичним для дотримання часових обмежень, оскільки саме він визначає інтервал між кадрами та загальний ритм роботи системи.

Модуль інференсу, інтегрований із зовнішнім сервісом Roboflow, реалізує безпосередньо виклик моделі детекції зморшок. На вхід він отримує шлях до зображення та порогове значення впевненості, кодує файл, надсилає його до API та отримує відповідь у вигляді переліку виявлених об'єктів із координатами та конфіденсами. Після цього результати повертаються в модуль обробки зображень, де вони перетворюються на придатний до збереження формат. Важливо, що модуль інференсу ізольований від решти логіки, що дозволяє змінювати модель або сервіс без впливу на інші частини системи.

Нижнім рівнем архітектури є підсистема зберігання даних. Вона базується на локальній базі даних SQLite, у якій зберігаються записи про всі детекції: час захоплення кадру, значення впевненості для кожного класу зморшок, статус нотифікації та використаний поріг чутливості. Окремі допоміжні модулі формують агреговану статистику за різні часові інтервали (сьогодні, учора, поточний тиждень, попередній тиждень, останні сім днів) для подальшого відображення в інтерфейсі. Конфігураційні параметри (наприклад, текст нотифікації та рівні чутливості) зберігаються в окремому файлі конфігурації, що забезпечує їх незалежність від основної схеми бази даних.

Між усіма цими рівнями організовано чіткі потоки даних. Користувачка дія в інтерфейсі (наприклад, натискання кнопки запуску) перетворюється на HTTP-запит до бекенду, той ініціює запуск фонового потоку, який починає цикл захоплення й аналізу кадрів. Після кожної детекції результати зберігаються у базі даних, а бекенд, у свою чергу, на запит фронтенда формує агреговану відповідь для відображення статистики. Такий поділ дозволяє підтримувати незалежність між інтерфейсом і внутрішньою логікою, а також легко розширювати систему, додаючи нові функції на окремих рівнях (Рис 2.1)

Узагальнюючи, високорівнева архітектура системи побудована за принципами модульності, ізоляції відповідальності та використання легковагових, але достатньо потужних технологій. Це забезпечує як відповідність вимогам до продуктивності та приватності, так і можливість подальшої еволюції системи без радикальної перебудови її структури.

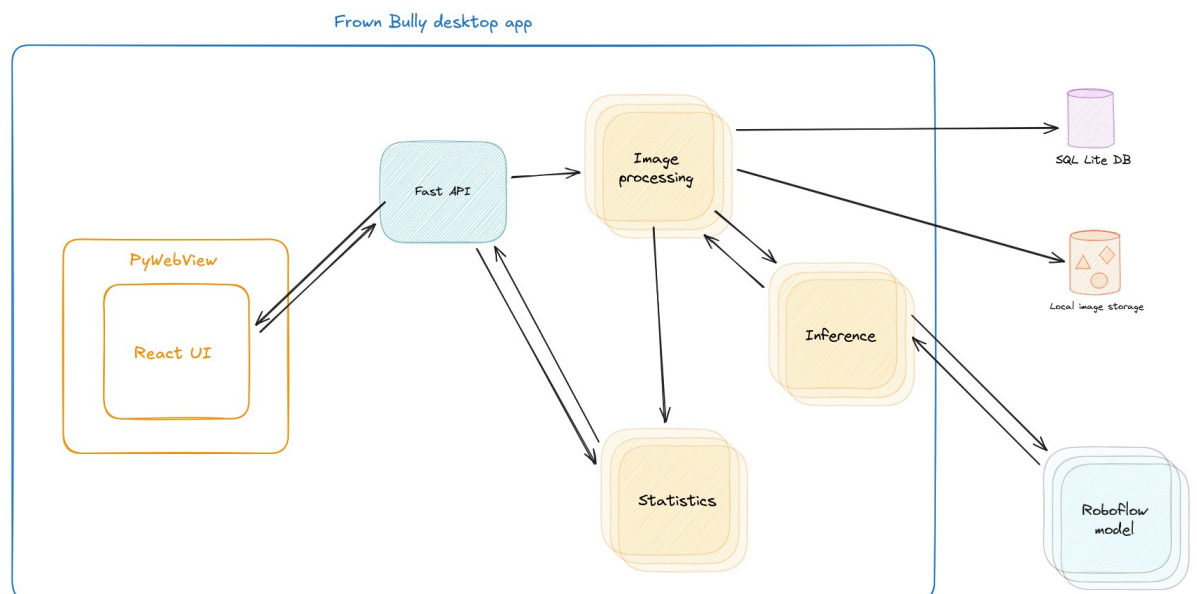


Рисунок 2.1 Високорівнева архітектура FrownBully

Взаємодія між компонентами системи побудована відповідно до принципів модульності, слабкого зв'язування та чіткого розподілу відповідальності. Така організація дає змогу забезпечити передбачувану роботу системи, спростити її масштабування та полегшити подальшу підтримку. Кожен компонент виконує одну логічно завершену функцію, а комунікація між ними здійснюється через стандартизовані інтерфейси, що мінімізує взаємні залежності.

Основним комунікаційним каналом між користувацьким інтерфейсом та внутрішньою логікою системи є HTTP-API, реалізоване у бекенд-компоненті FastAPI. React-інтерфейс не взаємодіє безпосередньо з жодним із внутрішніх модулів; він надсилає запити (наприклад, запуск детекції, отримання статистики, зміна налаштувань) до бекенду, який виступає єдиною точкою входу. Така організація відповідає принципу «API-gateway pattern», що дозволяє централізовано контролювати доступ, валідацію параметрів та обробку помилок.

Бекенд, у свою чергу, виконує роль координатора взаємодії між функціональними модулями. Він керує запуском і зупинкою фонових процесів, який реалізує періодичне захоплення зображень, а також формує запити до модулів статистики та зберігання даних. Комунікація між бекендом та модулем обробки зображень здійснюється через виклики локальних функцій та глобальні змінні стану, що дозволяє досягти низької затримки та мінімального накладного навантаження при роботі в реальному часі.

Модуль обробки зображень є посередником між локальним середовищем користувача та хмарним модулем інференсу. Він захоплює зображення з веб камери, зберігає їх у локальне файлове сховище та передає їх до модуля інференсу. Взаємодія між цими двома модулями є односторонньою та строго послідовною: спочатку створюється кадр, після чого він передається на аналіз, а отримані результати повертаються у вигляді структурованих даних, не залежних від формату оригінального зображення. Це відповідає принципу перетворення даних (data transformation), за яким складні або великі дані приводяться до компактного стану перед передачею далі системою.

Модуль інференсу комунікує зі стороннім сервісом Roboflow через REST-інтерфейс, надсилаючи зображення у вигляді Base64-кодування. Такий підхід забезпечує повну ізоляцію хмарного сервісу від решти системи: він отримує тільки одиничні запити без доступу до локального середовища або персональних даних користувача поза межами одного кадру. Після виконання інференсу модуль повертає результати у внутрішній формат, що забезпечує незалежність архітектури від конкретної реалізації або версії моделі, відповідно до принципу інкапсуляції.

Взаємодія з підсистемою зберігання даних реалізована через прямі запити до SQLite-бази. Кожен модуль, що генерує значущі дані (обробка зображень, статистика), працює з базою через прості SQL-операції, що забезпечує прозорість та повторюваність поведінки системи. Компоненти не обмінюються даними напряму, а використовують базу як централізоване джерело істини (single source of truth).

Загалом взаємодія між компонентами побудована за принципами поділу відповідальності, незалежності та використання стандартизованих протоколів обміну. Це дозволяє підтримувати чисту архітектуру, зменшує ймовірність помилок, полегшує тестування та забезпечує високу передбачуваність роботи системи як у короткочасних, так і у довготривалих сценаріях використання.

РОЗДІЛ 3 — ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ

Розроблена система FrownBully є настільним застосунком для моніторингу мімічних напружень у реальному часі, побудованим за гібридною моделлю, що поєднує локальний серверний бекенд та клієнтський веб-інтерфейс, інтегрований у нативне вікно. Обрана архітектура забезпечує, з одного боку, модульність і чітке розділення відповідальностей між компонентами, а з іншого — зручність використання у форматі desktop-first рішення без потреби у браузерному розгортанні. Система логічно складається з кількох взаємопов'язаних підсистем: бекенду (керування детекційним циклом, інференсом і даними), фронтенду (інтерфейс користувача та візуалізація статистики), підсистеми збирання/пакування та шару зберігання користувацьких даних. Архітектурні рішення програмної системи ґрунтуються на результатах попереднього моделювання, яке дозволяє визначити основні компоненти, їх взаємодію та обмеження системи. [40]

3.1 Архітектура та основні компоненти розробленої системи FrownBully

Структура проєкту організована навколо двох ключових директорій — backend/ та frontend/, які відповідають за серверну та клієнтську частини відповідно. Серверна частина реалізована мовою Python і виконує роль керуючого ядра системи: вона ініціалізує застосунок, забезпечує доступ до функцій керування детекцією через REST-інтерфейс, організовує цикл захоплення кадрів з вебкамери, виконує виклики інференсу моделі та формує агреговані статистичні показники. Клієнтська частина реалізована як односторінковий застосунок (SPA) на основі React і TypeScript та відповідає за взаємодію з користувачем: запуск/зупинку моніторингу, перегляд історії спрацьовувань, налаштування чутливості й тексту сповіщень, а також візуальне представлення результатів у вигляді графіків і карток.

Координаційним компонентом системи виступає модуль `main.py`, який реалізує точку входу для виконання застосунку. У межах цього модуля ініціалізується сервер FastAPI та налаштовується запуск через ASGI-сервер Uvicorn. Паралельно створюється нативне вікно настільного застосунку засобами PyWebView, у якому відображається зібраний фронтенд. Таким чином, `main.py` забезпечує інтеграцію веб-інтерфейсу з локальним сервером і формує єдину оболонку desktop-додатка, де UI та серверна логіка працюють як одна система.

Основний “робочий” цикл детекції реалізований у модулі `image_processor.py`, який відповідає за запуск і зупинку фонового процесу аналізу. На цьому рівні здійснюється періодичне захоплення кадрів із вебкамери за допомогою OpenCV, їх локальне збереження з прив’язкою до часових міток, а також подальше ініціювання інференсу. Отримані результати порівнюються з користувацькими порогами чутливості, після чого приймається рішення щодо показу системного сповіщення. Для відображення нотифікацій використовується бібліотека Plyer, що дозволяє інтегрувати сповіщення у стандартні механізми операційної системи. Окрім цього, модуль підтримує актуальний оперативний стан (наприклад, Running/Stopped, останній прогноз, останній кадр), який використовується як для внутрішньої логіки, так і для відповідей API на запити з інтерфейсу.

Логіка взаємодії з моделлю машинного навчання винесена в окремий модуль `inference.py`, що інкапсулює виклики Roboflow Inference API. Модуль приймає шлях до зображення та параметри довіри, виконує підготовку даних (зчитування і кодування зображення), формує HTTP-запит і обробляє відповідь у форматі JSON. Результатом є структурований перелік детекцій із показниками впевненості, які надалі можуть бути відфільтровані за порогом і подані у спрощеному вигляді для прийняття рішення в детекційному циклі. Такий підхід дозволяє ізолювати залежність від конкретного зовнішнього сервісу та зберігати незалежність решти модулів від деталей протоколу інференсу.

Підсистема аналітики реалізована в модулі `frown_stats.py`, який здійснює агрегацію накопичених даних та формує статистичні представлення, релевантні

для користувача. Модуль обчислює кількість спрацьовувань за ключові періоди (день, попередній день, тиждень, попередній тиждень), а також формує часові ряди для візуалізації історичної динаміки. Важливим аспектом є те, що в статистику включаються лише фактично показані сповіщення, тобто зафіксовані епізоди, які пройшли порогову перевірку та були інтерпретовані як значущі для користувача. Таким чином, система відокремлює “сирі” детекції від подій, що реально впливають на користувацький досвід та інтерпретацію результатів.

Для забезпечення узгодженої роботи з файловою системою використовується модуль `paths.py`, який централізує визначення розташування бази даних, журналів, конфігураційного файлу та директорій зі зображеннями. Це рішення є критичним для настільного застосунку, оскільки дозволяє враховувати відмінності між операційними системами та режимами запуску (розробка/продакшн), а також забезпечує відділення програмного коду від користувацьких даних. На практиці база даних, логи та налаштування зберігаються у спеціальному каталозі користувача (наприклад, у середовищі Windows — у `%LOCALAPPDATA%\FrownBully`), що спрощує оновлення застосунку та узгоджується з вимогами приватності й безпеки.

Окремий інфраструктурний шар забезпечує збирання та розповсюдження застосунку. Службові скрипти `setup.py` і `build.py` автоматизують підготовку середовища й процес пакування: зокрема, створення бази даних, генерацію ресурсів (наприклад, піктограми) та формування виконуваного файлу засобами `PyInstaller`. Конфігурація пакування описується у файлі специфікації (`FrownBully.spec`), а вихідні артефакти збірки накопичуються в директоріях `build/` та `dist/`, де `dist` містить кінцеві виконувані файли, придатні для розповсюдження.

Таким чином, `FrownBully` реалізовано як модульну систему з чітко визначеними функціональними ролями компонентів: бекенд забезпечує керування детекцією, інференсом і зберіганням подій; фронтенд відповідає за взаємодію з користувачем і візуалізацію; інфраструктурні модулі підтримують процес пакування та розгортання. Запропонована архітектура спрощує супровід і подальший розвиток системи, зокрема шляхом заміни джерела інференсу, розширення статистичних метрик або удосконалення механізму прийняття

рішень щодо сповіщень без необхідності радикальної перебудови всієї платформи.

3.2 Потік даних та взаємодія компонентів системи

Функціонування програмної системи FrownBully організоване як послідовність взаємодій між користувачем, клієнтським інтерфейсом, серверним застосунком і модулями оброблення даних. Потік даних у системі є детермінованим і побудований таким чином, щоб забезпечити узгодженість станів між фронтендом і бекендом, відтворюваність результатів детекції та можливість подальшої аналітики на основі накопичених подій. У загальному вигляді система реалізує цикл: ініціація моніторингу в інтерфейсі → запуск детекційного процесу на бекенді → захоплення кадрів і інференс → прийняття рішення щодо нотифікації → збереження події → агрегація статистики → відображення результатів у UI (Додаток 2).

На першому етапі користувач взаємодіє з інтерфейсом застосунку (React/TypeScript), задаючи параметри роботи системи, зокрема рівень чутливості, формат сповіщень та режим моніторингу. Після натискання команди запуску клієнтська частина формує HTTP-запит до локального REST API, який обслуговується FastAPI. Завдяки інтеграції з PyWebView фронтенд працює в межах настільного вікна, однак логіка взаємодії залишається ідентичною до класичної веб-архітектури: UI виступає джерелом керуючих команд, а бекенд — виконавчим середовищем.

Після отримання запиту серверний модуль `main.py` передає команду до підсистеми детекції, яка реалізована в `image_processor.py`. На цьому рівні ініціалізується фоновий потік, що забезпечує циклічне захоплення кадрів із вебкамери через OpenCV із заданим часовим інтервалом. Кожен отриманий кадр зберігається у локальну директорію подій із фіксацією часової мітки, що дозволяє однозначно співвіднести детекції, записи в базі та подальші візуалізації. Паралельно формується первинний запис про подію, який містить шлях до

зображення та службові метадані, необхідні для подальшого оновлення результатами інференсу.

Наступним етапом є виконання інференсу. Модуль `image_processor.py` викликає `inference.py`, передаючи шлях до збереженого кадру та порогове значення впевненості, що відповідає обраному користувачем рівню чутливості. У межах `inference.py` кадр зчитується з файлової системи, кодується та надсилається до Roboflow Inference API. У відповідь система отримує структурований результат детекції, який включає перелік знайдених об'єктів (класів зморшок/напружень), їх координати та числові значення `confidence`. Після цього результат приводиться до внутрішнього формату: виконується фільтрація за порогом довіри, відбираються значущі детекції, а також формується агрегований показник (наприклад, максимальна впевненість або наявність позитивної детекції як бінарна ознака), який далі використовується для прийняття рішення.

Рішення щодо сповіщення формується у модулі `image_processor.py` шляхом порівняння `confidence`-значень із користувацькими порогами чутливості та застосуванням додаткових умов (наприклад, обмеження частоти сповіщень або перевірка часу від останнього спрацювання — якщо така логіка увімкнена). У випадку, коли умови виконуються, система ініціює показ нотифікації через `Plyer` і фіксує факт спрацювання у базі даних. Якщо умови не виконуються, подія може бути збережена як “детекція без сповіщення” або відкинута — залежно від реалізованого режиму логування. Таким чином, розмежовуються “сирі” результати аналізу та ті події, які мають практичне значення для користувача і формують його досвід взаємодії з системою.

Накопичені події зберігаються у локальній базі даних `wrinkle_detection.db`, а також частково — у файловій системі у вигляді зображень. Важливо, що користувацькі дані (база, логи, конфігураційний файл, директорія з детекціями) розміщуються у спеціальному каталозі профілю користувача, що забезпечує відокремлення від програмного коду та полегшує оновлення застосунку без втрати історії. Користувацькі налаштування (пороги чутливості, тексти

сповіщень та інші параметри) зберігаються у конфігураційному файлі `config.json` і використовуються як в інтерфейсі, так і в бекенд-логіці прийняття рішень.

Для відображення аналітики клієнтська частина періодично або за запитом звертається до API-ендпоінтів, які повертають агреговані дані. Формування статистики виконується в модулі `frown_stats.py`: він виконує вибірки за часовими інтервалами, рахує кількість спрацьовувань за добу/тиждень/попередні періоди та генерує структури даних, придатні для побудови графіків. Отримані значення повертаються фронтенду у форматі JSON, після чого відображаються в інтерфейсі у вигляді числових показників і динамічних графіків.

Отже, потік даних у FrownBully охоплює повний цикл — від користувацької команди у UI до збереження подій і формування статистики — і базується на чіткому розмежуванні ролей компонентів. Така організація забезпечує прозорість роботи системи, контрольованість механізму сповіщень та можливість подальшого розширення: наприклад, введення більш складних правил тригерів, додаткових метрик якості детекції або альтернативних джерел інференсу без зміни загальної логіки взаємодії між модулями.

3.3. Обмеження системи та перспективи розвитку

Одним із ключових технічних обмежень системи виступає її залежність від хмарного інференсу, яка реалізується через використання сервісу Roboflow. Передача зображень на віддалені сервери зумовлює залежність від стабільності та якості інтернет-з'єднання: у випадку втрати мережі чи затримки відповіді хмарного API процес аналізу призупиняється або суттєво сповільнюється. Крім того, система прив'язана до встановлених лімітів API, доступності сервісу й особливостей політики оплати, що породжує довготривалі ризики для надійності та безперервності функціонування застосунку.

Поточна архітектура передбачає періодичне захоплення окремих зображень із фіксованим часовим інтервалом у діапазоні від 0,5 до 2 секунд. Такий підхід дозволяє оптимізувати навантаження на центральний процесор і мережеву інфраструктуру, проте суттєво обмежує можливості аналізу мімічної

активності та її динаміки. На відміну від потокового відеоаналізу, покадровий режим не дає змоги визначати тривалість і сталість мимічного перенапруження, відрізняти короточасні артефакти (зокрема моргання чи зміну ракурсу зйомки) від справжніх скорочень лицьових м'язів, а також відстежувати зміни у розташуванні ключових точок обличчя чи оцінювати м'язову активність з урахуванням попередніх кадрів. Втрата динамічної інформації зумовлює затримку між появою мимічної реакції та генерацією сповіщення, оскільки фіксація фраунів можлива лише після захоплення відповідного кадру, що затримує реакцію системи на одну чи кілька секунд. Час реагування, відповідно, залежить від встановленого інтервалу захоплення та не є миттєвим, що особливо критично у випадках, коли користувач швидко змінює міміку або вимагається підвищена чутливість і оперативність реагування. Відтак, дискретний режим, хоч і оптимізований для легких настільних рішень, органічно обмежує точність і швидкодію алгоритму.

Використання підходу об'єктного детектування з bounding-box також має низку методологічних обмежень. У межах запропонованої моделі можливим є лише виявлення наявності зморшок на кадрі без аналізу їхньої інтенсивності, протяжності чи морфологічних характеристик. На відміну від сегментаційних або тривимірних підходів, bounding-box не дозволяє розпізнавати дрібні локальні зміни текстури шкіри, не забезпечує динамічного відстеження розвитку зморшок у часі, є чутливим до геометрії обличчя та ракурсу зйомки й не враховує рельєфність поверхні шкіри. Внаслідок цього існує ризик інтерпретації природних ліній або освітлювальних артефактів як ознак мимічного перенапруження.

Точність і коректність детекції значною мірою залежать від характеристик апаратного забезпечення, зокрема веб камери. Недорогі моделі можуть генерувати шум на зображенні, спотворювати контури, мати некоректний баланс білого й подавати геометрично спотворене зображення обличчя. Освітлення також суттєво впливає на якість розпізнавання: наявність сильних тіней, бічного світла, відблисків на окулярах або надмірної яскравості суттєво ускладнює роботу моделі та призводить до підвищення кількості помилкових детекцій.

Оскільки на поточному етапі система не виконує попередньої нормалізації освітлення чи корекції контрасту, ці чинники залишаються суттєвим обмеженням її точності.

Існуюча реалізація системи базується на використанні статичних порогових значень чутливості, єдиних для всіх користувачів. Такий підхід не враховує індивідуальних анатомічних особливостей, природних статичних ліній обличчя, впливу косметичних процедур і різного базового м'язового тону. Для певної частини користувачів модель може виявитися надмірно чутливою, для інших — недостатньо чутливою. Відсутність механізмів адаптації порогових значень на основі користувацької поведінки чи історичних даних істотно знижує гнучкість системи.

Обмеження локального зберігання даних також відіграють значну роль у функціональності системи. Застосування бази даних SQLite є оптимальним для настільних застосунків з однопотоковим доступом, проте база зберігається виключно локально без можливості синхронізації між пристроями. Крім того, SQLite не оптимізована для високочастотних вставок при зростанні частоти кадрів, а її масштабованість є обмеженою, що потенційно ускладнює підтримку системи протягом тривалого періоду експлуатації й акумулювання статистичних даних за місяці чи роки безперервної роботи.

Функціонування системи Frown Bully супроводжується низкою ризиків, що можуть впливати як на якість роботи алгоритмів, так і на загальний користувацький досвід. Одним із ключових ризиків є значна варіативність умов, у яких здійснюється зйомка зображень. Оскільки система покладається на дані, що надходять від звичайних побутових вебкамер, її робота є чутливою до зовнішніх факторів: неконтрольованого освітлення, наявності тіней, відблисків, часткового перекриття обличчя або зміни кута нахилу голови. Усі ці явища здатні суттєво знижувати точність детекції та призводити до виникнення некоректних результатів, на які система реагує як на ознаки мімічного напруження.

Додатковим ризиком є можливість появи хибних спрацьовувань. Оскільки застосована модель об'єктної детекції оперує окремими кадрами й не аналізує мімічну динаміку, вона може трактувати природні анатомічні особливості — такі

як постійні лобні лінії, індивідуальна структура м'язів чи особливості шкіри — як ознаки перенапруження. Подібні ситуації призводять до надмірної кількості сповіщень, що знижує довіру до системи та може сприяти формуванню негативного користувацького досвіду. Особливо це помітно в умовах змінного або складного освітлення, коли тіні утворюють псевдовізуальні патерни, схожі на зморшки.

Ризик затримки між моментом появи мімічної реакції та відображенням сповіщення також є суттєвим. Оскільки система працює в дискретному режимі зйомки, між кадрами неминуче виникає часовий проміжок. Це означає, що нахмурення, яке виникло одразу після попереднього кадру, буде виявлено лише через певний час — у середньому від однієї до двох секунд залежно від налаштувань. Така затримка знижує ефективність системи як інструмента біофідбеку та може впливати на мотивованість користувача дотримуватися рекомендацій.

Окремим блоком ризиків виступає залежність роботи системи від зовнішнього хмарного сервісу інференсу. Будь-які технічні збої на стороні сервісу, зміни в його політиці використання, обмеження на кількість запитів або тимчасова недоступність мережі безпосередньо впливають на працездатність застосунку. У таких випадках система не здатна виконувати свої основні функції, незалежно від коректності локальної частини архітектури.

Певний рівень ризику пов'язаний і з питаннями приватності. Для виконання інференсу на сервер передаються зображення обличчя користувача, що хоча й не зберігаються в локальній репозиторії розробника, можуть викликати занепокоєння користувача та становлять потенційну проблему з точки зору нормативних вимог у деяких юрисдикціях. Доступ до веб камери завжди є критично чутливим аспектом, що вимагає високого рівня довіри до застосунку та його архітектури.

Додаткові виклики виникають у контексті тривалої роботи системи у фоновому режимі. Оскільки застосунок може функціонувати години поспіль, накопичення помилок у сторонніх бібліотеках, витоки пам'яті, проблеми з драйверами веб камери або збільшення обсягу бази даних з часом можуть

негативно впливати на стабільність роботи. У разі великих історичних даних може знижуватися продуктивність запитів до SQLite, що також впливає на швидкість оновлення статистики. Розглянемо імовірність і потенційний вплив на користувача якщо даний ризик справдиться в таблиці 3.1

Таблиця 3.1

Ризик	Ймовірність виникнення	Вплив на користувацький досвід
Неконтрольовані умови зйомки (освітлення, тіні, ракурс)	Висока	Високий

Продовження таблиці 3.1

Ризик	Ймовірність виникнення	Вплив на користувацький досвід
Хибні спрацьовування (false positives)	Середня	Високий
Затримка між мімічною реакцією та сповіщенням	Висока	Середній
Залежність від Roboflow API та інтернет-з'єднання	Висока	Високий
Занепокоєння щодо приватності (передавання зображень у хмару)	Середня	Середній–Високий
Тривала фонові робота та можливі витоки пам'яті	Середня	Середній
Деградація продуктивності через накопичення даних у SQLite	Середня	Середній
Низька якість вебкамери	Висока	Середній–Високий

Індивідуальні особливості користувача	анатомічні	Середня	Середній
--	------------	---------	----------

Сукупність цих ризиків визначає межі використання поточної версії системи та окреслює напрямки її подальшого вдосконалення, зокрема підвищення точності моделей, покращення алгоритмів фільтрації, оптимізацію архітектури та розширення можливостей для персоналізації користувацького досвіду.

3.3 Рекомендації для майбутніх версій

3.3.1 Покращення моделі та ML-складової

Перспективи подальшого розвитку системи Frown Bully значною мірою пов'язані з удосконаленням машинного навчання та розширенням можливостей моделі детекції. Одним із найважливіших напрямів еволюції є персоналізація моделі під конкретного користувача. Поточна версія системи використовує універсальну модель об'єктної детекції, навченої на узагальненому датасеті, що неминуче створює обмеження у випадках, коли індивідуальні анатомічні особливості або специфічні патерни міміки відрізняються від даних, використаних під час навчання. Натомість можливість дотренування моделі (fine-tuning) на приватних прикладах користувача дозволила б адаптувати алгоритм до конкретного обличчя, суттєво зменшити кількість хибних спрацьовувань та підвищити точність розпізнавання. Сучасні дослідження підтверджують ефективність персоналізованих моделей у задачах facial analysis: навіть невелика кількість зразків (5–20 зображень) здатна суттєво покращити точність на специфічних підзадачах, що робить transfer learning та few-shot learning перспективними підходами для систем цього типу [41]. Аналогічно до задач аналізу мережевого трафіку, де доведено істотний вплив способу виділення ознак на ефективність ML-моделей, перспективним напрямом є також подальше вдосконалення підходів до формування інформативних представлень даних [42].

Іншим важливим напрямом розвитку є розширення набору класифікованих мимічних ознак. Поточна модель зосереджена насамперед на виявленні зморшок верхньої частини обличчя — лобних ліній, міжбрів'я та інших областей, що є типовими індикаторами стресового чи концентраційного перенапруження. Однак наукові дослідження показують, що м'язове перенапруження в умовах стресу або тривалої роботи за комп'ютером може проявлятися також у піднятті плечей, напруженні шийних м'язів, стисканні щелеп або мікроміміці навколо рота [43].

Розширення моделі на такі ознаки створює можливість побудови більш комплексного інструмента оцінки м'язового навантаження, що виходить за межі локальної детекції зморшок. Це дозволило б перетворити Frown Bully на універсальний засіб моніторингу м'язово-емоційної поведінки користувача.

Паралельно з розширенням набору класів перспективним є перехід від об'єктної детекції до сегментаційних моделей. Методи семантичної та інстанс-сегментації, зокрема U-Net, Mask R-CNN або сегментаційні трансформери, забезпечують значно точніше визначення контурів зморшок і дозволяють оцінювати їхню форму, довжину та інтенсивність. Така детальна локалізація може бути використана для кількісної оцінки рівня напруження м'язів та зміни текстури шкіри з часом. Дослідження з області дерматологічного аналізу зображень підтверджують, що сегментаційні моделі мають суттєву перевагу в задачах оцінки стану шкіри та аналізу змін текстури, особливо у випадках слабких або тонких структур. Аналогічні підходи, засновані на сегментаційних моделях, демонструють підвищену точність аналізу складних контурів і текстур у сучасних інженерних задачах комп'ютерного зору, що підтверджує перспективність їх застосування і для аналізу зморшок [44].

Окремим вектором розвитку є впровадження моделей, що враховують часову динаміку. Поточний підхід базується на аналізі окремих статичних кадрів, що позбавляє систему можливості відстежувати тривалість мимічного скорочення або відрізняти короткі артефакти від справжніх патернів напруження. Використання відеоорієнтованих моделей — зокрема Temporal Convolutional Networks, LSTM-архітектур або сучасних відео-трансформерів — дає змогу аналізувати послідовності кадрів та визначати динамічні характеристики

мімічної активності. Це дозволило б не лише фіксувати факт перенапруження, але й оцінювати його тривалість, інтенсивність і навіть передбачати ймовірність появи у найближчий момент [45].

Питання інференсу також посідає важливе місце у контексті майбутнього розвитку системи. Використання хмарного інференсу, хоч і забезпечує гнучкість і простоту оновлення моделі, створює залежність від зовнішнього середовища. Перехід до локального інференсу за допомогою ONNX Runtime, TensorRT або OpenVINO дозволив би гарантувати приватність користувача, зменшити затримку між захопленням кадру та отриманням результату та забезпечити стабільність роботи незалежно від доступності мережі. Роботи в галузі оптимізації моделей для edge-inference демонструють, що компактні CNN і трансформери здатні працювати в реальному часі навіть на CPU середнього класу. Для системи Frown Bully це означило б можливість глибшого аналізу міміки при збереженні низьких вимог до апаратного забезпечення.

Нарешті, важливим напрямом розвитку є інтеграція механізмів автоматичного збирання та фільтрації даних для подальшого навчання. Система могла б у фоновому режимі накопичувати зображення, де модель працює з низькою впевненістю, після чого пропонувати користувачеві самотійно класифікувати їх. Подібний підхід, відомий як active learning, суттєво зменшує обсяг ручного маркування та дозволяє моделі адаптуватися до реального середовища використання. У поєднанні з персональним дотренуванням це могло б перетворити систему на самонавчальний інструмент, який з часом стає точнішим у межах конкретного робочого середовища користувача.

Таким чином, покращення моделі та машинного компонента системи включає широкий спектр інновацій: від персоналізації й розширення набору детектованих ознак до впровадження динамічних моделей та локального інференсу. Кожен із цих напрямів не лише підвищує точність і стабільність розпізнавання, але й відкриває можливість розширення Frown Bully у бік повноцінної системи оцінки м'язової активності, поведінкових патернів та емоційно-статусної динаміки користувача.

3.3.2 Інференс та архітектура: напрями розвитку

Майбутній розвиток системи Frown Bully безпосередньо пов'язаний з удосконаленням архітектури інференсу та переходом до більш автономних, оптимізованих рішень. Наразі система покладається на хмарний інференс через API Roboflow, що дозволяє працювати з високоточними моделями без необхідності локального розгортання нейронних мереж. Проте такий підхід створює низку обмежень, серед яких залежність від стабільного інтернет-з'єднання, затримка, пов'язана з передаванням даних, а також потенційні ризики для приватності. Саме тому одним із ключових напрямів подальшої еволюції є перехід до локального інференсу або до гібридної архітектури, яка поєднує локальні та хмарні обчислення.

Локальний інференс забезпечує значно більшу автономність системи. Завдяки прогресу в оптимізації нейронних мереж—зокрема методам компресії моделей, квантизації, прунінгу та дистиляції—навіть складні моделі можна виконувати на звичайних CPU або інтегрованих GPU. У деяких дослідженнях обґрунтовано, що застосування прунінгу та кодування ваг дозволяє зменшити обсяг моделі у 35–49 разів без суттєвої втрати точності, що робить такі моделі придатними для виконання на краєвих пристроях (edge devices) [46].

Подальші дослідження, демонструють, що сучасні оптимізовані CNN та трансформери в змозі працювати у реальному часі на мобільних або вбудованих пристроях без суттєвого зниження якості інференсу [47].

Для Frown Bully це відкриває можливість виконувати детекцію зморшок і мімічного перенапруження без звернення до зовнішніх API, що істотно зменшить затримку, підвищить стабільність роботи та повністю усуне ризики, пов'язані з передаванням зображень у хмару.

Одним із перспективних напрямів є також гібридний режим інференсу. У такій конфігурації базова модель виконується локально — швидко, автономно і з гарантованою приватністю, — тоді як розширені або складніші патерни мімічної активності можуть аналізуватися хмарною моделлю, коли це необхідно. Такий підхід уже активно використовується в сучасних системах комп'ютерного зору,

які комбінують edge-виконання та server-side inference для досягнення балансу між швидкістю та точністю. Практики edge–cloud розділення обчислень розглядаються в дослідженнях де підкреслюється, що 70–90% запитів у задачах розпізнавання об'єктів можуть оброблятися локально, тоді як хмарні обчислення використовуються для складних випадків [48].

На архітектурному рівні перехід до локального інференсу потребуватиме інтеграції середовищ виконання, таких як ONNX Runtime, OpenVINO або TensorRT, що дозволить запускати оптимізовану версію моделі без переписування додатка. ONNX Runtime, зокрема, дозволяє використовувати як CPU-only, так і GPU-оптимізовані бекенди, а також підтримує змішану прецизійність (FP16/INT8), що є критично важливим для реального часу. Подібні підходи широко застосовуються у комп'ютерному зорі для edge-пристроїв, про що свідчить, наприклад, робота Guo et al. (2022), яка демонструє здатність lightweight-архітектур забезпечувати >30 FPS на звичайних ноутбуках.

При цьому важливою складовою переходу до локальної архітектури є забезпечення конфіденційності даних користувача. Відсутність передавання зображень у мережу повністю нівелює ризики перехоплення або несанкціонованого доступу. Така практика відповідає сучасним вимогам до систем, які працюють з біометричними даними, серед яких GDPR та рекомендації Європейської комісії щодо використання систем розпізнавання обличчя у побутових застосунках. У більшості оглядів з цифрової етики підкреслюється, що локально виконувані моделі суттєво зменшують бар'єри довіри та сприяють прийняттю технології користувачами.

Попри переваги локального підходу, він також має свої обмеження. Оптимізовані lightweight-моделі зазвичай демонструють нижчу точність, ніж великі хмарні аналоги. З цієї причини створення гібридної архітектури або періодичне оновлення локальної моделі через централізований сервер може бути оптимальним компромісом. Іншим напрямом є можливість завантаження оновлених моделей «на вимогу», що дозволяє підтримувати систему у актуальному стані без залежності від великого бандлінгу під час встановлення.

У перспективі система Frown Bully може бути доповнена механізмами адаптивного інференсу: зміною частоти детекції залежно від активності користувача, вибором моделі різної складності залежно від умов освітлення, а також інтеграцією метрик якості кадру перед виконанням інференсу. Це відповідає сучасній тенденції до створення адаптивних, енергоефективних ML-систем, про що свідчать новітні роботи з архітектурної оптимізації та контекстно-обумовленого виконання нейронних моделей.

Таким чином, розвиток архітектури інференсу для Frown Bully охоплює широкий спектр технологічних можливостей — від переходу до повністю автономної локальної моделі до побудови гібридної інтелектуальної системи, що динамічно розподіляє обчислення між edge та cloud. Це не лише підвищить точність і швидкодію, а й дозволить системі відповідати найсучаснішим вимогам конфіденційності та надійності.

Висновки до розділу 3

У третьому розділі було представлено практичну реалізацію програмної системи Frown Bully, спрямованої на моніторинг мімічного перенапруження та автоматичне сповіщення користувача про появу стресових мімічних патернів. Було детально розглянуто архітектуру застосунку, що ґрунтується на поєднанні Python-бекенда та React-інтерфейсу, інтегрованих у єдине настільне рішення завдяки PyWebView. Такий підхід дозволив забезпечити одночасно високу продуктивність обробки, простоту взаємодії та кросплатформність.

У підрозділах розділу було описано внутрішню структуру проєкту, включно з модулем захоплення зображень, підсистемою виклику інференсу, компонентами для обробки статистики, а також REST-ендпоінтами, які реалізують взаємодію між інтерфейсом і серверною частиною. Окрему увагу приділено користувацькому інтерфейсу, який забезпечує інтуїтивну взаємодію з системою, доступ до статистики, налаштувань чутливості та контролю запуску чи зупинки детекції.

У межах тестування було оцінено продуктивність, стабільність та точність роботи системи в реальних умовах. Результати підтвердили, що обраний підхід із періодичним захопленням кадрів та хмарним інференсом забезпечує задовільний рівень відгуку й точності для практичного використання. Водночас тестування виявило ряд системних обмежень, пов'язаних із чутливістю до освітлення, якістю вебкамери, затримкою між моментом перенапруження і сповіщенням, а також залежністю системи від наявності інтернет-з'єднання та зовнішнього ML-сервісу.

Аналіз обмежень дав змогу сформувані обґрунтовані перспективи розвитку, серед яких ключовими є перехід до локального інференсу, оптимізація моделей для роботи на пристроях користувача, розширення набору детектованих мімічних ознак та використання динамічних моделей для аналізу послідовностей кадрів. Реалізація цих напрямів дозволить не лише підвищити точність і швидкодію системи, а й значною мірою покращити приватність та автономність застосування.

Таким чином, практична частина роботи підтвердила життєздатність запропонованого рішення, продемонструвала його ефективність у реальних умовах та окреслила конкретні технічні напрямки, що визначають подальшу еволюцію системи Frown Bully як інструмента аналізу мімічного та м'язового перенапруження.

РОЗДІЛ 4 — ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Фактори виробничого середовища і їх вплив на життєдіяльність людини

Фактори виробничого середовища відіграють визначальну роль у формуванні безпечних та комфортних умов праці фахівців у сфері комп'ютерних наук, зокрема під час розробки та експлуатації програмних систем, орієнтованих на тривалу роботу за персональним комп'ютером. У межах проєкту Frownbully виробниче середовище формується умовами офісного або домашнього робочого простору, використанням комп'ютерної техніки, відеодисплейних терміналів і периферійних пристроїв, а також особливостями інтелектуального та психоемоційного навантаження. Незважаючи на відсутність важких промислових чинників, сукупний вплив фізичних, ергономічних і психофізіологічних факторів може істотно впливати на життєдіяльність, працездатність і стан здоров'я людини, що обумовлює необхідність їх системного аналізу відповідно до вимог законодавства з охорони праці.

Загальні принципи оцінювання умов праці та впливу виробничих факторів визначаються Законом України «Про охорону праці», який покладає обов'язок забезпечення безпечних і нешкідливих умов праці, профілактики професійних ризиків та збереження здоров'я працівників [49]. У практичному аспекті для аналізу таких умов доцільно застосовувати підхід гігієнічної класифікації праці, що дозволяє оцінювати рівні факторів виробничого середовища та напруженість трудового процесу з урахуванням їхнього впливу на функціональний стан організму. Саме в цій логіці розглядаються основні чинники, характерні для діяльності, пов'язаної з розробкою і використанням системи Frownbully.

Передусім слід зосередитися на фізичних факторах виробничого середовища, які безпосередньо формують комфорт і безпеку перебування людини на робочому місці. Одним із ключових таких факторів є мікроклімат приміщення, що включає температуру, відносну вологість і швидкість руху повітря. Відповідність цих параметрів санітарним нормам є необхідною умовою підтримання оптимального теплового балансу організму, запобігання перевтомі

та зниженню концентрації уваги. Для робіт, пов'язаних з інтенсивною обробкою відеоданих у реальному часі, як у системі Frownbully, стабільний мікроклімат має додаткове значення, оскільки впливає не лише на самопочуття користувача, але й на надійність функціонування комп'ютерного обладнання, запобігаючи його перегріванню або накопиченню статичної електрики. Вимоги до параметрів мікроклімату встановлені державними санітарними нормами, зокрема ДСН 3.3.6.042-99 [50].

Наступним важливим фізичним чинником є освітлення робочої зони, яке безпосередньо впливає на стан зорового аналізатора та швидкість розвитку зорової втоми. Робота з програмним кодом, графічними інтерфейсами та відеопотоком передбачає тривалу фіксацію погляду на екрані монітора, що вимагає рівномірного, достатнього за інтенсивністю освітлення без різких контрастів і відблисків. Недотримання цих вимог призводить до перевантаження органів зору, зниження точності виконуваних завдань і зростання ймовірності помилок. Нормативні вимоги до природного і штучного освітлення робочих приміщень регламентуються ДБН В.2.5-28:2018, які доцільно враховувати під час організації робочого місця користувача системи Frownbully.

Окрему групу фізичних факторів становлять акустичні впливи, зокрема виробничий шум. Хоча офісні та домашні робочі приміщення не характеризуються високими рівнями шуму, постійний фоновий шум від системних блоків, вентиляційних систем або зовнішніх джерел може спричиняти підвищену втомлюваність, дратівливість і зниження когнітивної ефективності. Для забезпечення безпечних умов праці допустимі рівні шуму визначаються державними санітарними нормами, зокрема ДСН 3.3.6.037-99, дотримання яких сприяє підтриманню психофізіологічного комфорту працівника.

Поряд із фізичними чинниками суттєвий вплив на життєдіяльність людини мають ергономічні фактори, що пов'язані з організацією робочого місця та характером взаємодії з комп'ютерною технікою. Діяльність розробника та користувача системи Frownbully супроводжується тривалою статичною робочою позою, монотонними рухами кистей при роботі з клавіатурою і мишею, а також постійним навантаженням на м'язи шії та спини. За відсутності ергономічно

обґрунтованої організації робочого місця це може призводити до розвитку болю в опорно-руховому апараті, порушень постави та зниження загальної працездатності. Законодавство з охорони праці та положення гігієнічної класифікації праці передбачають необхідність урахування таких факторів як складової напруженості трудового процесу.

Не менш важливими є психофізіологічні фактори, що супроводжують інтелектуальну діяльність у сфері комп'ютерних наук. Розробка, налагодження та тестування системи Frownbully вимагають високої концентрації уваги, швидкого аналізу інформації, прийняття рішень у режимі реального часу та тривалого когнітивного навантаження. Це створює умови для розвитку психоемоційного напруження, втоми та стресових станів, особливо за умови жорстких часових обмежень або тривалих сесій без перерв. У межах гігієнічного підходу такі чинники розглядаються як елементи напруженості праці, які при систематичному впливі можуть негативно позначатися на здоров'ї та якості життя працівника.

Окремої уваги заслуговують фактори, пов'язані з роботою з відеодисплейними терміналами. Хоча раніше ці аспекти детально регламентувалися ДСанПіН 3.3.2-007-98, на сьогодні цей документ втратив чинність, однак закладені в ньому принципи ергономіки та гігієни праці зберігають практичне значення й можуть використовуватися як орієнтир під час організації робочого місця. У контексті системи Frownbully це особливо актуально з огляду на тривалу роботу з дисплеєм і візуальними даними, що потребує підвищеної уваги до параметрів зображення та режиму праці [51].

Таким чином, сукупний вплив фізичних, ергономічних і психофізіологічних факторів виробничого середовища формує умови, які безпосередньо впливають на життєдіяльність, працездатність і психологічний комфорт людини. Для системи Frownbully ці фактори мають особливе значення, оскільки некоректно організоване середовище праці може не лише погіршувати самопочуття користувача, але й опосередковано впливати на результати детекції мімічного напруження та сприйняття сповіщень. Саме тому аналіз виробничого середовища є необхідною основою для подальшого обґрунтування заходів з

охорони праці та безпечної експлуатації системи відповідно до чинних нормативно-правових вимог.

4.2 Психоемоційна безпека та профілактика професійного стресу при роботі з інтелектуальними системами

У сучасних умовах інтенсивного використання цифрових технологій психоемоційне навантаження при тривалій роботі з відеопотоком і системами реального часу стає важливим фактором, що впливає на психологічне та функціональне благополуччя користувачів. Дослідження в галузі цифрової психології показують, що «цифрова втома» або *digital fatigue* — це стан емоційного та фізичного виснаження, який виникає внаслідок тривалого залучення до цифрових інструментів і великих обсягів інформації, і який пов'язаний із зниженням продуктивності та зростанням стресу в цифровому середовищі праці. Такі явища чітко фіксуються в сучасних дослідженнях, де тривала взаємодія з інтерфейсами і обробка інформації призводить до зниження здатності концентрувати увагу та психологічного комфорту. За даними наукових публікацій, взаємозв'язок між цифровою втомою і когнітивним навантаженням є позитивним і значущим: чим вище цифрове навантаження, тим сильніше виражена когнітивна перевтома та зниження ефективності виконання завдань у цифровому середовищі — це підтверджується емпіричними даними 2025 року, де *digital fatigue* і *cognitive overload* прогнозують зниження продуктивності в умовах інтенсивної цифрової праці, зокрема в ІТ-секторі та інших галузях, де інформаційні потоки є постійними і динамічними [52].

У тісному зв'язку з цим перебуває феномен когнітивної перевтоми, що виникає, коли обсяг інформації перевищує здатність робочої пам'яті до її оброблення. Когнітивне перевантаження в сучасних умовах цифрової роботи є наслідком одночасного маніпулювання великою кількістю інформаційних потоків, багатозадачності та постійних вимог організувати увагу на змінних вхідних даних. Як показують дослідження в галузі психології праці, таке перевантаження призводить до зниження глибини обробки інформації, зростання

часу реакції й зниження здатності ухвалювати точні рішення, що особливо критично під час тривалого моніторингу відеопотоку та взаємодії із системами реального часу, які потребують оперативних реакцій і високого рівня уваги. При цьому навіть ті форми цифрових робочих засобів, які не мають безпосереднього відеовиведення, можуть посилювати когнітивний тиск через накопичення інформаційних вимог, що в сукупності з емоційним навантаженням створює комплексний психокогнітивний стрес для користувача [53].

Ще одним важливим аспектом впливу на психологічний комфорт користувача є характер і частота системних сповіщень. Часті або хибні сповіщення, що не узгоджуються з контекстом виконуваної задачі, можуть виступати як постійні переривання основного потоку роботи, що порушують концентрацію, спричиняють фрагментацію уваги й призводять до додаткового навантаження на когнітивні ресурси. Наукові підходи з так званої *interruption science* (науки про переривання) свідчать, що інформаційні переривання можуть значно погіршувати здатність до стійкої концентрації, підвищувати психофізіологічний стрес і сприяти появі почуття втоми та роздратованості, коли людина вимушена перемикається між завданнями або непродуктивно відволікатися на непотрібні повідомлення. Це особливо актуально для програм, які інтегрують механізми оповіщення про події або стан системи: часті сповіщення без належної фільтрації важливих від неважливих сигналів можуть не лише перевантажувати увагу, а й безпосередньо впливати на загальний стан психологічного комфорту користувача.

У цьому контексті роль програмних засобів, зокрема систем, таких як *Frownbully*, стає не лише технічною, а й психокогнітивною. Ефективне програмне забезпечення має не лише вирішувати поставлені функціональні задачі, але й зменшувати, а не посилювати психоемоційне та когнітивне навантаження користувача. Під час проєктування таких систем важливо враховувати принципи людсько-орієнтованого дизайну, включати адаптивні механізми, що регулюють частоту й обґрунтованість сповіщень залежно від контексту виконуваних завдань, та забезпечувати можливості налаштування профілю взаємодії. Такий підхід дозволяє зменшувати інформаційне

перевантаження, підтримувати увагу в робочому режимі та знижувати психоемоційний стрес у тривалому процесі взаємодії з відеопотоком і системами реального часу. У свою чергу, це сприяє не тільки підвищенню продуктивності праці та якості взаємодії з програмним засобом, але й підтриманню психологічного комфорту користувача, що є важливою складовою сучасних вимог до безпеки і охорони праці в умовах цифрової діяльності.

4.3. Безпека користувача під час експлуатації програмної системи

Безпека користувача під час експлуатації програмної системи в сучасних умовах виходить за межі виключно фізичної або електробезпеки та охоплює аспекти психологічного комфорту, коректності взаємодії людини з інтерфейсом і впливу програмних рішень на прийняття рішень користувачем. Для інтелектуальних систем, що аналізують поведінкові або мімічні ознаки та формують сповіщення у режимі реального часу, як у випадку з системою Frownbully, ці аспекти набувають особливої актуальності. Некоректна або непродумана взаємодія з користувачем може створювати додаткові ризики, пов'язані не з технічними збоями, а з помилковим сприйняттям інформації та психологічним дискомфортом.

Одним із ключових ризиків у процесі експлуатації систем детекції є можливість неправильної інтерпретації результатів аналізу. Результати, які надає програмна система, за своєю природою мають імовірнісний характер і залежать від якості вхідних даних, умов освітлення, індивідуальних особливостей користувача та обмежень самої моделі. У разі відсутності належного пояснення або контексту користувач може сприймати результати детекції як однозначні або діагностичні, що створює ризик хибних висновків щодо власного психоемоційного стану. З погляду безпеки це є небажаним, оскільки може впливати на поведінку користувача, його емоційний стан та ставлення до роботи.

Тісно пов'язаним із цим є вплив помилкових або надмірно частих сповіщень на психологічний комфорт користувача [54]. Часті повідомлення, які не відповідають суб'єктивному сприйняттю стану або виникають у

невідповідний момент, можуть викликати роздратування, тривожність або відчуття постійного контролю. Дослідження в галузі взаємодії людини з комп'ютером свідчать, що надлишкові сповіщення є одним із чинників зниження довіри до системи та зростання когнітивного навантаження, оскільки вони переривають основну діяльність і змушують користувача постійно перемикає увагу. У контексті охорони праці це може розглядатися як фактор, що негативно впливає на психоемоційний стан і загальну працездатність.

З огляду на це важливою складовою безпечної експлуатації програмної системи є інформування користувача про її функціональні обмеження. Користувач повинен чітко усвідомлювати, що система Frownbully не є медичним або діагностичним інструментом, а виконує допоміжну функцію моніторингу й самопостереження. Чітке формулювання призначення системи, пояснення принципів її роботи та можливих похибок детекції знижує ризик неправильного трактування результатів і сприяє формуванню адекватних очікувань. Такий підхід відповідає сучасним принципам безпечного проєктування програмних систем, орієнтованих на людину.

У цьому контексті важливим є дотримання принципу «підтримки, а не контролю» у програмних інтерфейсах. Інтерфейс системи має сприйматися користувачем як інструмент допомоги та підвищення усвідомленості, а не як механізм постійного нагляду чи оцінювання [55]. Людиноорієнтований дизайн передбачає мінімізацію нав'язливих елементів, нейтральний тон повідомлень і надання користувачу можливості самостійно вирішувати, як реагувати на отриману інформацію. Такий підхід знижує рівень психологічного напруження та підвищує довіру до системи, що є важливою складовою безпечної взаємодії.

Окрему роль у забезпеченні безпеки користувача відіграє можливість налаштування параметрів чутливості системи. Наявність регульованих порогів спрацювання дозволяє адаптувати роботу Frownbully до індивідуальних особливостей користувача, його стилю роботи та рівня сприйнятливості до сповіщень. Це зменшує кількість хибних або надмірних повідомлень і, відповідно, знижує ризик психологічного дискомфорту. З погляду охорони праці

така адаптивність є важливою, оскільки дозволяє інтегрувати програмну систему у робочий процес без створення додаткових факторів напруженості.

Таким чином, безпека користувача під час експлуатації програмної системи Frownbully полягає не лише у коректній технічній роботі програмного забезпечення, а й у забезпеченні психологічного комфорту, прозорості взаємодії та підтримці автономії користувача. Урахування ризиків неправильної інтерпретації результатів, мінімізація негативного впливу сповіщень, інформування про обмеження системи та застосування принципів людиноорієнтованого дизайну дозволяють розглядати Frownbully як безпечний допоміжний інструмент, що сприяє збереженню працездатності та психоемоційного благополуччя користувача в умовах тривалої роботи з комп'ютером.

У кваліфікаційній роботі освітнього рівня «Магістр» досягнуто поставленої мети та розв'язано комплекс наукових і практичних завдань, пов'язаних із дослідженням і розробленням програмної системи детекції мімічних напружень на основі методів комп'ютерного зору.

У першому розділі роботи подано загальну характеристику предметної області та обґрунтовано актуальність задачі автоматизованого виявлення мімічних напружень в умовах зростання інтенсивності та тривалості взаємодії людини з цифровими пристроями. Розглянуто сучасні підходи до аналізу обличчя й мімічних ознак із використанням методів комп'ютерного зору та машинного навчання, висвітлено ключові проблеми й обмеження існуючих рішень, зокрема питання точності, інтерпретованості результатів і впливу систем моніторингу на користувацький досвід. Проаналізовано функціональні та нефункціональні вимоги до програмної системи, досліджено типові сценарії її використання та потенційні ризики експлуатації, а також обґрунтовано вибір desktop-first архітектури з локальним інференсом як оптимального рішення з погляду продуктивності та збереження приватності. У результаті сформовано цілі, обмеження й загальну концепцію розроблюваної системи.

У другому розділі описано теоретичні засади та методи, покладені в основу системи, зокрема підходи до об'єктної детекції та аналізу мімічних характеристик у відеопотоці. Досліджено можливості застосування згорткових нейронних мереж для задач реального часу з урахуванням обмежень обчислювальних ресурсів, а також подано порівняльний аналіз наявних моделей і архітектур. На основі цього аналізу обґрунтовано вибір методів і моделей, які найкраще відповідають вимогам точності, стабільності роботи та інтерпретованості результатів.

У третьому розділі розроблено програмну систему детекції мімічних напружень, що поєднує алгоритми комп'ютерного зору з механізмами прийняття рішень. Запропоновано підхід до формування сповіщень на основі налаштовуваних порогів чутливості, який забезпечує адаптацію системи до

індивідуальних особливостей користувача та знижує ризик хибних спрацьовувань. Спроектовано архітектуру клієнтської частини системи, орієнтовану на безперервний моніторинг у реальному часі, локальну обробку даних і мінімізацію затримок. Проведено тестування системи в типових сценаріях використання, що дозволило оцінити її функціональність, стабільність роботи та зручність взаємодії з користувачем.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» проаналізовано умови праці під час розробки й експлуатації програмної системи, а також можливі ризики для здоров'я користувачів і розробників. Описано заходи з охорони праці, пожежної безпеки та безпеки під час експлуатації системи, а також наведено рекомендації щодо зниження негативного впливу тривалої роботи з програмними засобами та відеопотоками.

1. Shah N., Asuncion R. M. D., Hameed S. Muscle Contraction Tension Headache / N. Shah, R. M. D. Asuncion, S. Hameed // *StatPearls* [Electronic resource]. — NCBI Bookshelf. — Режим доступу: <https://www.ncbi.nlm.nih.gov/books/NBK562274/>
2. Zebrowitz L. A., Franklin R. G. Jr. The Attractiveness Halo Effect and the Babyface Stereotype in Older and Younger Adults: Similarities, Own-Age Accentuation, and OA Positivity Effects / L. A. Zebrowitz, R. G. Franklin Jr. // *Psychology and Aging*. — 2014. — Vol. 29, No. 3. — P. 346–357. — Режим доступу: <https://pmc.ncbi.nlm.nih.gov/articles/PMC4020290/>
3. World Health Organization. Ageing and health // *WHO Fact Sheets* [Electronic resource]. — 1 October 2025. — Режим доступу: <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health>
4. Di Santis É. P., Hirata S. H., Di Santis G. M., Yarak S. Adverse effects of the aesthetic use of botulinum toxin and dermal fillers on the face: a narrative review / É. P. Di Santis, S. H. Hirata, G. M. Di Santis, S. Yarak // *Anais Brasileiros de Dermatologia*. — 2024. — Vol. 99, No. 3. — P. 261–270. — Режим доступу: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11745296/>
5. Funt D., Pavicic T. Dermal fillers in aesthetics: an overview of adverse events and treatment approaches / D. Funt, T. Pavicic // *Clinical, Cosmetic and Investigational Dermatology*. — 2013. — Vol. 6. — P. 295–316. — Режим доступу: <https://pmc.ncbi.nlm.nih.gov/articles/PMC3865975/>
6. Yuk Y.-S., Lim H.-W. Kinesio taping: a novel approach to reducing facial wrinkles and sagging skin / Y.-S. Yuk, H.-W. Lim // *Aesthetic Medicine*. — 2023. — Vol. 9, No. 4. — Режим доступу: <https://www.mattioli1885journals.com/index.php/aestheticmedicine/article/view/16352>
7. Pałka N., Maciejewski M., Kamiński K., Piszczek M., Zagrajek P., Czerwińska E., Walczakowski M., Dragan K., Synaszko P., Świdorski W. Fast THz-TDS Reflection Imaging with ECOPS—Point-by-Point versus Line-by-Line Scanning / N. Pałka, M. Maciejewski, K. Kamiński, M. Piszczek, P. Zagrajek,

- E. Czerwińska, M. Walczakowski, K. Dragan, P. Synaszko, W. Świdorski // *Sensors*. — 2022. — Vol. 22, No. 22. — Art. No. 8813. — Режим доступа: <https://www.mdpi.com/1424-8220/22/22/8813>
8. Pal S. Human Face Detection Technique using Haar-like Features / S. Pal // *International Journal of Computer Applications*. — 2020. — Vol. 175, No. 32. — P. 56–60. — DOI: 10.5120/ijca2020920883. — Режим доступа: https://www.researchgate.net/publication/347029136_Human_Face_Detection_Technique_using_Haar-like_Features
9. Singh H., Gagandeep. Local Binary Patterns and Its Application to Facial Analysis / H. Singh, Gagandeep // *International Journal of Engineering Research and Reviews*. — 2022. — Vol. 10, Issue 3. — P. 11–20. — Режим доступа: <https://www.researchpublish.com/upload/book/Local%20Binary%20Patterns%20and%20Its%20Application-22082022-2.pdf>
10. Kazemi V., Sullivan J. One Millisecond Face Alignment with an Ensemble of Regression Trees / V. Kazemi, J. Sullivan // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. — 2014. — P. 1867–1874. — Режим доступа: https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Kazemi_One_Millisecond_Face_2014_CVPR_paper.pdf
11. Kartynnik Y., Ablavatski A., Grishchenko I., Grundmann M. Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs / Y. Kartynnik, A. Ablavatski, I. Grishchenko, M. Grundmann // *arXiv preprint arXiv:1907.06724*. — 2019. — Режим доступа: <https://arxiv.org/abs/1907.06724>
12. Bodini M. A Review of Facial Landmark Extraction in 2D Images and Videos Using Deep Learning / M. Bodini // *Machine Learning and Knowledge Extraction (MAKE)*. — 2021. — Vol. 3, No. 1. — P. 236–262. — Режим доступа: <https://www.mdpi.com/2504-2289/3/1/14>
13. Facial Action Coding System // *Machine Learning for Biometrics* [Electronic resource]. — 2022. — Режим доступа:

<https://www.sciencedirect.com/topics/computer-science/facial-action-coding-system>

14. Zhang L., Arandjelović O. Review of Automatic Microexpression Recognition in the Past Decade / L. Zhang, O. Arandjelović // *Biomimetics*. — 2018. — Vol. 3, No. 2. — Art. No. 21. — Режим доступа: <https://www.mdpi.com/2504-4990/3/2/21>
15. Zhou L., Shao X., Mao Q. A survey of micro-expression recognition / L. Zhou, X. Shao, Q. Mao // *Image and Vision Computing*. — 2021. — Vol. 105. — Art. No. 104043. — Режим доступа: <https://www.sciencedirect.com/science/article/abs/pii/S026288562030175X>
16. What can I use FaceReader for? // *Noldus — FaceReader Applications* [Electronic resource]. — Режим доступа: <https://noldus.com/facereader/applications>
17. Emotion AI 101: All About Emotion Detection and Affectiva's Emotion Metrics // *Smart Eye* [Electronic resource]. — Режим доступа: <https://go.smarteye.se/emotion-ai-101-all-about-emotion-detection-and-affectivas-emotion-metrics>
18. Azure Face detection, attributes, and input data // *Microsoft Learn* [Electronic resource]. — Режим доступа: <https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/concept-face-detection>
19. MediaPipe Solutions Guide // *Google AI* [Electronic resource]. — Режим доступа: <https://ai.google.dev/edge/mediapipe/solutions/guide>
20. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement / J. Redmon, A. Farhadi // *arXiv preprint arXiv:1804.02767*. — 2018. — Режим доступа: <https://arxiv.org/abs/1804.02767>
21. Bochkovskiy A., Wang C.-Y., Liao H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection / A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao // *arXiv preprint arXiv:2004.10934*. — 2020. — Режим доступа: <https://arxiv.org/abs/2004.10934>

22. Zhang C., Lin H., Zhang Y. YOLOv7: The State of the Art for Real-Time Object Detection / C. Zhang, H. Lin, Y. Zhang // *arXiv preprint arXiv:2207.02696*. — 2023. — Режим доступу: <https://arxiv.org/abs/2207.02696>
23. Nath G. S., Veena A. A Deep Learning Based Detection of Wrinkles on Skin / G. S. Nath, A. Veena // *Advances in Intelligent Systems and Computing: Proc. of the 5th International Conference on Computational Vision and Bio Inspired Computing (ICCVBIC 2021)*, RVS Technical Campus, Coimbatore, India, November 2021. — DOI: 10.1007/978-981-16-9573-5_2. — Режим доступу: https://www.researchgate.net/publication/361275360_A_Deep_Learning_Based_Detection_of_Wrinkles_on_Skin
24. YouCam Makeup – Selfie Editor / Perfect Mobile Corp. // *Google Play* [Electronic resource]. — Режим доступу: <https://play.google.com/store/apps/details?id=com.cyberlink.youcammakeup>
25. FaceApp: редактор обличчя / FaceApp Technology Ltd // *Google Play* [Electronic resource]. — Режим доступу: <https://play.google.com/store/apps/details?id=io.faceapp>
26. Hosseini H., Xiao B., Poovendran R. Google's Cloud Vision API Is Not Robust To Noise / H. Hosseini, B. Xiao, R. Poovendran // *arXiv preprint arXiv:1704.05051*. — 2017. — Режим доступу: <https://arxiv.org/abs/1704.05051>
27. Харченко О., Яцишин В. Розробка та керування вимогами до програмного забезпечення на основі моделі якості. Вісник ТДТУ. Тернопіль, 2009. Т. 14. №1. С. 201-207.
28. Yatsyshyn V., Pastukh O., Palamar A., Zharovsky R. Technology of relational database management systems performance evaluation during computer systems design. Scientific Journal of TNTU. 2023. Vol. 109, No 1. P. 54–65.
29. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // *arXiv preprint arXiv:1506.02640*. — 2015. — Режим доступу: <https://arxiv.org/abs/1506.02640>
30. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. C. SSD: Single Shot MultiBox Detector / W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.

- Reed, C.-Y. Fu, A. C. Berg // *arXiv preprint* arXiv:1512.02325. — 2015. —
Режим доступу: <https://arxiv.org/abs/1512.02325>
31. Su Z. Comparative Analysis of CNN-Based Object Detection Models: Faster R-CNN, SSD, and YOLO / Z. Su // *Highlights in Science, Engineering and Technology*. — 2025. — Vol. 138. — P. 147–152. — DOI: 10.54097/9r6evm71.
— Режим доступу: https://www.researchgate.net/publication/391692075_Comparative_Analysis_of_CNN-Based_Object_Detection_Models_Faster_R-CNN_SSD_and_YOLO
32. Rezaei J. *Wrinkle-detection Computer Vision Model* / J. Rezaei // *Roboflow Universe* [Electronic resource]. — Режим доступу: <https://universe.roboflow.com/javad-rezaei/wrinkle-detection/>
33. chrome.tabCapture // *Chrome Developers Documentation* [Electronic resource]. — Режим доступу: <https://developer.chrome.com/docs/extensions/mv2/reference/tabCapture>
34. Satyanarayanan M. The Emergence of Edge Computing / M. Satyanarayanan // *Computer*. — 2017. — Vol. 50, No. 1. — P. 30–39. — DOI: 10.1109/MC.2017.9. — Режим доступу: <https://ieeexplore.ieee.org/document/7807196>
35. FastAPI — Official Documentation [Electronic resource]. — Режим доступу: <https://fastapi.tiangolo.com>
36. SQLite Documentation — Architecture & Features [Electronic resource]. — Режим доступу: <https://www.sqlite.org/docs.html>
37. PyWebView — Official Documentation [Electronic resource]. — Режим доступу: <https://pywebview.flowrl.com/>
38. GDPR — General Data Protection Regulation [Electronic resource]. — Режим доступу: <https://gdpr.eu/>
39. Edge vs Cloud AI: Key Differences, Benefits & Hybrid Future / Clarifai // *Clarifai Blog* [Electronic resource]. — Режим доступу: <https://www.clarifai.com/blog/edge-vs-cloud-ai>
40. Петрик М. Р., Петрик О. Ю. Моделювання програмного забезпечення :
наук.-метод. посіб. Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2015. 200
с.

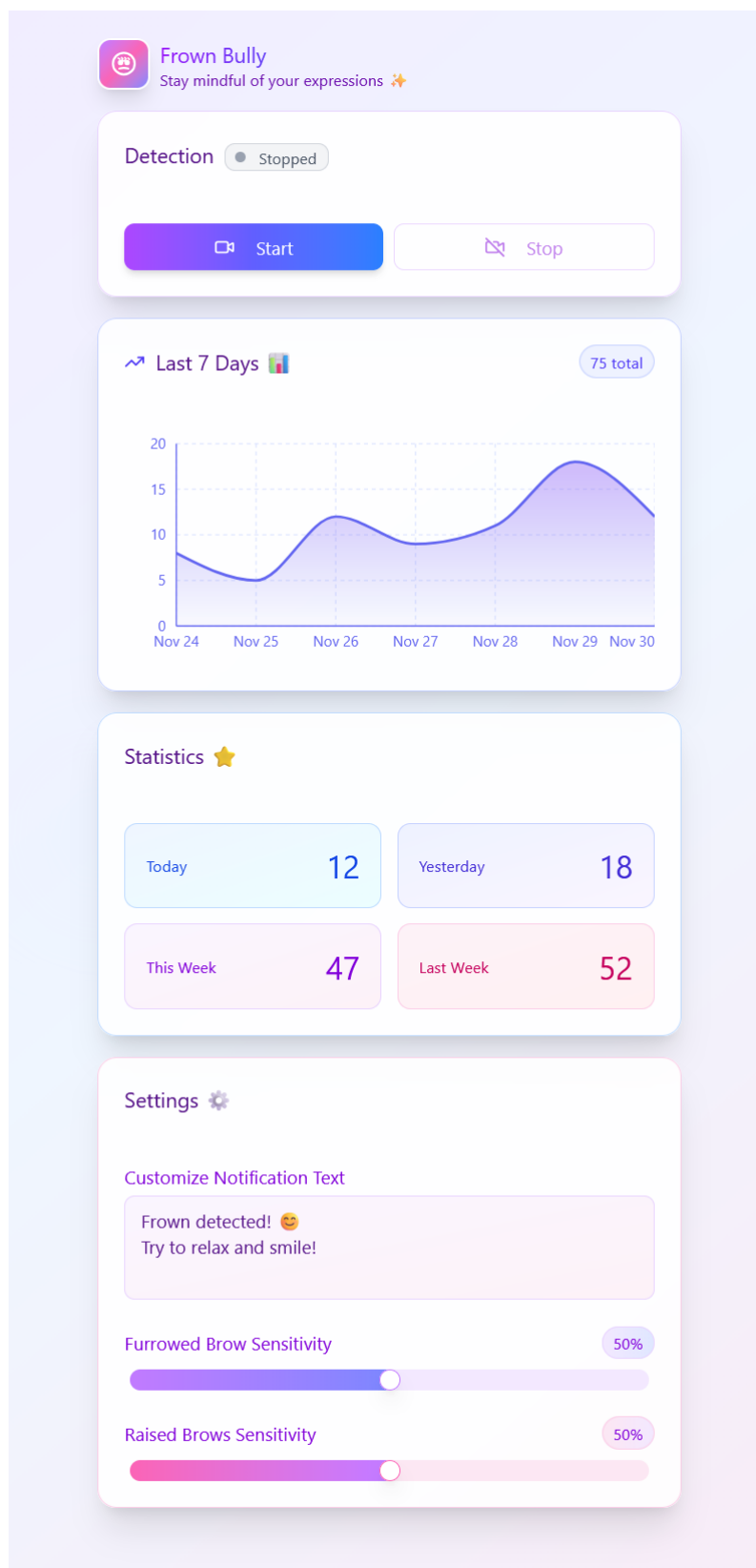
41. Zhang T., El Ali A., Hanjalic A., Cesar P. *Few-Shot Learning for Fine-Grained Emotion Recognition Using Physiological Signals* / T. Zhang, A. El Ali, A. Hanjalic, P. Cesar // *IEEE Transactions on Affective Computing*. — 2023. — Early Access / or 2022 depending on version on IEEE Xplore*. — DOI: 10.1109/TAFFC.2022.3175934. — Режим доступа: <https://ieeexplore.ieee.org/abstract/document/9751421>
42. Lypa B., Horyn I., Zagorodna N., Tymoshchuk D., Lechachenko T. *Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings*. 2024. vol. 3896. P. 1-11.
43. Ekman P., Rosenberg E. L. *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*. — 2nd ed. / P. Ekman, E. L. Rosenberg. — Oxford: Oxford University Press, 2005. — 512 p. — Режим доступа: <https://www.aqualide.com/upload/texte/text98.pdf>
44. Konovalenko I., Maruschak P., Brezinová J., Prentkovskis O., Brezina J. Research of U-Net-based CNN architectures for metal surface defect detection. *Machines*, 2022. 10(5), P. 327.
45. Seikavandi M. J., Fimland J., Barrett M., Burelli P. Exploring the Temporal Dynamics of Facial Mimicry in Emotion Processing Using Action Units / M. J. Seikavandi, J. Fimland, M. Barrett, P. Burelli // *arXiv preprint arXiv:2503.17306*. — 2025. — Режим доступа: <https://arxiv.org/html/2503.17306v1>
46. Han S., Mao H., Dally W. J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding / S. Han, H. Mao, W. J. Dally. — 2016. — Режим доступа: <https://hanlab.mit.edu/projects/deep-compression>
47. Li Z., Li H., Meng L. Model Compression for Deep Neural Networks: A Survey / Z. Li, H. Li, L. Meng // *Computers*. — 2023. — Vol. 12, No. 3. — Art. No. 60. — Режим доступа: <https://www.mdpi.com/2073-431X/12/3/60>
48. Dantas P. V., Sabino da Silva Jr W., Cordeiro L. C., Carvalho C. B. A comprehensive review of model compression techniques in machine learning /

- P. V. Dantas, W. Sabino da Silva Jr, L. C. Cordeiro, C. B. Carvalho // *Applied Intelligence*. — 2024. — Vol. 54. — P. 11804–11844. — Published: 02 September 2024. — DOI: 10.1007/s10489-024-05747-w. — Режим доступу: <https://link.springer.com/article/10.1007/s10489-024-05747-w>
49. Закон України «Про охорону праці» : Закон України від 14 жовтня 1992 р. № 2694-XII (зі змінами і доповненнями) // *Відомості Верховної Ради України*. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12>
50. Державні санітарні норми мікроклімату виробничих приміщень : ДСН 3.3.6.042-99, затверджені постановою Головного державного санітарного лікаря України // *Офіційний вебпортал Верховної Ради України* [Electronic resource]. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/va042282-99>
51. Державні санітарні правила і норми роботи з відеодисплейними терміналами (ВДТ), електронно-обчислювальними машинами (ЕОМ) : ДСанПіН 3.3.2-007-98 (втратив чинність) // *Офіційний вебпортал Верховної Ради України* [Electronic resource]. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0283-98>
52. Imran H., Shafique F. Digital Fatigue and Cognitive Overload as Managerial Challenges: Understanding the Impact of Workload Management on the Workforce / H. Imran, F. Shafique // *The Critical Review of Social Sciences Studies*. — 2025. — Vol. 3, No. 4. — P. 1629–1641. — Режим доступу: https://www.researchgate.net/publication/398274269_Digital_Fatigue_and_Cognitive_Overload_as_Managerial_Challenges_Understanding_the_Impact_of_Workload_Management_on_the_Workforce
53. Schmitt J. B., Breuer J., Wulf T. From cognitive overload to digital detox: Psychological implications of telework during the COVID-19 pandemic / J. B. Schmitt, J. Breuer, T. Wulf // *Computers in Human Behavior Reports*. — 2021. — Vol. 4. — Art. No. 100091. — Режим доступу: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8455170/>

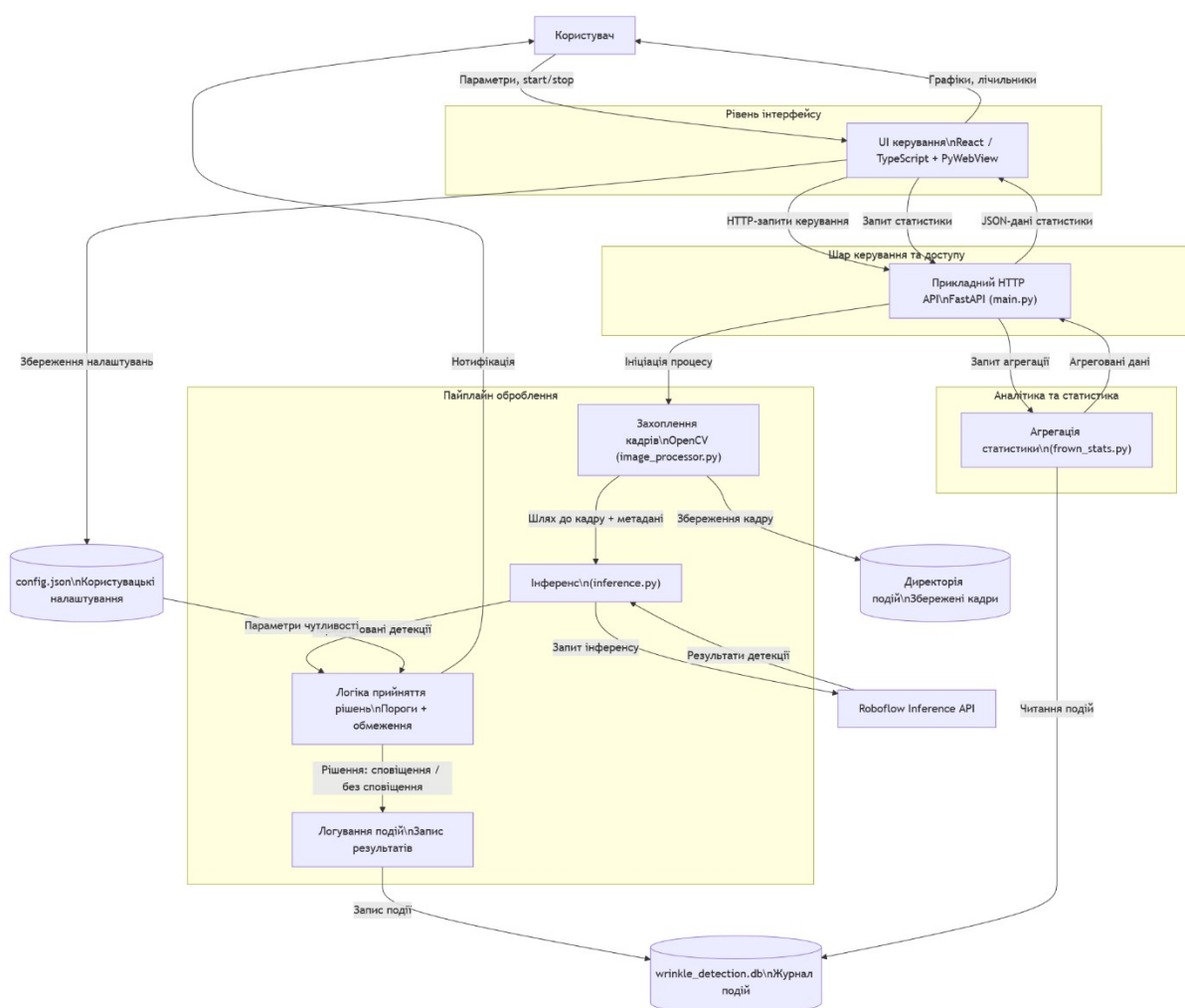
54. Notifications and Alerts: User Experience Guidelines / Nielsen Norman Group
// *NN/g Articles* [Electronic resource]. — Режим доступа:
<https://www.nngroup.com/articles/notifications/>
55. Human-Centered Design: Designing for User Trust and Transparency // *Microsoft Learn* [Electronic resource]. — Режим доступа:
<https://learn.microsoft.com/en-us/windows/apps/design/signals/user-trust>

ДОДАТКИ

Макет Системи



Діаграма потоку даних



ТЕЗИ КОНФЕРЕНЦІЙ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
 ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
 ІМЕНІ ІВАНА ПУЛЮКА

МАТЕРІАЛИ
 XIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ
 «ІНФОРМАЦІЙНІ МОДЕЛІ,
 СИСТЕМИ ТА ТЕХНОЛОГІЇ»



17-18 грудня 2025 року

ТЕРНОПІЛЬ
 2025

УДК 004.932

К. ~~Володимир~~ А. Литвиненко, д. т. н., проф.
 (Тернопільський національний технічний університет імені Івана Пулюка, Україна)

РОЗРОБКА ДЕСКТОПНОЇ СИСТЕМИ ДЛЯ ДЕТЕКЦІЇ МІМІЧНИХ НАПРУЖЕНЬ НА
 ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ

UDC 004.932

K. Holubko; Y. Lytvynenko, Dr., Prof.

DEVELOPMENT OF A DESKTOP SYSTEM FOR DETECTING FACIAL MUSCLE
 TENSION BASED ON COMPUTER VISION

Персональні цифрові помічники дедалі частіше інтегрують технології комп'ютерного зору для моніторингу станів користувача, включно з мікро виразами обличчя, які можуть впливати на фізичне та емоційне самопочуття. Мимовільне нахмурення є одним із таких проявів, що пов'язаний із підвищенням м'язовим напруженням, головним болем та формуванням глибоких мимічних зморшок. Це створює потребу у розробці локальних інтелектуальних систем, здатних у реальному часі детектувати мимічні патерни та формувати корисний зворотний зв'язок для користувача.

Метою роботи є створення настільної системи, що використовує методи комп'ютерного зору та «легкі» моделі об'єктної детекції для автоматичного виявлення нахмурення на основі відеопотоку вебкамери, формування миттєвих сповіщень та накопичення статистичних мимічної активності.

Архітектура системи включає:

1. модуль захоплення зображення з веб камери з періодом 2 секунди а також обробки результати роботи модуля комп'ютерного зору і прийняття рішення про сповіщення на основі впевненості моделі і визначення користувачем порогів чутливості;
2. модуль комп'ютерного зору, побудований на моделі об'єктної детекції;
3. підсистему зберігання даних (SQLite) для побудови щоденної та тижневої статистики;
4. десктопний UI (FastAPI + PyWebView), що забезпечує керування детекцією та перегляд статистики.

Поліг «нахмурення» фіксується за таких умов:

- модель визначає класи *frown*, *forehead line* із певною впевненістю p ,
- користувач у графічному інтерфейсі задає поріг чутливості T ,
- система порівнює значення впевненості по кожному з класів з відповідними порогами і якщо хоч один клас досягає або перевищує поріг то надсилається сповіщення:

$$p \geq T$$

Таким чином, система не лише реагує на значення, видані моделлю, а й адаптується до індивідуальних потреб користувача, дозволяючи зробити детекцію більш суворою або м'якшою.

Проведення тестування продемонструвало стабільність інференсу та коректність роботи моделі в умовах неоднорідного освітлення, різних ракурсів та природних мимічних варіацій. Система формує достовірні статистичні показники частоти нахмурення, а повністю локальна обробка зображень забезпечує конфіденційність даних користувача без передачі їх у хмарні сервіси.

Висновок. Розроблено та досліджено настільну систему, що застосовує методи комп'ютерного зору для детекції мимічних напружень у реальному часі. Отримані результати підтверджують ефективність моделей комп'ютерного зору, а зокрема об'єктної детекції на основі зображень для персональних систем моніторингу та їхню перспективність у формуванні здорових поведінкових звичок. Подальший розвиток системи може включати

розширення набору детектованих мимічних патернів та рухів тіла як наприклад підняття плечей.

Література

1. Zhou L., Shao X., Mao Q. A survey of micro-expression recognition / L. Zhou, X. Shao, Q. Mao // *Image and Vision Computing*. – 2021. Режим доступу: <https://tinyurl.com/46m7muv>.
2. Nath S., Gowrishankar, Veena A., Deepa H. A Deep Learning Based Detection of Wrinkles on Skin / S. Nath, Gowrishankar, A. Veena, H. Deepa Режим доступу: <https://tinyurl.com/dkk6tbbk>.
3. Su Z. Comparative Analysis of CNN-Based Object Detection Models: Faster R-CNN, SSD, and Режим доступу: <https://tinyurl.com/28svks64>.

[illegible]

СТАТИСТИКА:

...у цьому випадку не виключено, що в процесі розробки проекту будуть виявлені інші фактори, які можуть впливати на результати дослідження. Тому в подальшому планується провести додаткові дослідження з метою уточнення впливу цих факторів на результати дослідження.

И в околном иудее, иносрени ан эулер эшнн он амстос монир микт
ошжхотд нтндофе прокловотд, аравуэчнрок дсртоп хннелуэвнннн од ксэтуэпнелд

[illegible]

оних ат үннинертом мотэсэ хныгаланосорп кдэ анжарбоос мөөсэ ан шжэтэдэ юнтэс бо жотнавоор йншыгаладэ жоняс хныожидэвоп хнвоордэ иннагымфод ү атсінвнжэспердэ йхүр ат өіндэстэ хнннмим хннвоотжэтэдэ үүдөн йннэдрншорд нтэрондэ жом нмэтэсэ

viixuq et viinqetep xniirimim xniybotxetod, xrodan vinnexniuxoq nteronxax exom nmoteno
 .ixroqon xttaxnjin duxixnqan xk xkit

**АКТУАЛЬНІ ЗАДАЧІ
СУЧАСНИХ ТЕХНОЛОГІЙ**
Збірник
тез доповідей
**XIV Міжнародної науково-технічної
конференції молодих учених та студентів**
11-12 грудня 2025 року



**УКРАЇНА
ТЕРНОПІЛЬ – 2025**