

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Сумський державний університет**

**Факультет електроніки та інформаційних технологій**

**Кафедра прикладної математики та моделювання складних систем**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Ігор КОПЛИК

\_\_\_\_\_ 2023 р.

## **КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня магістра**

зі спеціальності «Наука про дані та моделювання складних систем» освітньо-наукової програми 113 «Прикладна математика» на тему: «Створення інтелектуальної системи визначення функціонального стану оператора»

Здобувача групи ПМм-21 Марченко Ігоря Олександровича

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

\_\_\_\_\_ Ігор МАРЧЕНКО

Керівник канд. фіз.-мат. наук, доцент Ігор КОПЛИК \_\_\_\_\_

# АНОТАЦІЯ

**Кваліфікаційна робота:** 165 с., 30 рис., 131 джерело.

**Мета роботи:** дослідити взаємозв'язок показників організму та функціонального стану оператора та побудувати інформаційну систему визначення функціонального стану оператора, що здатна до самонавчання.

**Об'єкт дослідження:** відеодані, отримані спостереженням за оператором під час виконання роботи.

**Предмет дослідження:** характеристики методів та алгоритмів, які дозволяють визначити функціональний стан оператора.

**Методи аналізу:** метод однокрокових аномалій, метод локального викиду, метод ізоляційного лісу.

В ході виконання роботи було проаналізовано існуючі технології та підходи до визначення функціонального стану оператора. Виконано огляд і детальний аналіз параметрів організму, що дозволяють визначити функціональний стан оператора. Дані параметри було розділено на групи і виділено лише ті, які можливо отримати шляхом візуального спостереження. Побудовано систему, що здатна до самонавчання та адаптації до персональних характеристик оператора. Дана система не є ресурсоемкою може базуватися на більшості існуючих платформ. Отримана інформаційна система є нечутливою до умов спостереження і не потребує специфічного обладнання. Побудова навчальної вибірки для функціонування системи виконується безпосереднім спостереженням за станом оператора протягом перших 4-х хвилин роботи. Визначення функціонального стану оператора відбувається шляхом виявлення аномалій вхідних.

**Ключові слова:** ФУНКЦІОНАЛЬНИЙ СТАН ОПЕРАТОРА, ПИЛЬНІСТЬ, БЛИМАННЯ, ВИЯВЛЕННЯ АНОМАЛІЙ, ISOLATION FOREST, ONE CLASS SVM, FACE DETECTION

# Зміст

Вступ	5
<b>1 Аналіз сучасних методів визначення функціонального стану оператора</b>	<b>8</b>
1.1 Медична характеристика параметрів організму під час сну . . . . .	8
1.2 Оцінка сонливості за допомогою тестів та шкал . . . . .	14
1.3 Огляд методів визначення функціонального стану оператора, що базуються на візуальній інформації . . . . .	15
1.4 Алгоритми, що базуються на поведінці транспортного засобу . . . . .	22
1.5 Методи, що базуються на даних електроенцефалографії . . . . .	24
1.6 Параметри тіла . . . . .	32
1.7 Порівняння алгоритмів . . . . .	36
<b>2 Дослідження залежності параметрів організму та функціонального стану оператора</b>	<b>39</b>
2.1 Опис навчальної вибірки . . . . .	39
2.2 Алгоритми і методи локалізації ключових точок обличчя . . . . .	40
2.3 Метрика Eye Aspect Ratio . . . . .	42
2.4 Попередня підготовка даних . . . . .	44
2.5 Дослідження залежності характеристик елементів обличчя та функціонального стану оператора . . . . .	45
<b>3 Реалізація методів визначення функціонального стану оператора</b>	<b>64</b>
3.1 Математична модель методів визначення функціонального стану оператора . . . . .	64
3.2 Оцінка достовірності розпізнавання класифікатору . . . . .	72
3.3 Виявлення блимання . . . . .	73
3.4 Формування набору доречних ознак . . . . .	108

<b>4 Апаратне та програмне забезпечення адаптивної системи визначення функціонального стану оператора</b>	<b>114</b>
4.1 Апаратна частина . . . . .	114
4.2 Структура програмного додатку . . . . .	114
4.3 Побудова навчальної вибірки . . . . .	116
4.4 Визначення функціонального стану оператора . . . . .	140
4.5 Результати фізичного моделювання . . . . .	143
<b>Висновки</b>	<b>146</b>
<b>Список використаних джерел</b>	<b>147</b>
<b>Додатки</b>	<b>163</b>
Приклад опису обличчя . . . . .	163

## Вступ

Стрімкий розвиток технологій створив умови до різкого збільшення кількості інформації які зберігаються в цифровому вигляді. Не стало виключенням і фото-матеріали. Можливість зробити цифрове фото з використанням мобільного телефону або звичайного фотоапарату окреслили проблему подальшої обробки цієї інформації. В першу чергу це проблема збереження та пошуку.

Суттєво ускладнилась проблема пошуку інформації враховуючи її формат зберігання тим, що існуючі моделі та методи які застосовувались по пошуку текстовою інформації стали некорисними, і це наприкінці 70х років призвело до появи такого наукового напрямку як комп'ютерний зір задачі яких розв'язуються і досі.

Як науковий напрям, комп'ютерний зір належить до теорії та технології створення штучних систем, які отримують інформацію у вигляді зображень. Відеодані можуть бути представлені у вигляді багатьох форм, таких як відеопослідовність, зображення з різних камер або тривимірними даними з медичного сканера.

З технологічного боку, комп'ютерний зір прагне застосувати теорії та моделі комп'ютерного зору до створення систем комп'ютерного зору. Прикладами таких систем можуть бути: системи керування процесами (промислові роботи, автономні транспортні засоби); системи відеоспостереження; системи організації інформації (наприклад, для індексації баз даних зображень); системи моделювання об'єктів або оточуючого середовища (аналіз медичних зображень, топографічне моделювання); системи взаємодії (наприклад, пристрої введення для систем людино-машинної взаємодії).

Як підзадачі задач комп'ютерного зору включають відтворення дій, виявлення подій, стеження, розпізнавання образів, відновлення зображень, тощо.

Класична задача в комп'ютерному зорі, обробці зображень і машинному зорі — це визначення того, чи містять відеодані деякий характерний об'єкт, особливість чи активність. Ця задача може бути достовірно і легко вирішена людиною, але досі не вирішена задовільно в комп'ютерному зорі в загальному випадку: випадкові

об'єкти у випадкових ситуаціях.

В літературі описане різноманіття проблем розпізнавання:

Розпізнавання: один чи декілька попередньо заданих чи вивчених об'єктів або класів об'єктів можуть бути розпізнані, зазвичай разом з їх двовимірним положенням на зображенні чи тривимірним положенням в сцені.

Ідентифікація: розпізнається індивідуальний екземпляр об'єкта. Приклади: ідентифікація визначеного людського обличчя чи відбитка пальців чи автомобіля.

Виявлення: відеодані перевіряються на наявність визначеної умови. Наприклад, виявлення можливих неправильних клітин чи тканин в медичних зображеннях. Виявлення, що базується на відносно простих і швидких обчисленнях, іноді використовується для знаходження невеликих ділянок в зображенні, що аналізується, які потім аналізуються за допомогою заходів, що потребують більше ресурсів, для отримання правильної інтерпретації.

Враховуючи зростаючу кількість задач пов'язаних з проблемами комп'ютерного зору, актуальність робіт в цій галузі є дуже високою.

Тому головною метою роботи є розробка системи виявлення засинання оператора, що базується на аналізі поведінкових показників та активності, з метою вчасного виявлення та попередження стану засинання для підвищення продуктивності та зменшення ризику виникнення надзвичайної ситуації.

Для вирішення завдання необхідно:

1. дослідити взаємозв'язок показників організму та функціонального стану оператора;
2. розбити показники організму на групи та виділити такі, що можливо отримати використовуючи візуальну інформацію про оператора;
3. створити пристрій для отримання та обробки візуальної інформації;
4. спроектувати та реалізувати інформаційну систему побудови навчальної вибірки на основі динамічних характеристик елементів обличчя;

5. побудувати інформаційну систему визначення функціонального стану оператора.

# 1 Аналіз сучасних методів визначення функціонального стану оператора

## 1.1 Медична характеристика параметрів організму під час сну

Перехід організму з бадьорого стану до сну ініціюється супрахіазматичним ядром, що керує 24-годинним циклом організму за допомогою інформації про кількість світла, отриманої від очей[1]. Завдяки цьому, досягається синхронізація добових ритмів організму з циклом день-ніч. Окрім регуляції добових ритмів бадьорості, до функцій ядра відноситься терморегуляція та регуляція секреції гормонів. При відсутності інформації від очей, підтримання циркадних ритмів виконується за допомогою циклічності функціонування певних груп генів [2].

Залежно від значень основних параметрів організму розрізняють дві фази сну: повільний та швидкий сон. Головною відмінністю між даними фазами є вища активність процесів, які відбуваються в організмі під час стадії швидкого сну. Значення параметрів організму під час швидкого сну близькі до показників, отриманих для людини в бадьорому стані. Окрім того, саме для фази швидкого сну характерні сновидіння. Результати досліджень, наведені в [3] свідчать про те, що під час фази повільного сну відбувається відновлення енергії, що витрачена в фізичному аспекті, а під час швидкого - структуризація та опрацювання даних, отриманих протягом дня.

Загалом організм протягом сну проходить 5 стадій[4], деякі з яких періодично повторюються:

1. **Дрімота.** На даній стадії відбувається перехід від бадьорого стану до сну. Для даної стадії характерні повільні рухи очима та зниження активності організму в цілому, а саме - рухів, частоти дихання, пульсу, сповільнення обміну речовин та зниження температури тіла. На електроенцефалограммі



зменшується альфа-ритм та з'являється повільний тета-ритм низької амплітуди.

2. **Легкий сон.** На даному етапу припиняються рухи очима та температура тіла починає знижуватися. Дана стадія є проміжною між бадьорим станом та сном. Пробудження з даної стадії не викликає жодних проблем.
3. **Помірно глибокий сон.** Під час даної фази продовжується сповільнення всіх процесів та секреція гормону росту. Таким чином відбувається відновлення енерговитрат організму. Під час даної фази можливі міоклонії (незначні рухи кінцівками).
4. **Глибокий сон.** Під час даної стадії всі процеси в організмі максимально сповільнені, м'язи максимально розслаблені, частота серцевих скорочень мінімальна, а людина найменш чутлива до зовнішніх подразників. Для даної стадії характерна наявність дельта-хвиль на електроенцефалограммі, прояви парасомнії та нейтральні сновидіння, які не запам'ятовуються. При пробудженні людини, що знаходиться на даній стадії сну характерна дезорієнтація, що триває протягом декількох хвилин.
5. **Швидкий сон.** При переході до даної фази спостерігаються зовнішні показники такі, як: рухи очними яблуками, збільшення інтенсивності дихання, рухи м'язів обличчя та активізація серцево-судинної діяльності. Показники життєдіяльності організму під час даної фази не є стабільними: артеріальний тиск змінюється, частота серцевих скорочень та електрична активність головного мозку є хаотичними величинами. Для даної фази характерні сновидіння, які можливо запам'ятати та відтворити. Пробудження з даної стадії не викликає жодних проблем з бадьорістю.

Дані стадії повторюються протягом всієї довжини сна від 4 до 6 разів, залежно від загальної тривалості сну. Тривалість кожної стадії може незначним чином ко-

ливатись. Окрім того, послідовність стадій не завжди зберігається. Так, зазвичай після фази швидкого сну організм, зазвичай, переходить до легкого сну.

Всі параметри організму розбиті на наступні групи, залежно від способу їх вимірювання:

- **Актиграфія.** Рухи людини при бадьорому стані є більш активними, що можливо зафіксувати за допомогою актиграфа. Головним недоліком даного підходу є неможливість розділити фазу безпосередньо перед сном, під час якої людина знаходиться у відносно нерухомому стані та саму фазу сну.[5] До параметрів, що характеризують початок сну є зменшення частоти блимання[6], зменшення активності рухів зіниці[7]. Так, для блимання характерне збільшення часу закриття ока та знаходження його у закритому стані. Окрім того, суб'єктивними параметрами можуть бути: підвищена частота позіхань та нахил голови.
- **Електроокулографія.** Рух очей провокує зміну біопотенціалів, які можливо точно відстежити за допомогою електроокулографії[8, 9]. Для побудови електроокулограм використовуються електроди, які розміщені навколо очей. Найбільшу кореляцію з коливаннями значень потенціалів мають умови освітлення[10]. За допомогою даного підходу можливо з високою точністю розв'язати задачі оцінки часу фіксації погляду та частоту зміни точки фіксації[11].
- **Серцево-судинна активність.** Під час процесу засинання спостерігається незначне зменшення частоти серцевих скорочень на 10-15%, а в фазі сну - на 30-35% менше за аналогічний показник для бадьорого стану. [12, 13] В середньому під час сну відбувається близько 60 серцевих скорочень за хвилину. Під час сну існує три найбільш розповсюджені патерни зміни серцевого ритму: парабола, монотонне зниження частоти серцевих скорочень або стрімке зростання частоти з подальшим її зменшенням до початкового значення[14].

Для патерну, що формою нагадує параболу характерне зниження зниження кров'яного тиску та частоти серцевих скорочень протягом першої фази сну. Дані параметри досягають свого мінімального значення в середині сну, при цьому в даний момент рівень мелатоніну максимальний. Даний патерн справедливий лише для фази повільного сну. Монотонне спадання частоти серцевих скорочень свідчить про активний процес метаболізму. Причиною цього може бути пізній прийом їжі або фізична активність перед сном. Третій патерн характерний для випадку, коли людина засинає через певний час після прояву перших ознак сонливості.

- **Електроенцефалографія.** Одним з показником засинання є зниження тону м'язів. Експеримент, за допомогою якого встановлено залежність між тонусом м'язів під час засинання та ритмами ЕЕГ полягав у наступному: двома пальцями трималась котушка, котра випадала з рук, при значному зниженні тонусу м'язів. Перед падінням котушки, в середньому за 0.5 - 1.5 секунди на енцефалограмі значним чином зменшувалося значення альфа ритму, при цьому в деяких випадках затримка коливалась в діапазоні від 6.5 до 25 секунд.[15, 16]. Залежно від стадії сну спостерігається зміна взаємної активності альфа, бета, гамма, дельта та тета-ритмів.
- **Зміна дихання.** Під час сну значним чином зменшується інтенсивність дихання, а, як наслідок, і кількість поглинутого кисню та виробленого двоокису вуглецю. Внаслідок цього під час фази глибокого сну рівень двоокису вуглецю в крові вищий за норму. Для фази швидкого сну характерне незначне збільшення інтенсивності дихання, що в будь-якому разі нижче за показники для звичайного стану людини.[17] Під час другої фази спостерігається зменшення інтенсивності дихання на 13%, а під час третьої та четвертої - до 15%. Для стадії швидкого сну спостерігається вища інтенсивність дихання, аніж при всіх інших фазах.[18, 19].

- **Незначні рухи ока.** Дана характеристика актуальна для фази швидкого сну, що характеризується активною мозковою діяльністю, яка провокує швидкі рухи очних яблук та повік, амплітуда яких збільшується з часом.[20] Результати досліджень, що наведені в [21] свідчать про те, що дана поведінка спостерігається також у людей з вадами зору.
- **Опір шкіри.** Електрична активність шкіри обумовлена роботою потових залоз. Величина реакції потових залоз залежить від інтенсивності емоційних стимулів. Відмічено, що при засинанні опір шкіри в середньому зростає на 5 КОм, а при прокиданні - навпаки спадає на аналогічну величину.[22] Окрім того, залежно від "якості" сну опір шкіри змінюється на різну величину.[23]
- **Коливання температури тіла.** Під час сну температура тіла зменшується, залежно від особливостей організму, в середньому на 0.3 градуси Цельсія.[23] Дослідження, проведені[24] свідчать про те, що дана характеристика значним чином залежить від фізіологічних особливостей організму, адже при аналогічних умовах проведення експерименту даного результату не було досягнуто. Згідно з теоретичними припущеннями, зниження температури тіла дозволяє організму довше знаходитися в стадії глибокого сну, що дозволяє максимально відновити організм.
- **Рівень гормонів в крові.** Мелатонін - гормон, який провокує сонливість, зростання його кількості в крові є предвісником сну. Під час сну активізується секреція гормонів. Так, для мелатоніну під час сну його вироблення зростає в 30 разів, для соматропіну максимальна кількість гормону виробляється через 1-2 години після засинання, при недостатці сну вироблення лептину знижується на 20%. окрім того, а такі гормони як тестостерон, кортизол та грелін активно виробляються саме під час сну.[25]

Серед розглянутих параметрів, які проявляються при переході організму від бадьорого стану до сну лише деякі можливо оцінити без використання спеціального

обладнання, що передбачає фізичний контакт з людиною та використання спеціальних датчиків. Даний підхід не є практичним з точки зору ергономіки. Окрім того, вартість таких датчиків робить неможливим їх масового використання для пересічного користувача.

Залежно від способу оцінки значення параметру, множина ознак розбита на дві підмножини (табл.1).

Візуальне спостереження	Спеціалізоване обладнання
Частота блимання	Частота серцевих скорочень
Активність рухів зіниці	Зниження тону м'язів
Підвищена частота позіхань	Ритми ЕЕГ
Нахил голови	Інтенсивність дихання
	Електрична активність шкіри
	Температура тіла
	Рівень гормонів в крові
	Частота пульсу
	Зміна артеріального тиску

Табл. 1: Тип ознак, залежно від способу визначення їх значень.

Ознаки, для оцінки значення яких необхідно використання спеціальних датчиків є внутрішніми параметрами організму. Так, наприклад, для оцінки рівня гормонів необхідно проводити аналіз крові, для оцінки ритмів ЕЕГ, електричної активності шкіри, зміни артеріального тиску, інтенсивності дихання необхідно спеціальне обладнання та знаходження людини в стані спокою для запобігання пошкодження обладнання та втрати зв'язку з ним.

Всі параметри, значення яких можливо оцінити дистанційно залежать від обличчя людини та його частин. Для оцінки значень даної групи ознак та їх зміни у часі достатньо звичайної веб-камери. Використання інфрачервоного ліхтаря з довжиною хвилі близько 940 нм незначним чином спрощує процес локалізації по-

зиції зіниць в умовах недостатнього освітлення. При цьому необхідно врахувати потужність джерела випромінювання для запобігання травм ока. [26, 27, 28]

## 1.2 Оцінка сонливості за допомогою тестів та шкал

До найбільш розповсюджених шкал для оцінки сонливості відносяться: епвортська, стенфордська та каролінська шкали сонливості. З використанням даних шкал можливо самостійно оцінити свій рівень сонливості безпосередньо під час тесту або виявити розлади сну.

За допомогою епвортської шкали[29] оцінюється сонливість в різноманітних побутових ситуаціях, таких як перегляд телевізору, відпочинок після обіду, розмова, перегляд телевізору і т.д. Процес тестування полягає в самостійній оцінці респондентом ймовірності засинання в різноманітних ситуаціях балами від 0 до 3, де 0 - “не засну ніколи”, а 3 - “обов’язково засну”. Залежно від набраної кількості балів діагностується наявність або відсутність порушень сну. Так, при набраних 10-24 балах необхідна консультація лікаря.

Стенфордська шкала сонливості[30] призначена для оцінки сонливості безпосередньо під час тестування. Дана шкала складається з єдиного питання та семи варіантів відповіді, кожен з яких описує ступінь бадьорості в даний момент.

При суб’єктивній оцінці людиною рівня своєї бадьорості найбільш розповсюдженою технікою є Каролінська шкала сонливості.[31]. Дана шкала використовується для оцінки рівня сонливості у професійних водіїв, машиністів, пілотів та клінічних дослідженнях лікарських засобів. Серед головних недоліків даного підходу є неможливість аналізу великого обсягу показників, які не є очевидними, а набір ознак та їх значення значним чином залежать від особливостей організму. Окрім того, при аналізі бадьорості з використанням даного підходу знижується концентрація людини-оператора.

Для оцінки фізичної та ментальної втоми використовується список із 14 запитань, для кожного з яких передбачено 4 варіанта відповіді: “краще, ніж зазви-

чай”, “як зазвичай”, “гірше ніж зазвичай” та “набагато гірше, ніж зазвичай”[32]. Фінальна оцінка обчислюється аналогічно до методу, який використовується у загальному анкетуванні з питань здоров’я[33] або за допомогою шкали Лікерта[34]. За допомогою методу головних компонент з отриманої вибірки обирається комбінація ознак, що найбільш повно описують функціональний стан оператора.

Головним недоліком даної групи методів є суб’єктивність відповідей респондентів. Так, відповіді можуть бути перебільшені або применшені, що кардинально відобразиться на фінальному результаті.

### **1.3 Огляд методів визначення функціонального стану оператора, що базуються на візуальній інформації**

Найбільш практичною групою методів для визначення функціонального стану оператора(ФСО) є алгоритми, що базуються на параметрах, які можливо визначити візуальним спостереженням: частота блімання, інтенсивність зміни напрямку погляду, нахил голови, позіхання, рухи обличчя, зміна емоцій, тощо.

Одним з факторів, що може свідчити про зміну ФСО є зменшення інтенсивності зміни напрямку погляду. В роботі [35] для визначення напрямку погляду, простір перед оператором було розділено на 9 секторів, що дозволяє підвищити точність визначення напрямку погляду. Локалізація та відстеження обличчя у відеопотоці виконується за допомогою комбінації методу Віюлі-Джонса та алгоритму описаного в [36]. За допомогою алгоритму Віюлі-Джонса виконується обробка першого фрейму з метою початкової локалізації обличчя, а за допомогою алгоритму MOSSE відстежується його зміна у відеопотоці. Визначення напрямку погляду обчислюється виходячи зі зміщення зіниці відносно її нормального положення та області ока, в якому вона знаходиться. Визначальним фактором при обчисленні напрямку погляду є позиція зіниці в визначеному регіоні. Збільшення точності локалізації зіниці досягається шляхом використання інфрачервоного освітлення, аналогічно підходу, описаному в [37]. В якості класифікатора побудовано нейронну

мережу, структурно схожою на AlexNet[38] та містить 5 згорткових шарів і 1 прихований. Класифікатор з описаною структурою має достовірність розпізнавання 95%. Так як на обробку одного фрейму необхідно 25-28мс, то даний класифікатор можливо використовувати в режимі реального часу. Використання даної системи дозволяє попередити оператора про втрату пильності шляхом аналізу точки, на якій зосереджена увага. Аналогічно до сну з закритими очима можливий мікросон, під час якого очі наполовину відкриті або повністю закриті. Для даного стану характерна втрата концентрації, а зовнішньою ознакою є зміщення точки уваги на певний проміжок часу. При використанні описаної ідеї у комбінації з байесівською мережею в якості класифікатора, отримано достовірність розпізнавання близько 85%[39].

Чисельна характеристика, що найбільш точно описує поточний стан ока людини є видима площа ока (PERCLOS[40] - percentage of eyelid closure). Чисельний опис взаємного положення повік, яке описує ФСО дозволяє використовувати значення PERCLOS для діагностування засинання оператора. Для визначення ФСО побудовано класифікатор, з використанням системи опорних векторів[41], який в якості вхідних даних використовує характеристику PERCLOS. Зменшення розмірності даних виконується за допомогою методу головних компонент, що базується на розподілі Бернуллі[42]. З метою оцінки зміни значення PERCLOS у часі використовується адаптація експоненційного згладжування:

$$\left( \sum 0.8^{t-i} ecv_i \right) + 0.2 * ecv_t, \text{ де}$$

$t$  - номер поточного фрейму,  $i$  - номер попередніх фреймів. При визначенні значення  $ecv_t$  використовується наступний алгоритм:

$$\begin{cases} ecv_t = 0, CES \notin ES \vee CES = Closed \\ ecv_t, CES = Open \end{cases}$$

Еквівалентність відсутності ока та його ФСО, *Closed* не матиме значного впливу на достовірність розпізнавання завдяки тому, що локалізація очей виконується на



фронтальному зображенні обличчя людини, а кількість фреймів, для яких неможливо локалізувати очі є незначною. При використанні вхідних даних зображень без попередньої обробки достовірність розпізнавання складає 73% для відкритого ока та 80% для закритого. При обробці зображення за допомогою фільтру Кенні або оператора Собеля[43], залежно від виду реалізації методу головних компонент, достовірність розпізнавання для відкритого ока коливається в діапазоні від 87% до 93%, а для закритого - від 10% до 60%. За допомогою критерію хі-квадрат була знайдена кореляція між параметром PERCLOS та станом оператора. За нульову гіпотезу було прийнято, що дані дві величини незалежні між собою, в результаті проведених обчислень дана гіпотеза була відкинута зі значимістю близькою до 0%. Побудована система представляє собою метод опорних векторів, що оперує даними, обробленими за допомогою бінарного методу головних компонент, застосованого до зображень, на яких за допомогою оператора Собеля знайдено всі границі об'єктів. Достовірність розпізнавання отриманої системи становить 83.64%.

Окремим підходом до визначення ФСО є використання класифікаторів, які побудовані, на основі інформації про емоції на обличчі. Ключовими характеристиками, які використовуються при виявленні дрімоти у оператора є рух підборіддя, рух носогубової складки та зміна положення брів[44, 45]. Для представлення міміки обличчя використовується система кодування рухів обличчя FACS (facial action coding system)[46]. За допомогою даної системи представляється 46 різноманітних елементів обличчя (рухових елементів) та їх рухів (рухові дескриптори). Обчислення оцінок інтенсивності задіяності кожного з рухових елементів виконується за допомогою алгоритму, що базується на застосування фільтра Габора[47] та методу опорних векторів, для якого виконано навчання на всіх видах рухових дескрипторів. Для дослідження кореляції значення рухового дескриптора зі станом оператора побудовано мультимінальну логістичну регресію і отримано відповідні значення ROC AUC. Рухові дескриптори, зростання значень яких має найбільший вплив на достовірність розпізнавання дрімоти є: блимання (ROC AUC

= 0.94), підйом верхньої брови (ROC AUC = 0.81), підйом підборіддя (ROC AUC = 0.79), насуплення носу (ROC AUC = 0.78). Зі зменшенням значення наступних рухових дескрипторів зростає ймовірність настання стану дрімоти: посмішка (ROC AUC = 0.87), закриття повіки (ROC AUC = 0.86), звуження ноздрів (ROC AUC = 0.79), опускання щелепи (ROC AUC = 0.77). Зазначено, що при стомленні оператора та наближенні дрімоти зростає інтенсивність рухів голови, а безпосередньо в стані дрімоти в тестовій вибірці спостерігаються періодичні спроби утримання очей відкритими. Опускання щелепи, що може свідчити про позіхання, має дуже низьку частоту прояву за інтервал часу в 60 секунд, що свідчить про нецільність використання цього параметру при діагностуванні засинання. Шляхом комбінації всіх ознак та тестування методом крос-валідації з відкиданням одного елемента визначено, що оптимальним набором ознак для діагностування ФСО є: підйом верхньої брови, опускання кутів губ, опускання щелепи, блимання та показ язика. Для отриманого набору параметрів значення ROC AUC становить 0.9792. Ключовим фактором при визначенні стану оператора є часовий інтервал, для якого будуються спостереження. Так, максимальне значення ROC AUC = 0.99 досягається для інтервалу довжиною в 30 секунд.

Висока достовірність розпізнання ФСО досягається шляхом побудови нейронної мережі[48], що є ансамблем наступних нейронних мереж: AlexNet[38] - для пошуку ознак, що свідчать про дрімоту, VGG-FaceNet[49] - для пошуку ознак, що залежать від статі, раси, зачіски та аксесуарів, FlowImageNet[50] - для ознак, що залежать від поведінки та описуються рухами та позицією голови. Реалізовано дві різноманітні стратегії об'єднання результатів роботи нейронних мереж: середнє значення результатів всіх мереж та нейронна мережа, результат роботи якої отримано з шару, який отримано шляхом об'єднання нейронів 7-го прихованого шару. В результаті достовірність розпізнавання кожної окремої нейронної мережі не перевищує 70%, при використанні середнього значення досягнуто значення достовірності розпізнавання в 73.06%, а при доповненні нейронної мережі додатковим шаром даний показник дорівнює 70.81%.

Шляхом попередньої обробки зображення та застосування до нього штучної нейронної мережі не вдалося досягти достатньо високої достовірності розпізнавання[51] незважаючи на зменшення кількості похибок завдяки обробці зображення. Етап локалізації очей розділяється на 2 кроки: на першому етапі на зображенні відбувається локалізація обличчя, а на другому в знайденому регіоні відбувається безпосередня локалізація очей. При цьому, для підвищення достовірності локалізації відбувається попереднє згладжування зображення за допомогою фільтру Калмана[52]. Наступним кроком, за допомогою морфологічних перетворень регіонів з очима зменшується кількість перепадів яскравості. В подальшому, з метою підвищення точності локалізації зіниці виконується трансформація зображення, залежно від освітленості. Алгоритм пошуку зіниці полягає у проєкції значень пікселів на вертикальну та горизонтальну осі для визначення математичного очікування положення центра зіниці. Ступінь закритості ока обчислюється за допомогою стандартного відхилення значень, що розміщені на вертикальній осі. Обчислення значення характеристики PERCLOS можливе лише за умови попередньою калібрації системи, що полягає у побудові вибірки для нормального стану оператора протягом 10 секунд. Значення PERCLOS визначається як відношення значення поточної висоти зіниці та еталонної. При повороті голови неможливо точно обчислити дане значення, тому стан визначається як близький до дрімоти. Для побудови класифікатора було застосовано штучну нейронну мережу, завдяки чому вдалося досягти достовірності розпізнавання в 86.07%.

Іншим підходом до передбачення засинання оператора-водія є використання комбінації двох нейронних мереж та великої множини параметрів, які складаються з трьох груп: поведінка автомобіля(кут повороту керма, ступінь взаємодії з педаллю акселератора, зміщення відносно центру смуги руху, швидкість та кількість перетинів ліній розмітки), психологічні показники(частота серцевих скорочень, варіабельність серцевого ритму, частота дихання та його варіабельність), поведінкова(тривалість блимання та його частота, PERCLOS, рухи голови та частота саккадних рухів)[53]. В якості класифікатора побудована штучна нейронна ме-

режа з двома прихованими шарами. Аналогічна нейронна мережа застосовується для передбачення моменту часу, коли оператор досягне стану дрімоти. Система навчається протягом перших 5 хвилин роботи з нею. Даний стан визначається як еталонний. Еталонною вибіркою є вибірка, для якої середнє значення достовірності класифікації не нижче 95%. Для побудови еталонної вибірки з використанням лише одного параметру необхідний час від 50 до 60 хвилин, а комбінація мінімум з двох різноманітних груп параметрів скорочують даний час до 4-6 хвилин. В результаті тестування отримано, що оптимальний набір параметрів складається зі степені закритості ока, напрямку погляду та рух голови. Побудована система розпізнає стан дрімоти з достовірністю 94.69%.

Перехід з кольорової схеми RGB до HSV дозволяє зменшити асимптотику алгоритму локалізації регіонів з очима без суттєвого зниження достовірності розпізнавання. Аналогічно до описаних вище методів початкові фрейми використовуються для знаходження положення обличчя на зображенні. Якщо початкове положення знайдене, то для кожного наступного фрейму виконується відстеження зміни його позиції. Для пошуку обличчя на зображенні використовується алгоритм, що базується на кольоровій характеристиці шкіри. З метою спрощення маніпуляцій виконується перехід до кольорової моделі HSV, де кожен компонент представляє собою відтінок, насиченість та яскравість, відповідно. Емпіричним шляхом встановлено, що для шкіри компонента H має знаходитися в діапазоні  $[10^\circ; 45^\circ]$ . Попередня підготовка зображення для пошуку очей полягає у обробці за допомогою фільтру Собеля для виділення границь об'єктів. Шляхом проекції значень яскравості на вертикальну вісь обчислюється вертикальне положення ока. Для розділення компонент, що описують ліве і праве око об'єднуються близькі компоненти, близької яскравості. Визначення поточного стану ока виконується шляхом аналізу кольору пікселів в регіоні, що визначений як око. Визначення поточного стану здійснюється шляхом самодіагностики. Даний результат порівнюється з результатом класифікації, що базується на степені відкритості ока та частоті блимання. Отримана система дозволяє досягти достовірності класифікації в 88.9%.

Для оцінки залежності достовірності розпізнавання від індивідуальних особливостей параметрів очей оператора побудовано множину класифікаторів: для загального випадку, та з урахуванням анатомічних особливостей[55]. Для формування навчальної вибірки ФСО було оцінено трьома незалежними експертами за Каролінською шкалою сонливості[31]. Поточний стан оператора характеризується рухами ока, частотою блимання, PERCLOS, напрямком погляду та часом його фіксації. В якості класифікатора використовується метод опорних векторів з радіально-базисною функцією в якості ядра. З метою порівняння для кожного оператора було побудовано особисту модель шляхом підбору параметрів функціонування алгоритму класифікації. Окрім того, одна з моделей є загальним випадком, що не пов'язана з жодним з операторів. В результаті, для специфічних моделей середня достовірність розпізнавання становить 85.41%, а для загальної 72.23%. Таким чином, при побудові моделі для класифікації вагомий внесок у значення достовірності розпізнавання вносять індивідуальні особливості оператора.

Найбільш точним методом для визначення поточного стану ока є електроокулографія. Головним недоліком даної технології є необхідність розміщення електродів на обличчі. Достовірність розпізнавання рухів елементів обличчя, що досягається використовуючи електроокулографію можливо отримати, шляхом аналізу відеопотоку, отриманого зі високошвидкісної камери. Для порівняння похибки цих двох підходів проведено ряд дослідів[56]. За допомогою електроокулографії можливо отримати дані про наступні характеристики ока: тривалість блимання, час закриття та відкриття ока, амплітуда, швидкість закриття та відкриття ока, максимальна швидкість при закритті ока, PERCLOS, частота блимання. В зазначеній роботі для пошуку очей на зображенні використано підхід, що базується на комбінації градієнту та проекції. За допомогою горизонтального градієнту локалізуються та уточнюються контури, а за допомогою вертикальної та горизонтальної проекції отримано точне положення ока. Описаний метод має достовірність локалізації ока близьку до 97%. В якості еталонних значень обрано результати розпізнавання за допомогою електроокулографії. Локалізація зіниці в

межах ока виконується за допомогою обробки зображення з використанням OPL та IPL фільтрів. Достовірність розпізнавання класифікатору, що в якості вхідних даних використовує дані, отримані з відеопотоку коливається від 0.67 до 0.78, залежно від кількості кадрів за секунду (від 30 до 200). Аналогічний показник для частоти блимань знаходиться близько 0.99, незалежно від частоти відео, а для PERCLOS даних показник коливається від 0.78 до 0.93. Таким чином, при діагностуванні ФСО, використовуючи відеопотік очевидне незначне зменшення достовірності розпізнавання через недостатню точність вхідних даних. При цьому, частота відеопотоку повинна бути не нижча за 150 фреймів за секунду.

## 1.4 Алгоритми, що базуються на поведінці транспортного засобу

При визначенні стану оператора, якщо мова йде про автомобілі та водіїв, то в якості джерела даних може виступати автомобіль та взаємодія водія та транспортного засобу. В якості головних параметрів виступають: активність виконання маневрів, частота зміни смуги рухи, інтенсивність впливу на органи керування, частота самовільних маневрів автомобілем.

Найпоширеніші класифікатори, які в якості вхідних даних використовують сигнали про поворот керма. Множина факторів, що впливають на активність повороту керма доволі різноманітна: зміна напрямку погляду, відволікання, дорожні та погодні умови, швидкість. Швидкість повороту керма, що перевищує 150град/сек може свідчити про засинання водія[57]. Дана система активно використовується на автомобілях марки Мерседес-Бенц[58]. В якості даних використовується патерн кермування. Так, доведено, що патерн для кермування в стані дрімоти значно відрізняється від кермування в екстремальній ситуації. Залежно від виду ситуації патерн поведінки водія значно відрізняється. Так, при виникненні бокового вітру оператор повертається до свого нормального режиму лише за 5 секунд, а при уникненні перепоп на дорозі - за 3 секунди. Окрім того, дана система на-

вчається протягом всього свого функціонування.

Одним з підходом до опису даних про залежність повороту керма від стану водія є використання теорії хаосу[59]. Для навчання класифікатора дані розбиваються на інтервали. Найбільш оптимальним та репрезентативним інтервалом визначено вікно в 1 секунду. Після цього отриманий вектор значень перетворюється за допомогою швидкого перетворення Фур'є в діаграму густини частот. Наступним кроком є застосування вейвлет перетворення до отриманих даних. В результаті проведених перетворень визначено, що траєкторії, які описують рухи керма повторюють певні орбіти - аттрактор.

Шляхом побудови множини ознак, джерелом яких є одночасно аналіз характеру кермування, положення транспортного засобу на смузі руху та навколишньої ситуації отримано класифікатор, що аналізує поточний стан оператора, враховуючи ознаки різної природи[60]. Всі ознаки розділено на два класи: причинні та апріорні. Причинні ознаки - ознаки, до яких призводять дії оператора в стані втоми або дрімоти. Апріорні ознаки свідчать про ймовірність для оператора перейти в стан дрімоти через зовнішні подразники. Причинні ознаки є найбільш інформативними для визначення поточного стану оператора. Окрім того, залежно від частоти та природи ознак їх можливо розділити на неперервні та стихійні. Поява окремої стихійної ознаки не може свідчити про зміну стану оператора, в той же час показовим є частота таких подій. Для керма показовою є частота зміни напрямку. За зміну напрямку приймається будь-який поворот керма більш ніж на 0.5 градуса. При цьому враховуються корекції курсу. Основна ідея полягає в тому, що під час кермування водій здійснює мінімальні повороти керма для утримання транспортного засобу на заданій траєкторії. Залежно від ступеню бадьорості швидкість повороту керма змінюється. Так, для бадьорого стану рухи є більш різкими, в той час як для стомленого водія рухи є більш повільними. Зміщення транспортного засобу в праву частину смуги руху може свідчити про стан дрімоти у водія. Окрім того, фактором, що свідчить про зміну стану оператора є збільшення частоти перетину або наближення до ліній дорожньої розмітки.

Серед факторів, що характеризують стан оператора є час реакції на зміну положення та швидкості транспортного засобу внаслідок зміни дорожніх умов. Для оцінки поведінки оператора в нормальному стані аналізуються його дії протягом перших хвилин з початку взаємодії з системою. Для оцінки частоти подій використовується експоненційне зважене рухоме середнє, головною перевагою якого є зберігання лиш одного значення замість вектору, що описує інтервал. Для підвищення достовірності розпізнавання використовується адаптивна зміна розміру вікна сканування для врахування зміни умов. Стан водія класифікується як один із трьох: бадьорість, дрімота та проміжний стан. При використанні в якості класифікатора штучної нейронної мережі з трьома прихованими шарами достовірність класифікації, залежно від стану коливається в діапазоні від 54.6% до 89.9%. Так, для бадьорого стану достовірність класифікації становить 76%, для проміжного 89.9%, а для дрімоти 54.6%.

Окрім того, важливим типом параметрів, що застосовується на залізничному та автотранспорті є системи самодіагностики. Для підтвердження власного бадьорого стану в даному типі систем виконується взаємодія з пристроєм для введення даних (найчастіше - кнопка). Для даного типу систем можливо досягти будь-скільки високої достовірності розпізнавання за рахунок постійної взаємодії в системі. Головною перевагою даного роду методів є додаткова фізична активність, завдяки якій можливо відновити бадьорий стан оператора. Ступінь відновлення залежить від рівня фізичної активності.

## **1.5 Методи, що базуються на даних електроенцефалографії**

Електроенцефалографія(ЕЕГ) є методом дослідження функціонального стану мозку шляхом реєстрації його біологічної активності. Головною характеристикою ЕЕГ сигналу є його частота. Сигнал ЕЕГ, залежно від частоти, розділяється на діапазони, що називаються ритмами:



- Дельта (0.1 - 4 Hz). пов'язаний з відновлювальними процесами. Найчастіше спостерігається під час сну.
- Тета (4 - 8 Hz). Характеризує перехід від бадьорого стану до сну. Найбільша амплітуда спостерігається при емоційних переживаннях. Окрім того, значна амплітуда може свідчити про стан стомлення та сонливості.
- Альфа (8 - 13 Hz). Дані хвилі характеризують розслаблений стан бадьорості та спокою. Під час підвищення функціональної активності головного мозку амплітуда альфа хвиль значно зменшується.
- Бета (13 - 36 Hz). Пов'язаний з процесами, які потребують концентрації та когнітивними процесами. Під час рухової діяльності спостерігається зменшення його амплітуди.
- Гамма (36 - 44 Hz). Найбільш виражений при максимальній концентрації.

Зміни значень певних груп ритмів можуть свідчити про стомленість або засинання.[61, 62, 63, 64]

Стан оператора, а, як наслідок, і значення ЕЕГ в цілому залежить від циркадних ритмів та гомеостазу[65]. Циркадні ритми - циклічні коливання інтенсивності біологічних процесів, які пов'язані зі зміною дня та ночі. Гомеостаз - внутрішні процеси, які спрямовані на підтримання рівноваги в організмі. Залежно від добових ритмів в певні проміжки часу спостерігається пік сонливості. При невідповідності годин сну до звичного ритму спостерігається його порушення, що провокує проблеми зі сном. Так як, циркадні ритми та гомеостаз приймають безпосередню участь у регуляції процесів функціонування організму, то вони значним чином впливають на ритми ЕЕГ. Зміни ритмів ЕЕГ дозволяють точно оцінити поточний стан організму. Так, поява низькочастотних дельта ритмів свідчить про стадію дрімоти, а поява тета ритмів свідчить про початкову стадію сну.[66]

Для запису ЕЕГ з різних областей мозку використовується спеціальне обладнання, основною частиною якого є електроди, які можуть бути розміщені в спе-

ціальному шоломі або знаходиться у вільному стані. Для розміщення електродів на голові використовується міжнародна система "10-20"[67]. При цьому, залежно від необхідної повноти отриманих даних на практиці використовується 4, 8 або 32 канали. При встановленні електродів перевага віддається мінімальній їх кількості, яка максимально повно описує ритми ЕЕГ.[68, 69, 70, 71, 72, 73, 74, 75]

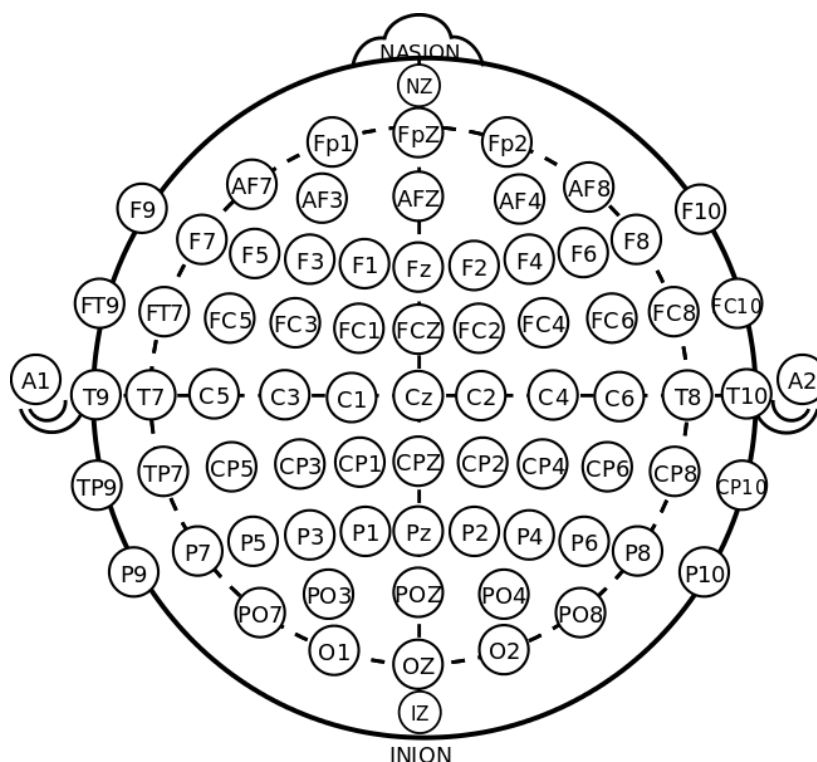


Рис. 1: Міжнародна система розміщення електродів "10-20"

Загальна структура більшості методів визначення поточного стану оператора, що базуються на значеннях ритмів ЕЕГ наступна:

- Отримання даних ЕЕГ за допомогою сенсорів.
- Розбиття даних на інтервали, що не перетинаються у часі.
- Побудова гістограм розподілу значень за частотами за допомогою алгоритму швидкого перетворення Фур'є.
- Визначення поточного стану оператора за допомогою методу класифікації.

При цьому, залежно від реалізації деякі етапи можуть бути пропущені. Окрім того, при постановці задачі прогнозування додатково існує етап обробки отриманих результатів за допомогою регресійного аналізу.

Одним з методів визначення поточного стану оператора є обробка даних ЕЕГ за допомогою використання аналізу кількісної регресії (RQA) [76], який призначено для кількісної оцінки процесів, що динамічно змінюються. В основі аналізу кількісної регресії лежать рекурентні графіки [77], що описують появу стану  $x_i$  в різні моменти часу. За допомогою RQA можливо отримати наступні характеристики, що характеризують сигнал:

- рівень рекурентності ( $RR$ ), що описує щільність точок на графіку рекурентності та характеризує ймовірність появи певного стану;
- детермінованість ( $DET$ ) - передбачуваність динамічної системи;
- ламінарність ( $LAM$ ) - кількість ламінарних фаз - станів, що не перетинаються;
- відносна кореляція змінної ( $RATIO$ ), що використовується для дослідження евристичних переходів у динамічних системах;
- максимальна довжина лінії у діагональному напрямку ( $D_{max}$ ), яка корелює з показником Ляпунова, що характеризує хаотичність процесів, які описує система;
- ентропія Шеннона частоти розподілу довжини діагональних ліній ( $ENT$ ), яка відображає складність детермінованої структури системи. Більше значення свідчить про більшу складність системи та може описувати некорельований шум;
- тренд ( $TND$ ), що описує стаціонарність або нестаціонарність процесу.

Комбінація цих та інших параметрів, отриманих за допомогою RQA використовуються при побудові різноманітних класифікаторів, що базуються на метод

опорних векторів(SVM)[78], наївного баєсівського класифікатора (НБК)[79], методу  $k$ -найближчих сусідів[80] та лінійного дискримінантного аналізу Фішера[81]. Для кожної комбінації отримані результати достовірності розпізнавання поточного стану та передбачення дрімоти.[68] До комбінацій, для яких отримана достовірність вища за 90% відносяться НБК, SVM та класифікатор Фішера, які побудовано з використанням характеристики детермінованості(*DET*) з точки *F8*. Найвища достовірність 93.2% отримана за допомогою наївного байєсівського класифікатора, побудованого з використанням даних, які є результатом роботи алгоритму лінійного дискримінантного аналізу з 12 характеристиками, отриманих за допомогою RQA.

Для побудови моделі, здатної прогнозувати перехід оператора в стан дрімоти використовується поєднання методу опорних векторів та його ймовірнісна модель[82]. За допомогою SVM для кожного моменту часу було визначено стан людини, а за допомогою ймовірнісної моделі на основі поточного стану отримано постеріорну ймовірність наступного стану. Джерелом даних в даному експерименті виступали точки *O1* та *O2*, як такі, що найбільш повно описують процеси, які відбуваються в корі головного мозку. В результаті використання даної комбінації методу досягнуто достовірність від 83.78% до 91.92%, залежно від рівня попередження про втрату концентрації.[70]

Одним з підходів до аналізу ритмів ЕЕГ є розбиття їх на інтервали і аналіз кожного з них окремо.[69] За допомогою частотно-часового аналізу з використанням перетворення Фур'є[83] отримано розподіл значень, залежно від частоти сигналу. Для визначення поточного стану оператора використовується лише інтервал, що складається з 512 значень та представляє собою проміжок часу в 2 секунди. Для кожного з таких інтервалів застосовано метод Уелша[84] для визначення потужності спектральної густини. Для цього, інтервал від 1 до 30 герц було розбито на 30 рівних сегментів на кожному з яких отримана величина потужної спектральної густини в логарифмічній шкалі. Для побудови регресійної моделі було застосовано регресійний метод опорних векторів з використанням отрима-

них на попередньому кроці даних. Даний метод є модифікацією методу опорних векторів для розв'язання задач прогнозування. Незважаючи на значну кількість операцій з обробки та аналізу даних, побудована система може використовуватися в режимі реального часу, а в якості пристрою для обробки даних виступає мобільний телефон. В результаті отримано середньоквадратичне значення похибки ( $0.115 \pm 0.053$ ).

Для отримання алгоритму визначення поточного стану оператору використовуються алгоритми, що базуються на словнику ознак[71], який побудовано за допомогою алгоритму K-SVD[85], що є модифікацією кластеризації методом K-середніх. Для отримання значень ЕЕГ використовують точки O1 та O2. За допомогою дискретного вейвлет-перетворення [86] у вхідних даних було значним чином зменшено рівень шумів. Обчислювальна складність значним чином залежить від частоти дискретизації. Зниження частоти дискретизації до 128Гц не призводить до значного зменшення достовірності розпізнавання але значним чином збільшується швидкість роботи алгоритму. Після цього виконується перетворення сигналів ЕЕГ у модель розрідженого представлення[87]. Суть цієї моделі полягає в тому, що початкове значення представляється у вигляді лінійної комбінації елементів словника. Для отримання словника мінімального та достатнього для класифікації розміру використано алгоритм K-SVD. Для класифікації поточного стану використовується словник, який представляє собою конкатенацію словників, які описують дрімоту та нормальний стан. Клас, який описує поточний стан оператора визначається як різниця між початковим значенням вхідного сигналу, та його представленням з використанням словника ознак. Завдяки даному підходу вдалося досягти достовірності близької до 93%.

Окрім того, відомий підхід до визначення поточного стану оператора без використання машинного навчання, а лише за допомогою статистичних методів.[72] Аналогічно до зазначених методів, за допомогою швидкого перетворення Фур'є виконана попередня обробка сигналу, після чого отримані значення альфа та тета ритмів. Модель для класифікації базується на багатовимірному нормальному роз-

поділі  $N(\mu, \Sigma^2)$ , де  $\mu$  - вектор середніх значень, а  $\Sigma$  - коваріаційна матриця. Для визначення значень  $\mu$  та  $\Sigma$  використовується метод максимальної правдоподібності. Для побудови моделі використовуються дані за 3 хвилини. Якщо побудована модель не відповідає нормальному розподілу, то модель будується з використанням даних за наступні 3 хвилини. На етапі класифікації для альфа та тета ритмів обчислюється відстань Махаланобіуса (MDA та MDT відповідно). Після цього їх лінійна комбінація  $MDC = \alpha * MDA + (1 - \alpha) * MDT$ ,  $0 \leq \alpha \leq 1$  є індикатором бадьорості. При перевищенні значення MDC порогового значення, стан оператора класифікується як дрімота. З використанням даного методу досягнуто чутливості в 88.7%.

Також для класифікації використовуються рекурентні нейронні мережі з гнучкою логікою, що самонавчаються[73]. Попередня обробка сигналу виконується за допомогою швидкого перетворення Фур'є. Для отримання даних, Аналогічно попереднім методам, використовуються точки  $O_1, O_2$  та  $O_z$ . За допомогою рекурентної нейронної мережі враховуються не тільки поточні дані, а і попередні значення параметру, що дозволяє простежити динаміку зміни параметру у часі. Побудована нейронна мережа складається з 6 шарів:

1. Вхідний шар. На даному шарі не виконуються жодні обчислення, а шар служить лише для передачі вхідних даних на наступний шар.
2. Кожний нейрон даного шару використовує функцію належності Гаусса для встановлення лінгвістичних позначок на вхідних даних з попереднього шару. Даний шар служить для підготовки даних до їх обробки за допомогою нечіткої логіки.
3. Кожний нейрон даного шару відповідає одному нечіткому правилу. В кожному нейроні обчислюється одновимірний ступінь належності, що базується на відповідному правилі.
4. На даному шарі для кожного нейрону характерна дуга, що дозволяє ура-

хувати вихід попереднього шару та лінійну комбінацію виходів поточного нейрону.

5. Кожний нейрон попереднього шару має відповідну вершину на поточному, зв'язок між якими досягається за допомогою ребра вагою  $a_{i,j}$ . Нейрони поточного шару формують лінійну комбінацію вхідних змінних.
6. Вихідний шар. Нейрони останнього шару застосовують всі правила, що сформовані на четвертому та п'ятому шарах. Результатом роботи є клас, до якого віднесено вхідні дані.

З використанням нечіткої логіки вдалося досягти середньоквадратичного значення похибки в 0.0997. Для нейронних мереж іншого типу аналогічний показник знаходиться в діапазоні від 0.1052 до 0.1157.

Для підвищення точності отримання даних розглянуто експеримент, в рамках якого дані отримувалися з 32 точок[74]. Значення ознак були обчислені за допомогою багатовимірної авторегресійної моделі[88]. Авторегресійна модель представляє собою модель, в якій поточне значення часового ряду лінійно залежить від попередніх значень цього ж ряду. Аналогічно до підходу з використанням нейронних мереж, дана модель дозволяє проаналізувати зміну параметрів у часі. В результаті отримано вектор ознак. При реалізації даного використовувалося 6 ознак, що описували дані, які отримано з точок Fp1, Fp2, C3, C4, O1 та O2. Для збільшення достовірності класифікації та зменшення розмірності даних, до отриманих векторів застосовано метод головних компонент[89]. З використанням отриманих даних побудовано модель, що базується яна методі опорних векторів. В результаті отримана достовірність класифікації 81.64%.

Таким чином, для методів, що в якості даних використовують значення ЕЕГ характерна висока достовірність. В якості точок, з яких отримуються дані найчастіше використовуються точки O1 та O2. Використання швидкого перетворення Фур'є дозволяє зі вхідного сигналу отримати гістограму, що описує густину сигналу певної частоти. Найвищу достовірність розпізнавання(93%) отримано за

допомогою методу, що базується на словнику ознак. Головними недоліками використання показань ЕЕГ для аналізу бадьорості є шуми, що виникають внаслідок дії електромагнітного поля, недостатній контакт електродів, шуми, що виникають внаслідок рухів тіла. Окрім того, для побудови електроенцефалограм необхідне обладнання, яке потребує спеціальних навичок при його монтажі та обережності при використанні.

За допомогою глибинної мережі переконань та її навчання з використанням методу, що базується на самодіагностиці побудовано систему, яка для визначення поточного стану оператора використовує значення одного джерела[90]. В якості еталонних значень стану оператора використовується метод самодіагностики, що базується на шкалі сонливості з 14 питань. Отримані дані ЕЕГ з частоти 512 Гц трансформуються до частоти 128 Гц для зменшення асимптотичної складності при обробці. Попередня підготовка даних полягає в побудові діаграм густини за допомогою швидкого перетворення Фур'є та методу Уелша. Наближена ентропія( $ArEn$ )[91] використовується для визначення складності та нерівномірності ЕЕГ сигналу. Високе значення  $ArEn$  свідчить про випадковість величини, в той же час низьке значення свідчить про передбачуваність даних. Значення  $ArEn$  зменшується зі збільшенням дрімоти. За допомогою групового вейвлет-перетворення[92], що представляє собою лінійну комбінацію вейвлетів вхідний сигнал розкладено на 5 частотних діапазонів. Наступним кроком є побудова глибинної мережі переконань для визначення поточного стану оператора. Для отриманої системи досягнуто достовірність розпізнавання в 98.86%.

## 1.6 Параметри тіла

Варіабельність серцевого ритму - різниця між двома послідовними скороченнями серця. Автономна нервова система людини ділиться на симпатичну та парасимпатичну. при активізації симпатичної нервової системи частота пульсу та варіабельність серцевого ритму збільшується, а при активізації парасимпатичної



- навпаки. Зниження варіабельності серцевого ритму свідчить про активізацію парасимпатичної нервової системи, що може свідчити як про хвороби так і про стомленість.

При використанні в якості джерела даних ЕКГ та показників серцевого ритму існує два різноманітних підходи до їх аналізу[93]. Спільним першим кроком для обох алгоритмів є перетворення вхідних даних у гістограму розподілу їх значень за допомогою швидкого перетворення Фур'є. Наступним кроком є побудова моделі, що за поточним значенням визначає стан оператора. Перший підхід полягає у обчисленні середнього арифметичного значень серцевого ритму для навчальної вибірки. На етапі тестування стан оператора визначається як дрімота, якщо частота серцевих скорочень менша за 95% від обчисленого середнього значення. В другому підході використовується розбиття вхідного сигналу на два діапазони: високої(0.15 -0.4 Гц) та низької(0.04 - 0.15 Гц) частоти. Відношення кількості значень низької частоти до кількості значень високої частоти отримало назву LF/HF ratio[94]. При перевищенні цього значення за 95% система класифікує поточний стан як дрімоту. Помилка першого роду для підходу з використанням в якості порогового значення середнього значення вибірки коливається в діапазоні від 1% до 49%, а середнє значення становить 20.5%. Аналогічний показник для другого підходу коливається від 3% до 53%, з середнім значенням рівним 21.4%.

Наступний підхід базується на варіабельності серцевого ритму[95]. Для кожного оператора емпіричним шляхом встановлено середнє значення частоти серцевих скорочень. Так як, при отриманні значень ЕКГ можливі різкі перепади значень, то введено поріг, при перевищенні якого поточні показники відкидається з розгляду. При попередній обробці сигналу ЕКГ, першим етапом є нормалізація значень, що описують кожен момент часу. Для всіх значень, що не перевищують порогове отримано значення стандартного відхилення. Наступним кроком є оцінка отриманого сигналу за шкалою: відмінно, добре, задовільно, незадовільно. Для цього додатково введено два параметри, що описують діапазон, якому повинен належати сигнал. Якщо сигнал не належить до даного діапазону, то він оцінюється як

незадовільний. Якщо даний сигнал не було відкинуто, то, залежно від додатково введених інтервалів, що описуються емпірично отриманими значеннями  $E_1$ ,  $E_2$ ,  $E_3$  та різницею у дисперсії між поточним інтервалом та середнім значенням попереднього інтервалу, сигнал відноситься до певної категорії. В результаті, лише половина отриманих значень може бути використана для побудови класифікатора. Шляхом застосування спеціалізованих алгоритмів[96, 97] до отриманого набору значень отримано значення варіабельності серцевого ритму та частоти дихання. Для визначення поточного стану оператору використовується лінійний дискримінантний аналіз. В результаті отримано достовірність розпізнавання в 59%, а похибка другого роду становить 4%. Важливим висновком є те, що класифікатор занадто чутливий до глобального стану оператору, а особливо до позбавлення сну. Під час проведення експериментальних досліджень було заборонено від'єднувати електроди від тіла для зменшення ризику неправильного подальшого підключення, що свідчить про складність використання даного підходу.

При використанні інфрачервоного датчику пульсу побудовано підхід, що базується на пульсі та варіабельності серцевого ритму[98]. Для нормального стану пульс оператора знаходиться в діапазоні від 70 до 100 ударів за хвилину. При дрімоті цей показник знижується до 45-65 ударів за хвилину. В якості основного параметру при визначенні стану використовується варіабельність серцевого ритму. Так, для нормального стану даний значення даного показнику знаходиться близько 0.7Гц, в той час як під час дрімоти він знижується до 0.183Гц. Перевагою даної системи над аналогічними є використання сенсору, що не потребує додаткового монтажу з боку оператора.

Параметром, що залежить від частоти серцевих скорочень та характеризує стан людини є інтенсивність дихання. Одним з підходом для оцінки інтенсивності дихання з використанням безконтактних сенсорів є використання сенсорного контролеру Kinect. При порівнянні з еталонним сигналом, кореляція між показами датчиків становить близько 99%[99]. При засинанні інтенсивність дихання знижується в середньому на 4 цикли в хвилину за 39 секунд[100]. Різниця між частотою

дихання у нормальному стані та під час дрімоти доволі значна і дозволяє доволі точно оцінити поточний стан оператора[101]. Серед головних недоліків даного підходу є занадто висока чутливість інтенсивності дихання до навколишнього середовища та фізичного стану оператора.

Одним з параметрів, що незначним чином змінюється при нормальному функціонуванні організму та значним чином коливається при переході між фазами сну є провідність шкіри[102]. Окрім значень провідності шкіри в даному алгоритму використовується значення пульсу. Для кожного двосекундного інтервалу отримано значення 6 статистичних функцій(середнє значення, стандартне відхилення, потужність сигналу, максимальна частота, стандартне відхилення частоти спектру та її середнє значення), значення їх градієнту та похідної. Отримані значення подані на вхід класифікатора. В якості класифікатора використано багатосаровий персептрон з гіперболічним тангенсом у якості функції активації. Для даного класифікатора отримано значення достовірності розпізнавання в 92.61%.

Таким чином, дана група параметрів доволі точно описує стан оператора але через складність використання побудованих систем даний підхід не отримав широкого розповсюдження. Головним недоліком є висока інерційність параметрів та їх висока кореляція зі станом оператора, що не пов'язаний зі станом дрімоти. Окрім того, всі досягнуті результати отримані внаслідок використання комбінації параметрів, тобто жоден з наведених вище параметрів в недостатній мірі визначає поточний стан оператора.

Аналіз провідності шкіри у поєднанні з методами для підтвердження власного стану активно використовується на залізничному транспорті[103]. Система представляє собою сукупність модулів для оцінки стану оператора. В якості датчика, що оцінює опір шкіри використовується браслет або годинник. У випадку, коли опір шкіри за своїм значенням близький до показників, які набуваються у стані дрімоти система потребує підтвердження бадьорості оператора шляхом впливу на кнопку системи. Середня кількість запитів на підтвердження стану становить близько 7.2 в годину. Завдяки структурі системи та необхідності своєчасної оцінки

свого стану, достовірність розпізнавання даної системи становить близько 100%. Головним недоліком побудованої системи необхідність мануального підтвердження власного стану.

## 1.7 Порівняння алгоритмів

Залежно від типу параметру та системи, що побудована для його аналізу існує ймовірність помилки. Для даних систем характерні помилки першого та другого роду. Ймовірність відмови кожної з систем [104] наведено в табл.3.

Параметри	Інтенсивність відмов системи	Середня кількість тривоги за годину
Зміна характеру водіння	0.3	5
Пульс	0.3	5
Мова	0.2	5
Поза та тонус м'язів	0.2	5
Напрямок погляду	0.2	3
Нахил голови	0.1	2
Блимання	0.05	3
Окулограма	0.0001	2
Опір шкіри	0.001	1

Табл. 3: Порівняння відмовостійкості систем розпізнавання стану оператора

Під інтенсивністю відмов системи розуміється кількість помилок першого роду в годину. Середня кількість тривоги складається з правильних спрацювань системи та помилок другого роду.

Попередня підготовка даних	Класифікатор	Використані параметри	Достовірність розпізнавання
----------------------------	--------------	-----------------------	-----------------------------

Перетворення Фур'є, метод Уелша, вейвлет-перетворення	Глибинна мережа довіри	Ритми ЕЕГ	98.86%
FACS, Фільтр Габора та метод опорних векторів	Мультиномінальна логістична регресія	Блимання, рух верхньої брови, рух підборіддя, насуплення носу, посмішка, звуження ноздрів	97.9%
	Нейронна мережа AlexNet	Напрямок погляду	95%
	Комбінація штучних нейронних мереж	PERCLOS, напрямки погляду, рух голови	94.69%
Аналіз кількісної регресії, лінійний дискримінантний аналіз	Наївний баєсівський класифікатор	Характеристика детермінованості ритмів ЕЕГ	93.2%
Дискретне вейвлет-перетворення	K-SVD	Ритми ЕЕГ	93%
Побудова статистичних функцій	Багатошаровий персептрон	Провідність шкіри	92.61%
	Метод опорних векторів, ймовірнісний метод опорних векторів	Ритми ЕЕГ	91.92%

Перетворення Фур'є, метод Уелша	Регресійний метод опорних векторів	Ритми ЕЕГ	89%
Перетворення Фур'є	Відстань Махалобіуса	Ритми ЕЕГ	88.7%
Фільтр Калмана	Штучна нейронна мережа	PERCLOS	86.07%
	Метод опорних векторів	PERCLOS, напрямки погляду, частота блимання	84.51%
Фільтр Кенні, оператор Собеля, метод головних компонент	Метод опорних векторів	PERCLOS	83,64%
	Багатовимірна авторегресійна модель	Ритми ЕЕГ	81.64%
Нормалізація даних	Лінійний дискримінантний аналіз	Варіабельність серцевого ритму та частоти дихання	59%
Перетворення Фур'є	Порівняння значень з еталонним	Показники серцевого ритму	21.4%

Таким чином найбільш точними є алгоритми, що базуються на показах ЕЕГ. За умови відсутності даних ЕЕГ, з використанням методів, що базуються на візуальній інформації (PERCLOS, блимання, рух верхньої брови, напрямки погляду, емоції) отримані результати достатні для точного визначення ФСО.

## 2 Дослідження залежності параметрів організму та функціонального стану оператора

### 2.1 Опис навчальної вибірки

Під час побудови математичного опису вхідного відеопотоку, для зменшення впливу положення голови в просторі - вона була зафіксована в одній позиції. Запис відео проводився офісною веб-камерою за умов нормального освітлення. Кожен фрейм відеопотоку представляє собою зображення з роздільною здатністю 640x480, а відео записане з частотою від 20 до 30 кадрів за секунду. Емпіричним шляхом встановлено, що для підвищення достовірності локалізації ключових точок обличчя та обчислення їх характеристик, доцільно виділити контури об'єктів на початковому зображенні за допомогою фільтру Кенні та використовувати отримане зображення в якості вхідного. Приклад попередньої обробки зображення приведено на рис.2.

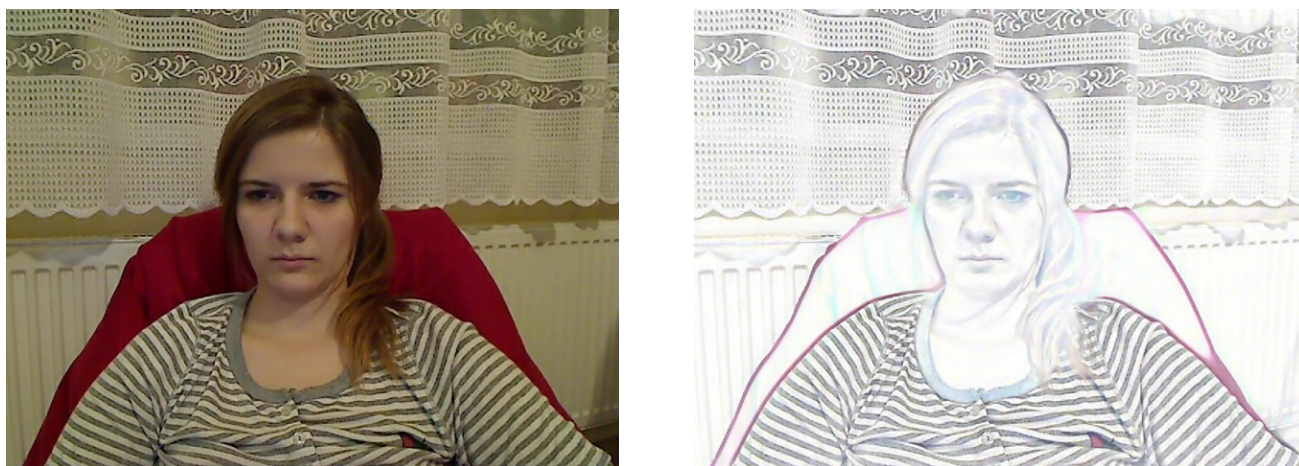


Рис. 2: Попередня обробка зображення

При побудові навчальної вибірки використовувались обидва зображення, а результуючі дані є усередненими показниками, що отримані в результаті аналізу пари зображень.

Кожний елемент опису вхідного відеопотоку містить наступні дані:

- час з точністю до мілісекунд

- координати кутів прямокутника, що обмежують обличчя у форматі лівого нижнього та правого верхнього кутів
- опис кожного ока, що складається з:
  - площу ока
  - координати центру та радіус зіниці
  - список точок, що описують нижню та верхню повіки
- координати точок, що описують центр скроні
- координати 4-х точок, що рівномірно розподілені вздовж спинки носа
- координати точок, що описують нижню та верхню губи

Вибірка, що описує нормальний стан оператора побудована шляхом зйомки рутинної роботи людини за комп'ютером в бадьорому стані. Побудова вибірки, що описує сонливий стан оператора побудована шляхом запису зовнішніх ознак оператора, що знаходиться в стомленому стані, а, як наслідок, на стадії дрімоти, що періодично переходить в легкий сон. Приклад опису фрейму наведено в додатку 1.

## **2.2 Алгоритми і методи локалізації ключових точок обличчя**

Головною задачею методів даної групи є швидке та точне визначення позиції обличчя та точок, що описують різні його регіони. В подальшому отримана інформація може бути використана при визначенні ФСО. З метою швидкої локалізації обличчя при зміні фреймів використовуються алгоритми відстеження позиції об'єкту в відеопотоці, що характеризується відсутністю необхідності обробки зображення цілком при незначній зміні сцени між фреймами. Найбільш розповсюдженим та оптимальним методом пошуку об'єкту на зображенні є алгоритм



Віоли-Джонса[105], що базується на каскадах Хаара. Для застосування даного алгоритму пошуку обличчя на зображенні необхідно використовувати відповідні каскади Хаара. Підвищення швидкодії алгоритму та збільшення достовірності розпізнавання досягається шляхом попередньої обробки зображень та зменшення його роздільної якості[106].

Відстеження зміни позиції об'єкту в відеопотоці реалізується за допомогою алгоритму, який побудовано з використанням адаптивних кореляційних фільтрів[36], а саме мінімізації вихідної суми квадратичної помилки (MOSSE). Головною перевагою даного алгоритму над алгоритмом Віоли-Джонса є висока швидкодія при зміні фрейму. Так, асимптотика методу Віоли-Джонса пошуку об'єкту на зображенні складає  $O(PWK)$ , де  $P$  - кількість пікселів на зображенні,  $W$  - кількість можливих позицій вікна сканування, а  $K$  - кількість ознак Хаара, необхідних для обробки регіону. В той же час, для алгоритму MOSSE обробка першого фрейму потребує  $O(NP \log P)$ , а кожного наступного  $O(P \log P)$ , де  $P$  - кількість пікселів на зображенні, а  $N$  - кількість афінних перетворень, необхідних для початкової ініціалізації фільтру.

Висока достовірність локалізації очей на зображенні досягається шляхом паралельної обробки відеопотоків з видимого та інфрачервоного спектрів[37]. За допомогою фільтру, що відсікає видимий спектр отримано зображення в інфрачервоному діапазоні. При цьому обробка зображень, що отримані в інфрачервоному спектрі значним чином відрізняється від аналогічної операції для видимого спектру. Головна відмінність - за допомогою інфрачервоного освітлення зменшується вплив умов освітлення в видимому діапазоні, а, як наслідок, отримані зображення менш контрастні та мають більший рівень шумів. В той же час, при даній реалізації системи спостереження спостерігається значний контраст зіниці відносно ока. Дана особливість спостерігається лише при близькості осі спостереження та джерела світла. Даний підхід може бути використаний для виявлення повороту або нахилу голови. Усунення артефактів контрасту та яскравості виконується за допомогою нелінійної обробки морфологічним фільтром. Наступним кроком

виконується локалізація ока та його складових частин на зображенні. Використовуючи модель, що базується на представленні зіниці та райдужної оболонки реалізується функція пошуку ока. Розміри зіниці  $p_w, p_h$  встановлені експериментально та становлять 11-13 пікселів у кожному з вимірів. Зіниця та райдужна оболонка представляють собою концентричні кола. Для пошуку зіниці виконується послідовна обчислення середньої яскравості пікселів для всіх регіонів розміром  $p_w \times p_h$ . Якщо отримане значення перевищує заданий поріг, визначений експериментальним шляхом, то даний регіон класифікується як можлива позиція зіниці. Наступним кроком, для всіх знайдених регіонів, аналогічним чином, виконується перевірка регіону навколо зіниці на можливість її належності до райдужної оболонки. Наступним кроком серед отриманої множини обираються регіони, що не перетинаються. Серед регіонів, що залишилися після виконання описаних вище фільтрацій обирається регіон з середньою яскравістю, що чисельно найближча до заданого значення. Для пошуку ока у видимому спектрі виконується пошук регіонів, що базується на кольоровій характеристиці шкіри. Класифікатор, який базується на сингулярному розкладенні вищого порядку[107] і призначений для виявлення відкритого ока отримує дані, що підготовлені на попередньому кроці. Таким чином отримана система, що з достовірністю близько 99% локалізує око на зображенні у видимому та інфрачервоному спектрі. Отримані дані можуть бути використані при класифікації ФСО.

## 2.3 Метрика Eye Aspect Ratio

Так як око представлено багатокутником, то шляхом його триангуляції можливо обчислити його площу як суму площ трикутників, з яких він складається. Для обчислення площі ока використовується знакова площа трикутника. За допомогою даного підходу можливо обчислювати площу будь-яких багатокутників. Площа багатокутника є сума площ трикутників, що утворені трьома точками, дві з яких представляють сторону багатокутника, а третя - будь-яка інша точка. Для

обчислення площі ока використовується орієнтована площа трикутника, що обчислюється як детермінант наступної матриці:

$$2 \cdot S = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

Внаслідок обчислення та спрощення формули для обчислення детермінанту даної матриці, отримано наступний вираз:

$$\begin{aligned} 2 \cdot S &= x_1 \cdot y_2 + x_2 \cdot y_3 + y_1 \cdot x_3 - x_3 \cdot y_2 - x_1 \cdot y_3 - x_2 \cdot y_1 = \\ &= x_1 \cdot (y_2 - y_3) + x_2 \cdot (y_3 - y_1) + x_3 \cdot (y_1 - y_2) = \\ &= (x_2 - x_1) \cdot (y_3 - y_1) - (y_2 - y_1) \cdot (x_3 - x_1) \end{aligned}$$

Для зменшення кількості обчислень використовується характеристика EAR (eye aspect ratio)[108] - характеристика ока, що описує відношення відстані між кутами ока та парами відповідних точок, які розміщені на верхній та нижній повіці (рис.3).

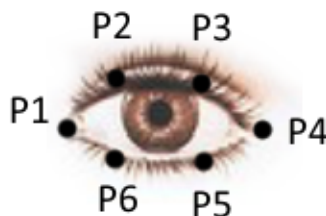


Рис. 3: Ключові точки для обчислення EAR.

Використовуючи функцію, що залежить від описаних ключових точок можливо чисельно представити поточний стан ока:

$$EAR = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2 * ||p_1 - p_4||}$$

Характеристика EAR має сталі значення при відкритому оці і оперативно реагує на процес закриття ока, шляхом зменшення свого значення до нуля. Окрім EAR, характеристикою, що описує поточний стан ока є його площа. На відміну від

визначення стану ока з використанням даних про його площу, параметр EAR позбавлений недоліків, що виникають внаслідок похибок, викликаних умовами освітлення та положенням голови відносно камери.

## 2.4 Попередня підготовка даних

З метою підвищення достовірності класифікації шляхом введення додаткових ознак, що є композицією існуючих, виконується трансформація вхідних даних. Окрім того, виходячи з припущення про взаємозв'язок між частотою блимання та ФСО, побудовано вибірку, що описує блимання оператора та їх характеристики. Для оцінки функціонального стану оператора та передбачення ймовірності його засинання необхідно сформувати навчальну вибірку з ознак, що характеризують стан оператора в конкретний момент часу. Шляхом обробки кожного фрейму побудовано 2 вибірки, що чисельно описують параметри тіла оператора:

1. вибірка з блиманнями (blinks) - кожний елемент вибірки описує блимання, які зафіксовано за весь період спостереження у форматі:
  - тривалість закриття ока
  - тривалість відкриття ока
  - амплітуда зміни площі ока
  - амплітуда зміни EAR
  - інтервал між блиманнями
2. вибірка з параметрами обличчя (timeline) - кожний елемент описує положення обличчя та окремих його елементів у просторі, та складається з наступних параметрів:
  - EAR лівого ока
  - EAR правого ока

- видима площа лівого ока
- видима площа правого ока
- видима площа ока
- повздовжній нахил голови
- поперечний нахил голови
- переміщення голови відносно попереднього фрейму
- переміщення зіниці відносно попереднього фрейму
- площа роту, що обмежена губами
- кумулятивна середня площа роту, що обчислена за допомогою ковзного середнього за даними останніх 100 фреймів
- кумулятивна середня площа лівого ока, що обчислена за допомогою ковзного середнього за даними останніх 100 фреймів
- кумулятивний середній повздовжній нахил голови, що обчислений за допомогою ковзного середнього за даними останніх 100 фреймів
- кумулятивна середній поперечний нахил голови, що обчислений за допомогою ковзного середнього за даними останніх 100 фреймів

Для вхідних даних, розміром в 37538 фреймів отримано вибірки: `timeline`, що складається з 37538 елементів та вибірку `blinks` з 312 елементів.

## 2.5 Дослідження залежності характеристик елементів обличчя та функціонального стану оператора

Для кожного стану, що описується Каролінською шкалою сонливості характерне певне числове значення параметрів, що описують око. Для спрощення реалізації алгоритмів визначення ФСО, задачу зведено до бінарної класифікації. Так, з множини станів оператору виділено два ключових: нормальний стан оператора

та стан сонливості. Нормальний стан відповідає звичайному функціонуванню, а стан сонливості включає в себе дрімоту та процес переходу до неї.

Серед параметрів, що найбільш повно залежать від ФСО виділено наступні: EAR, частота блимання, інтенсивність переміщення зіниці, видима площа ока, вертикальний та горизонтальний нахили голови та інтенсивність їх зміни, інтенсивність рухів губами. Для всіх елементів обличчя в стані сонливості спостерігається зменшення активності, що характерна для нормального стану. При сонливості у людини сповільнюються рухи та змінюється їх амплітуда та інтенсивність. В аспекті очей про це свідчить зменшення швидкості та інтенсивності блимання ока та швидкості переміщення і фокусування зіниці. Зниження швидкості переміщення зіницю вказує на стомленість та збільшення ризику засинання людини[109, 110]. Окрім того, в стані сонливості зменшується інтенсивність рухів головою без зменшення амплітуди. Залежність інтенсивності зміни значення параметру від ФСО наведено на графіках нижче.

## Повіки

Математично кожна повіка представляє собою параболу, для побудови якої необхідно знайти точки, що її описують. Для кожної повіки за допомогою алгоритму, який побудовано з використанням ансамблю регресійних дерев[111] та його реалізації в бібліотеці *dlib* знайдено 4 точки, включаючи внутрішній та зовнішній кути ока. Для більш точного результату інтерполяції, шляхом усереднення значень сусідніх точок було отримано додатково три точки. За допомогою методу найменших квадратів для даного набору точок була побудовано параболу, що описує повіку. В загальному матриця для інтерполяції методом найменших квадратів виглядає наступним чином:

$$\begin{pmatrix} n & \sum x & \sum x^2 \\ \sum x & \sum x^2 & \sum x^3 \\ \sum x^2 & \sum x^3 & \sum x^4 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum y \\ \sum x \cdot y \\ \sum x^2 \cdot y \end{pmatrix},$$

де коефіцієнти  $a, b, c$  - шукані коефіцієнти параболи. Шляхом множення матриць отримано наступну систему рівнянь, яка може бути розв'язана за допомогою методу Гаусса.

$$\begin{cases} a \cdot n + b \cdot \sum x + c \cdot \sum x^2 = \sum y \\ a \cdot \sum x + b \cdot \sum x^2 + c \cdot \sum x^3 = \sum x \cdot y \\ a \cdot \sum x^2 + b \cdot \sum x^3 + c \cdot \sum x^4 = \sum x^2 \cdot y \end{cases}$$

Розв'язком даної системи є коефіцієнти  $a, b, c$  рівняння параболи яка представлена у вигляді квадратичної функції  $y = a \cdot x^2 + b \cdot x + c$ , де залежно від зміни коефіцієнту  $a$  змінюється кривизна параболи, а при зміні значень коефіцієнтів  $b$  та  $c$  - її положення на площині. Коефіцієнти  $b$  і  $c$  жодним чином не використовуються в процесі визначення ФСО, адже залежать лише від положення обличчя в фреймі. Важливим параметром при визначенні ФСО є коефіцієнт  $a$ , що безпосередньо залежить від стану ока. Так, при зростанні абсолютного значення коефіцієнту  $a$  збільшується кривизна параболи, що є "кривизною" повіки. Позначимо коефіцієнт  $a$  отриманої параболи як коефіцієнт кривизни повіки(ККП). Зміна ККП свідчить про зміну напрямку погляду або відкриття чи закриття ока. Обчислені таким чином значення ККП для обох повік залежно від того закрито око чи відкрито зображено на рис.4, під час отримання вхідних даних, напрямком погляду є статичним.

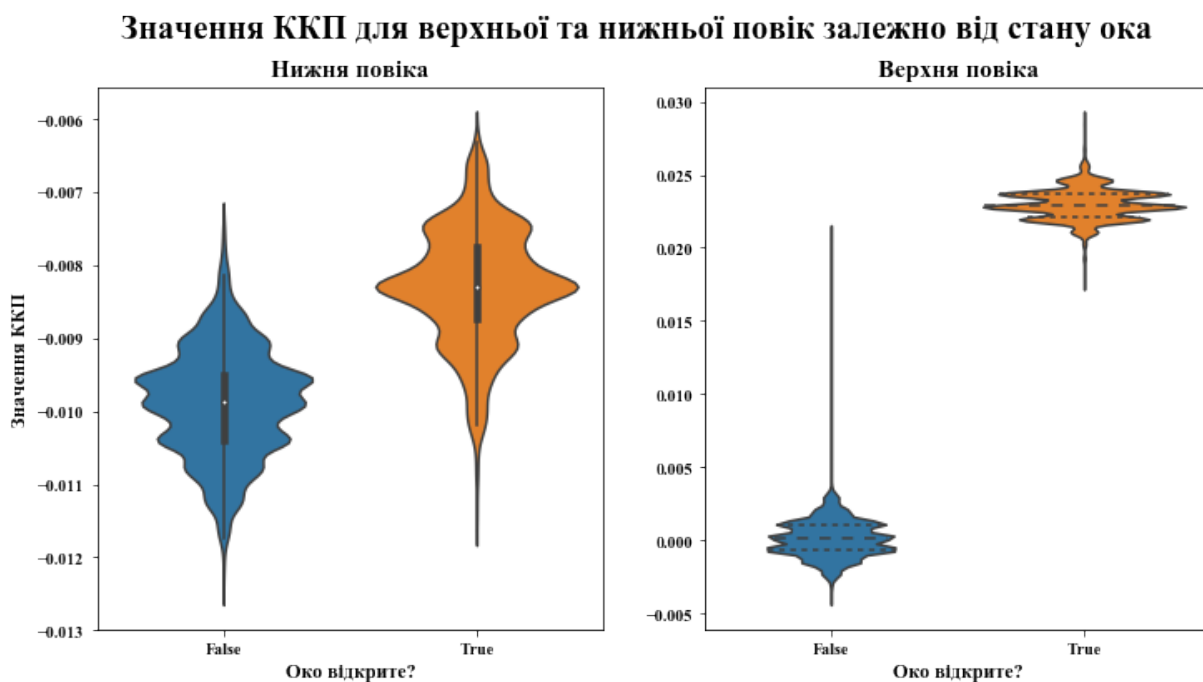


Рис. 4: Залежність коефіцієнту кривизни верхньої повіки від стану ока

На рис.4 спостерігаємо, що дисперсія значень ККП кожної повіки змінюється, залежно від ФСО. При цьому, дисперсія значень ККП верхньої повіки не перетинається, залежно від стану, на відміну від аналогічного параметру для нижньої повіки.

Так, для нижньої повіки значення при відкритому оці значення ККП знаходяться в інтервалі  $[-0.012; -0.006]$ , а для закритого -  $[-0.012; -0.007]$ , тобто незалежно від стану ока, ККП нижньої повіки знаходиться в одному діапазоні або діапазони розміщені близько та відсутня можливість їх чіткого розділення. В той же час, для верхньої повіки при відкритому оці значення ККП належить інтервалу  $[0.018; 0.029]$ , а при закритому -  $[-0.004; 0.21]$ . Тобто, ККП верхньої повіки можливо чітко розділити на два класи, залежно від ФСО, що свідчить про високу кореляцію між цими двома величинами, а як наслідок можна зробити висновок, що ККП верхньої повіки залежить від ФСО.

Зазначимо, що позитивне значення ККП отримане внаслідок того, що початок системи координат знаходиться у лівому верхньому куті зображення. Дані, що значним чином виходять за межі нормального розподілу відносяться до шу-



му. Для коректного функціонування класифікаторів та підвищення достовірності класифікації в цілому виконується відкидання величин, що є шумами. Для цього використовується z-оцінка, за допомогою якої відкидаються всі значення, які знаходяться на віддаленні більше ніж 2 стандартних відхилення від середнього значення послідовності.

На відміну від верхньої повіки кривизна параболи, що описує нижню повіку змінюється незначним чином залежно від класу та не дозволяє точно визначити поточний стан ока. Тому дані значення недоцільно використовувати при визначення стану ока. Значне зменшення абсолютного значення ККП свідчить про випрямлення параболи, а тому і про факт блимання людини. Зміна значення ККП за часом зображено на рис. 5.

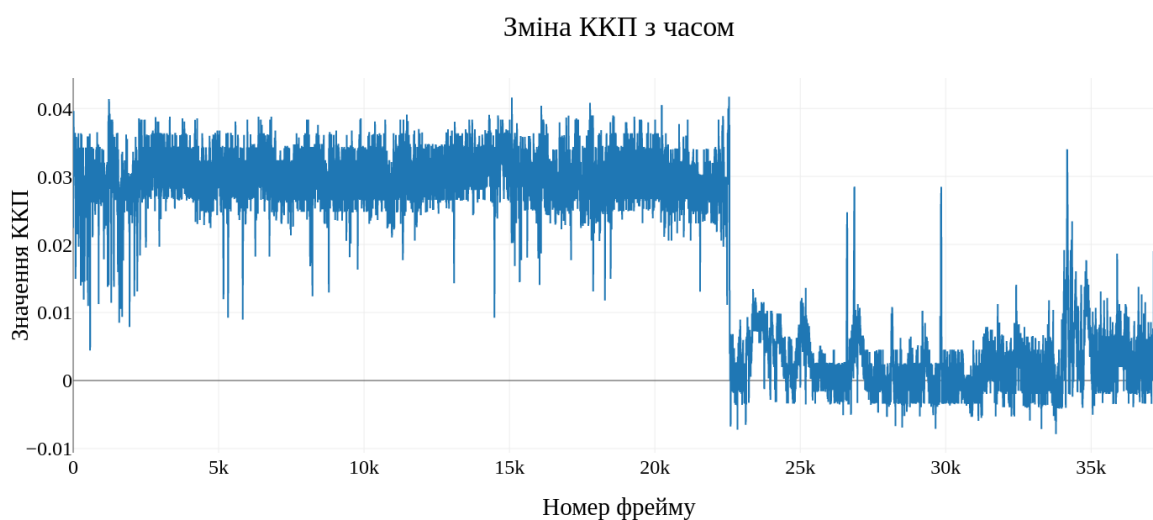


Рис. 5: Залежність коефіцієнту  $\alpha$  верхньої повіки від стану ока

На рис.5 спостерігається очевидний розподіл значень при звичайному функціонуванні ока та динаміку зміни значення ККП при переході оператора між функціональними станами. Так, після 22000 фреймів оператор переходить до сонливого стану, що чітко спостерігається на графіку. Очевидно, що при виконанні фільтрації з використанням z-оцінки, розбіжність між класами значним чином збільшиться. Таким чином даний параметр тісно корелює з ФСО, а тому є доцільним його використання в якості однієї з ознак при визначенні ФСО.

Для розбиття блимання на стадії доцільно дослідити зміну значення ККП верхньої повіки для одного циклу блимання.

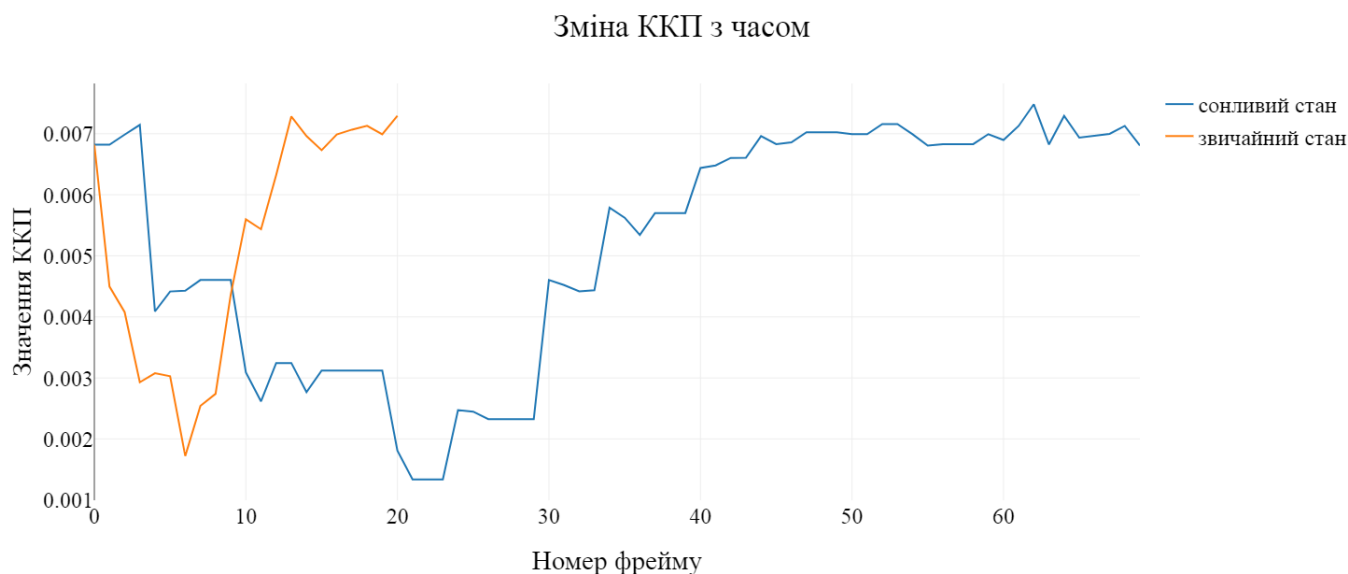


Рис. 6: Залежність коефіцієнту  $\alpha$  верхньої повіки впродовж одного циклу блимання від ФСО

На рис.6 спостерігається значна зміна ККП для верхньої повіки залежно від стадії блимання. Графік зміни можливо розділити на 3 стадії: зменшення значення ККП - око закривається, перегин - око закрите та зростання значення ККП - око відкривається. Залежно від поточного стану ока, значення ККП потрапляє до певного діапазону. Таким чином, значення ККП верхньої повіки корелює зі станом ока та може бути використано для локалізації моментів блимання та їх класифікації за тривалістю та амплітудою.

Дослідимо поведінку ККП для верхньої повіки, залежно для вибірок, що описують звичайний та сонливий стан оператора для одного циклу блимання. Дисперсія ККП для обох вибірок для одного циклу блимання однакова, але вона досягається за різний проміжок часу, тобто, амплітуда руху повіки залишається незмінною, незалежно від стану оператора. Для більш точного визначення ФСО, окрім кривизни повік доцільно досліджувати інтервал за який змінюється їх кривизна та значення ККП набуває свого початкового значення. Для звичайного циклу блимання мінімальна величина ККП становить 0.002 і досягається за 6

фреймів, а для сонливого становить 0.001 і досягається за 22 фрейми. Після цього величина ККП зростає до свого максимуму, що є початковим значенням, за 7 фреймів для нормального блимання та за 28 фреймів для сонливого. Тобто, час зміни ККП від свого максимального значення до мінімального та повернення до початкового значення корелює з ФСО.

Визначення проміжків часу, в які відбулося блимання виконано з використанням пошуку за патерном серед значень ККП для верхньої повіки. Через недосконалість обладнання та похибки алгоритмів локалізації ключових точок та інтерполяції, дані необхідно наблизити за допомогою рухомого середнього[112]. Завдяки даній трансформації вдалося позбутися шумів у вибірці та досягти більшої достовірності при локалізації інтервалів спадання та зростання величини. З метою підвищення точності обчислення ККП доцільно об'єднати проміжки часу, за які ККП змінилося незначним чином. Тобто, декілька блимань, під час яких око значний проміжок часу знаходилося в одному з проміжних станів і величина ККП не досягла свого початкового значення будуть локалізовані як одне блимання. Також, якщо між інтервалами спадання та зростання незначний проміжок часу у фреймах, то це свідчить про їх належність до одного циклу блимання ока.

Альтернативним підходом до локалізації блимань є оцінка зміни площі ока, що прямо залежить від відстані між верхньою та нижньою повікою. Відстань між повіками представляє собою середнє арифметичне відстаней між двома рівновіддаленими від центру повіки точок.

З метою дослідження кореляції між значенням ККП та ФСО побудовано графік, що характеризує розподіл величини (рис.7).

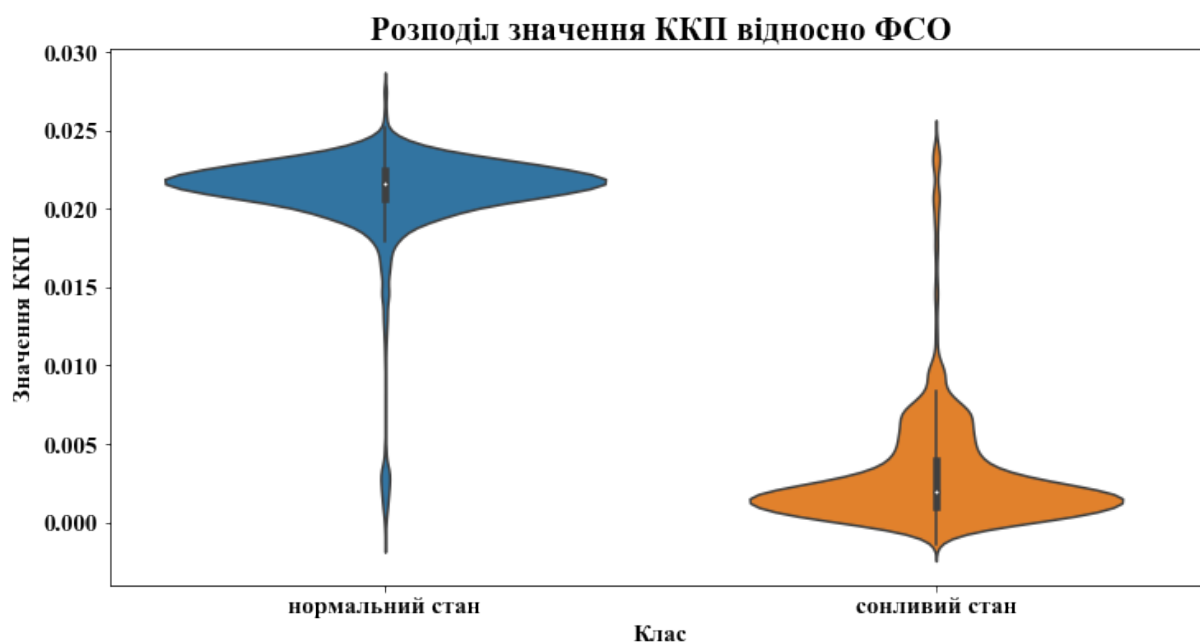


Рис. 7: Розподіл значень ККП залежно від ФСО

На рис.7 спостерігається значна відмінність дисперсій значень ККП для різних функціональних станів оператора. Відсутність значного перетину діапазонів значень свідчить про інформативність даного параметру при використанні його в якості ознаки під час визначення ФСО. Незначний перетин інтервалів виникає через наявність викидів та похибок обчислення значень параметру. Так як значення ККП корелює з площею ока, то отримані дані трактуються як такі, що описують око, яке частково перекривається верхньою повікою. Наприклад, для нормального стану характерне більше значення ККП, а як наслідок, і видима площа склери.

Отже, характеристики, що описують положення повік можливо використовувати в якості ознак при визначенні ФСО. Значною мірою з ФСО корелює значення ККП верхньої повіки. Значення ККП нижньої повіки змінюється незначним чином, залежно від стану оператора. Залежно від ФСО для одного циклу блимання спостерігається різна швидкість зміни значення ККП, що можливо використовувати при обчисленні характеристик блимання та їх інтенсивності.

## Частота блимання

Однією з характеристик, яка значним чином залежить від ФСО є частота блимання. Частота блимання є величиною індивідуальною та значною мірою може змінюватися залежно від зовнішніх подразників, тому при розрахунках в якості ознаки будемо використовувати кількість блимань протягом певного інтервалу часу. В якості інтервалу часу обрано останні 1000 фреймів, що приблизно відповідає 2 хвилинам спостереження. Таким чином нівелюються анатомічні особливості організму та вплив зовнішніх подразників. Взаємне розміщення дисперсій, обчислених для частот блимання для різноманітних ФСО зображено на рис.8.



Рис. 8: Залежність частоти блимання від ФСО

З рис.8 можна зробити висновок, що при сонливому стані дисперсія значень частоти блимання значно більша, ніж при нормальному стані. Отримані діапазони неможливо розділити на два класи, що не перетинаються, тому при доцільно використовувати дану величину в якості ознаки лише у комбінації з іншими ознаками.

## Очне яблуко

Одним з репрезентативних параметрів очного яблука, який можливо оцінити візуально - площа видимої частини склери та метрика EAR, що корелює з нею. Даний параметр значною мірою залежить від зовнішніх подразників і корелює з анатомічними особливостями організму та фізіологічним станом людини. Зміна площі склери може свідчити про наступні факти: зміна положення голови в просторі, значна зміна освітлення, вплив зовнішнього подразнику, блимання, зміна ФСО. Видима площа ока тісно корелює з ККП верхньої повіки, що наочно спостерігається на рис.9. З метою масштабування величин, абсолютне значення видимої площі ока розділено на 2000.

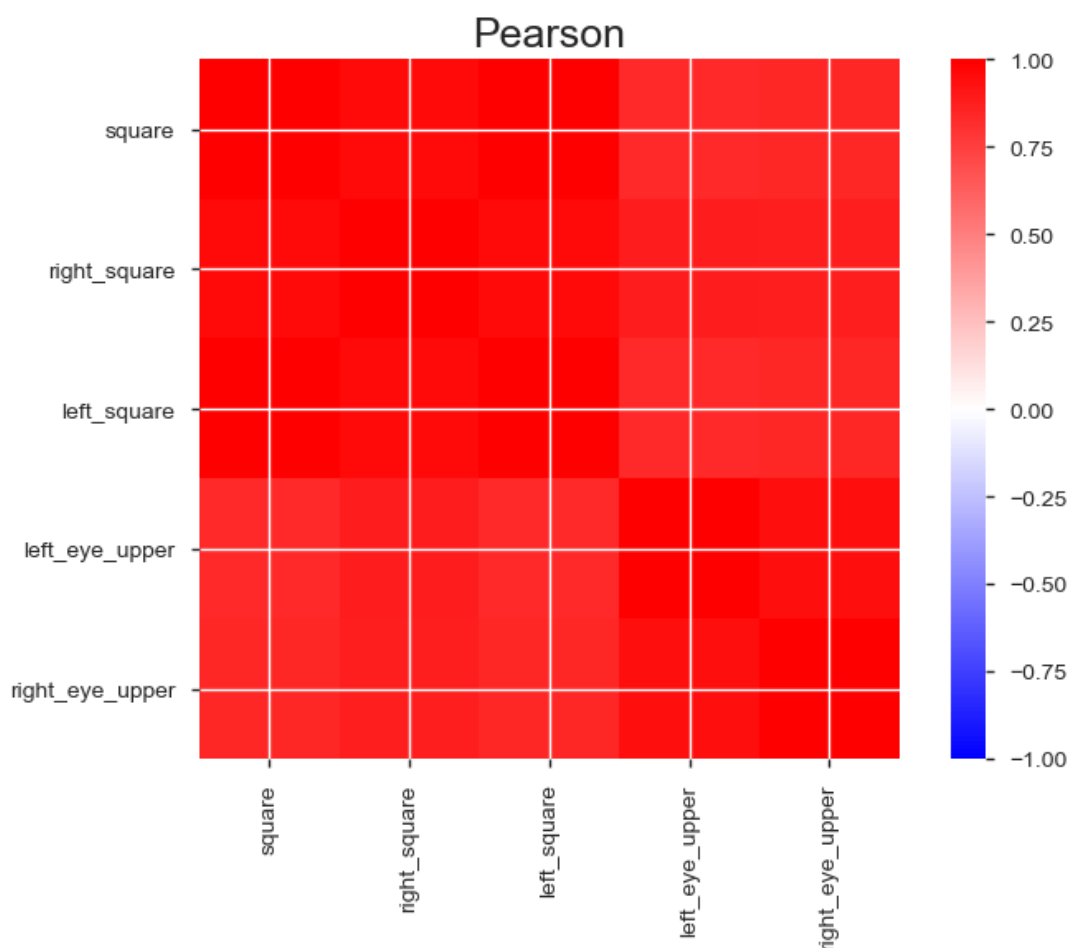


Рис. 9: Залежність площі від стану ока

На рис.9 спостерігається висока кореляція між усіма розглянутими параметрами, що описують стан окремих складових частин ока. Чисельно, для коефіцієнтів

кривизни верхньої повіки лівого та правого очей, значення коефіцієнту кореляції Пірсона становить  $p = 0.94173$ . Аналогічний показник для площ очей становить  $p = 0.95516$ . В той же час, коефіцієнт кореляції Пірсона між площею ока та ККП цього ж ока становить  $p = 0.82994$  та  $p = 0.87697$  для лівого та правого очей, відповідно. Таким чином, не має сенсу використовувати в якості ознак при визначенні ФСО кожне з очей окремо. В той же час, доцільно розглядати ККП та площу відповідного ока як окремі ознаки.

Для ККП характерні такі ж самі поведінкові патерни при зміні ФСО у часі, як і для площі ока. Головною відмінністю даних параметрів є їх абсолютне значення. Так як значення площі ока належить до множини цілих чисел, то при використанні його в якості ознаки, зменшується вплив похибок, що виникає при виконанні арифметичних операцій з дійсними числами.

Площа ока значним чином змінюється, залежно від ФСО. Так, при сонливому стані вона зменшується, а при бадьорому - навпаки, абсолютні значення будуть більшими за модулем. Оптимальним розподілом даної величини для розв'язання задачі класифікації є відсутність будь-якого перетину множин, що описують різні стани ока. Для підтвердження припущення про високу кореляцію між видимою площею ока та функціональним станом оператора, що визначає і стан ока, та відсутність перетину даних множин побудовані графіки розподілу величини, залежно від стану ока (рис.10). Навчальна вибірка була розбита на два функціональні стани, що не перетинаються: око закрите та око повністю відкрите, для кожної з отриманих вибірок побудову гістограму розподілу значень величини.

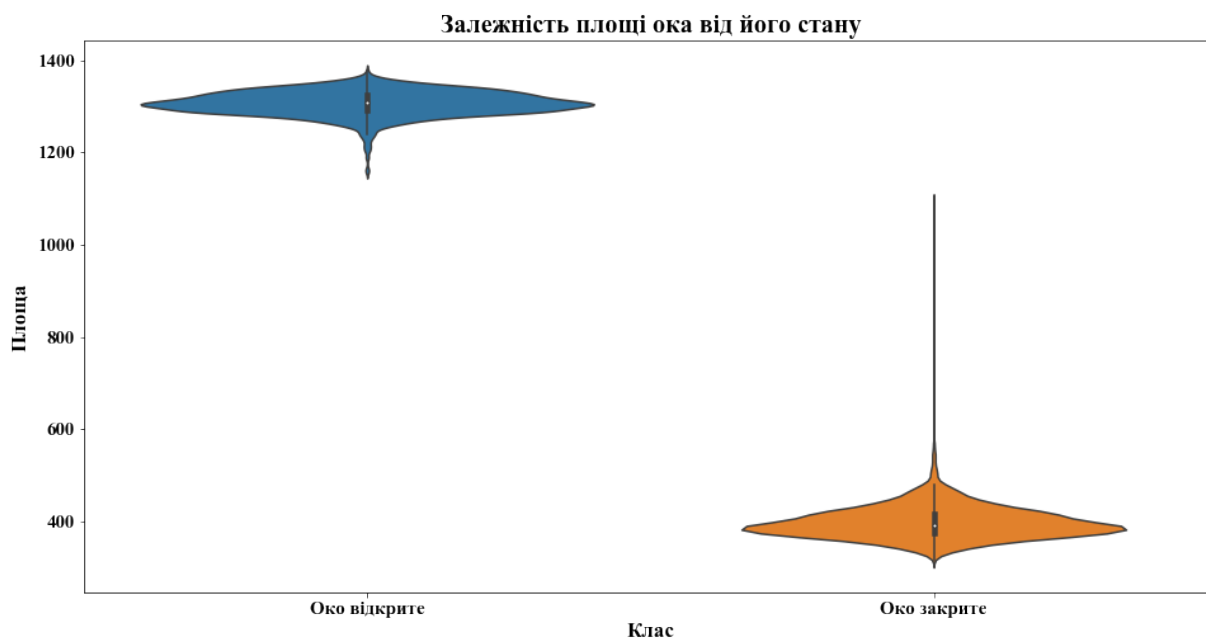


Рис. 10: Залежність площі від стану ока

На рис.10 спостерігається, що для відкритого ока значення площі знаходиться в інтервалі  $[1011; 1478]$  пікселів, для закритого ока аналогічний показник в інтервалі  $[215; 708]$  пікселів. Тобто, вибірки, що описують закрите та відкрите око є репрезентативними та придатними до використання їх в якості ознак розпізнавання.

Значна дисперсія значень для відкритого ока спровокована фізіологічними особливостями людини та неможливістю тримати верхню повіку нерухомою. Для закритого ока дисперсія значень спровокована незначними рухами зіниці при закритому оці та похибками алгоритму локалізації об'єктів. Ненульове значення площі для закритого ока пояснюється неможливістю точно оцінити межі верхньої та нижньої повіки, а, як наслідок, відстані між ними. Дана недосконалість алгоритму жодним чином не впливає на достовірність розпізнавання. За необхідності, можливо виконати нормалізації значень площі ока, та прийняти за нуль площу закритого ока. Під час блимання значення площі ока потрапляють до діапазону, що характеризує закрите око. Залежно від фізіологічних особливостей та функціонального стану оператора, абсолютні значення площі ока можуть незначним чином змінюватися, що не впливає на їх взаємне розташування.



Так як, площа склери залежить від множини параметрів, то для точного опису поточного стану ока необхідно враховувати поворот та нахил голови у просторі, зміну умов освітлення, тощо. При цьому, для зменшення впливу умов освітлення на точність оцінки площі видимої частини склери пропонується використовувати інфрачервоне освітлення[113, 114, 115, 116]. Отже, площа видимої частини склери доволі точно характеризує поточний стан ока. Незважаючи на відносно високу кореляцію значення площі ока зі значенням ККП цього ж ока, доцільно використовувати обидві величини в якості ознак розпізнавання.

## **Зіниця**

Одним з візуальних показників зміни функціонального стану оператора є числові значення характеристик зіниці ока оператора, а саме її положення в просторі та швидкість і інтенсивність переміщення. Локалізація зіниці в межах ока виконується за допомогою методу Оцу[117] для розділення компонентів зображення, що описують зіницю та склеру. Для знайденого регіону, що наближено описує зіницю було обчислено центр мас, який було прийнято за центр зіниці. Радіусом зіниці є мінімальна відстань від її центру до регіону іншого кольору, що є склерою. З метою дослідження гіпотези про залежність ФСО та інтенсивності переміщення зіниці побудовано діаграми розподілу значень залежно від стану оператора(рис.11). Під переміщенням зіниці в даному випадку розуміється зміна її координат відносно попереднього фрейму.

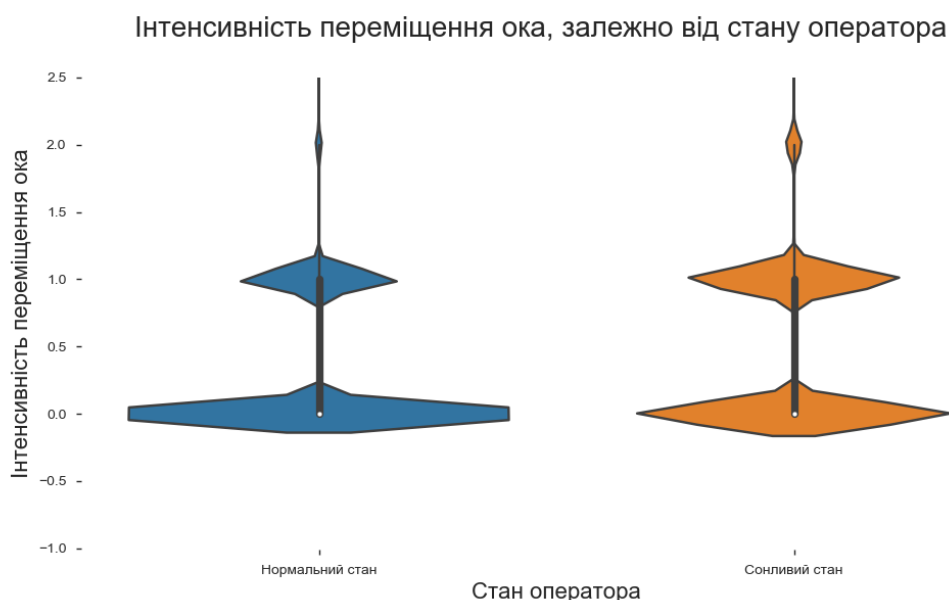


Рис. 11: Залежність переміщення зіниці від функціонального стану оператора

З рис.11 очевидно, що інтенсивність переміщення незначною мірою залежить від ФСО, але не може бути чітко розділена на два класи, що не перетинаються. Це може бути обумовлено низькою точністю локалізації зіниці та обчислення її переміщення. Навіть при даній точності локалізації можливо зробити певні висновки про взаємозв'язок переміщення зіниці та ФСО. На рис.11 незважаючи на ідентичні максимальні та мінімальні значення величини, розподіл незначною мірою відрізняється. Так, для обох величин спостерігається два інтервали, в які потрапляє значна частина значень. При порівнянні цих двох інтервалів спостерігаємо, що інтервал  $[0.75; 1.35]$  складається з більшої кількості значень, що описують сонливий стан оператора, а інтервал  $[0; 0.35]$  - зі значень, що описують нормальний стан. Для звичайного стану оператора характерна низька амплітуда та плавність переміщення зіниці. В той же час, хаотичність величини переміщення зіниці в сонливому стані свідчить про наявність саккадних рухів очей під час засинання. Хаотичність переміщення спостерігається у вигляді значної дисперсії величини, що не може бути пояснена викидами. Наявність саккадних рухів пояснюється спробами зберегти концентрацію на об'єкті при настанні дрімоти.

Головною проблемою при локалізації зміни є низька кадрова частота камери.

Очевидно, що для отримання всіх положень зіниці кадрової частоти звичайної камери недостатньо і більшість положень не було зафіксовано.

Таким чином, переміщення зіниці є показовою ознакою при оцінці ФСО, як і зміна площі ока та ККП. В той же час, недосконалість алгоритмів локалізації зіниці та низька якість вхідного зображення не дозволяє точно оцінити переміщення зіниці, а, як наслідок, дослідити взаємозв'язок між інтенсивністю переміщення зіниці та функціональним станом оператора.

## **Нахил голови**

Завдяки положення носу можливо визначити положення голови у просторі в цілому. Для цього використаємо 4 точки, що розміщуються рівновіддалено вздовж спинки носу, крайні з яких знаходяться на переніссі та на кінчику носа, а дві інші розміщені на рівній відстані між ними. За допомогою даних точок можливо оцінити нахил голови у двох площинах. Прийmemo нахил голови у площині, що паралельна лінії спостереження за повздовжній нахил, а нахил у площині, що перпендикулярна лінії спостереження - поперечний. При повздовжньому нахилі голови відстань між ключовими точками на носі зменшується або збільшується, залежно від напрямку нахилу та його амплітуди. Тобто, оцінюючи зміну відстані між крайніми точками на спинці носу можливо оцінити повздовжній нахил голови оператора. Поперечний нахил голови обчислюється як кут між прямою, що проводить вздовж спинку носу та прямою, що паралельна нижній границі фрейму.

З теоретичних міркувань, наведених вище, впливає гіпотеза про кореляцію між зміною інтенсивності рухів головою та ФСО. Для підтвердження або спростування даної гіпотези необхідно дослідити нахили голови у всіх припустимих площинах. Зміну повздовжнього нахилу голови оператора з часом зображено на рис.12. Числова характеристика нахилу голови в даному випадку є відстань між поточним та попереднім положенням ключових точок. Окрім амплітуди нахилу голови важливим параметром, що характеризує ФСО є швидкість її зміни. На

рис.12 для кожного фрейму зображено поточну відстань між двома найбільш віддаленими ключовими точками носу.



Рис. 12: Залежність переміщення спинки носа від нахилу голови

Різниця у значеннях відстані між точками для двох послідовних фреймів і є нахилом голови. Абсолютне значення цієї різниці є швидкістю зміни нахилу голови.

На рис.12 спостерігається, що при переході між ФСО інтенсивність зміни нахилу голови незначним чином змінюється. Так, для сонливого стану, що на графіку зображено після 23000 фреймів, значним чином зменшується амплітуда нахилів голови та їх інтенсивність. Для бадьорого стану характерна незначне збільшення дисперсії значень величини та її активна зміна. В той же час, для стану дрімоти дисперсія значень значно менша, а абсолютне значення нахилу голови незначним чином вище.

Аналогічні графіки зміни поперечного нахилу голови відносно її попереднього положення зображено на рис.13.



Рис. 13: Залежність поперечного нахилу голови від стану спостерігача

При зміні ФСО спостерігається зміна динаміки та амплітуди поперечного нахилу голови (рис.13). Загалом, значення поперечного нахилу голови корелює зі значенням повздовжнього нахилу при зміні ФСО (рис.12). Збільшення амплітуди нахилу голови пояснюється зменшенням тону м'язів під час засинання оператора. Окрім того, присутні різкі зміни нахилу, що описують різкі рухи, що характерні для людини під час засинання. Головною відмінністю значень, що описують поперечний нахил голови від аналогічних значень для повздовжнього нахилу - наявність патерну для будь-якого ФСО. Так, незважаючи на незначні переміщення голови, при менш детальному масштабі спостерігається повільна значна зміна нахилу голови. Тобто, на всьому проміжку часу, під час якого велось спостереження, відбувалась поступова повільна зміна поперечного нахилу голови.

Таким чином зміна амплітуди та інтенсивності нахилу голови є однією з ознак зміни ФСО. При цьому, незалежно розглядати дану групу характеристик не має сенсу, адже їх значення можуть свідчити лише про зміну положення голови у просторі, що пов'язана з поведінкою людини.

## Губи

Однією з характеристик, що описують функціональний стан оператора є міміка його обличчя. Найбільш впливовою частиною голови, що формує загальну

міміку обличчя є рот, а саме - губи. Так, аналогічно до інших частин обличчя, під час засинання зменшується амплітуда та дисперсія значень ознаки. Очевидною характеристикою роту є площа частини, що обмежена губами. Таким чином можливо розпізнати усвідомлені рухи: різноманітні емоції, позіхання, розмову. Зменшення інтенсивності таких рухів може свідчити про загальне зменшення фізичної активності оператора, що може свідчити про початок засинання.

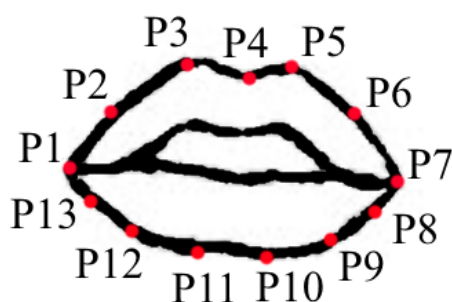


Рис. 14: Ключові точки роту

З використанням ключових точок, зображених на рис.14 та формули, що базується на формулі обчислення  $EAR$  для ока, отримаємо формулу для обчислення чисельної характеристики роту  $MAR$  – *MouthAspectRatio*:

$$MAR = \frac{||p_2 - p_{13}|| + ||p_3 - p_2|| + ||p_5 - p_9|| + ||p_6 - p_8||}{4 * ||p_1 - p_7||}$$

З метою аналізу припущення про залежність ФСО від площі роту проведемо аналіз навчальної вибірки. Для кожного фрейму обчислено значення  $MAR$ , а отримані результати розбито на дві групи, залежно від ФСО. Отримані дані представлено на рис.15.

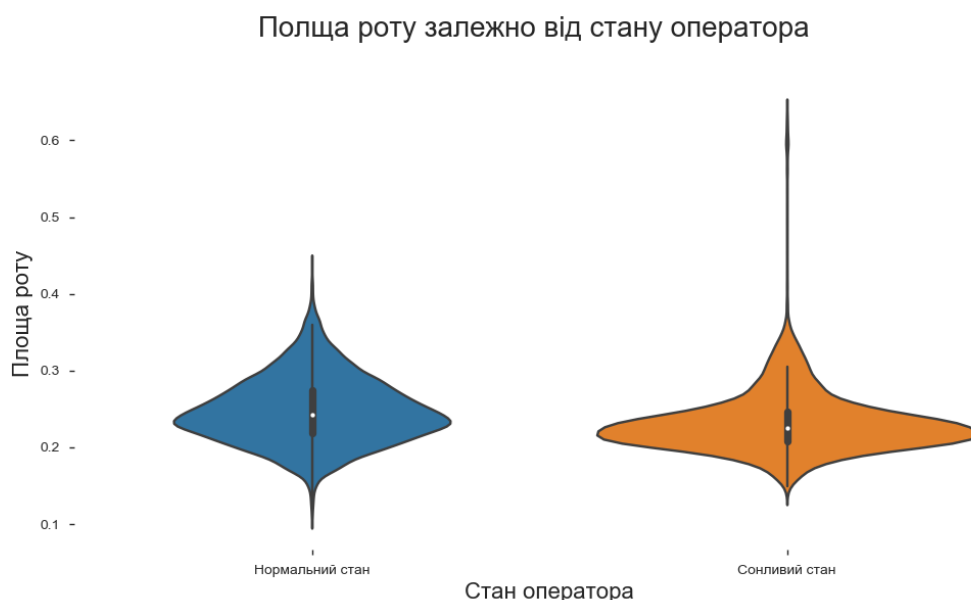


Рис. 15: Залежність площі роту від ФСО

З рис.15 випливає, що загалом характер зміни ознаки не корелює значним чином з ФСО. Для обох функціональних станів значення величини відповідає нормальному розподілу, а викиди відсутні. Відсутність точних значень параметру *MAR* може бути спричинена низькою якістю вхідного зображення. Крім того, одним з можливих факторів, що впливають на даний параметр є фізична активність, що в сонливому стані заміщується іншою з близьким значенням параметру *MAR*. Незважаючи на незначні відмінності і графічному представлені заданої величини, даного факту недостатньо для використання величини *MAR* в якості ознаки розпізнавання.

Таким чином, проаналізувавши характеристики оператора, що можливо отримати з візуальної інформації - доцільно побудувати навчальну вибірку, що містить дані про очі. Всі інші візуальні ознаки недоречно використовувати через їх низьку інформативність.

## 3 Реалізація методів визначення функціонального стану оператора

З метою досягнення максимальної достовірності класифікації функціонального стану оператора, використовуються алгоритми унарної та багатокласової класифікації і їх ансамблі.

Для навчання класифікатору, основною задачею якого є визначення функціонального стану оператора, необхідно попередньо побудувати навчальну вибірку. Формування навчальної вибірки виконується за допомогою унарного класифікатору для пошуку аномалій у вхідних даних. Для реалізації даної підзадачі використовуються наступні алгоритми: класифікатор, що базується на міжквартильній відстані, система правил, метод опорних векторів, класифікатор, що базується на локальному рівні викиду, інтелектуальна екстремальна інформаційна технологія та їх ансамбль.

Після досягнення необхідного об'єму навчальної вибірки, виконується побудова класифікатору для визначення ФСО. Для побудови класифікатору з максимальною достовірністю класифікації дослідимо можливість та ефективність використання наступних методів: інтелектуальна екстремальна інформаційна технологія, система правил, дерева прийняття рішень, нейронні мережі.

Для отримання інформації про положення обличчя у вхідному відеопотоці, використовується алгоритм Віоли-Джонса.

### 3.1 Математична модель методів визначення функціонального стану оператора

При визначенні функціонального стану оператора використовуються алгоритми: інтелектуальна екстремальна інформаційна технологія та система правил, для яких необхідна адаптація для розв'язання даної задачі. Наведемо математичну модель та проведемо асимптотичну оцінку кожного з побудованих алгоритмів.



## Інтелектуальна екстремальна інформаційна технологія

Вхідними даними для ІЕІТ[118] є множина класів розпізнавання навчальної вибірки  $X$ , що розбита на  $N$  функціональних станів (класів розпізнавання), кожен з яких представлений  $n^{(k)}$  реалізаціями, які описуються  $M$  ознаками, де  $V_i^{(k)}$  –  $i$ -та реалізація  $k$ -го класу. Кожен клас  $k$  характеризується матрицею реалізацій  $X^{(k)}$  та найближчим класом для нього  $d^{(k)}$ , що знаходиться на відстані  $P^{(k)}$ .

Головною метою ІЕІТ є побудова контейнерів, що певним чином визначають класи розпізнавання. Контейнер для  $k$ -го класу описується радіусом  $R^{(k)}$  та еталонним вектором (центром гіперсфери)  $Ev^{(k)}$ . Проміжним кроком для обчислення характеристик контейнеру є обчислення бінарної матриці для  $k$ -го класу  $B^{(k)}$  за допомогою вектору середніх значень ознак розпізнавання  $Av^{(k)}$  та ширини інтервалу допусків для ознак  $\gamma$ . За побудованою бінарною матрицею отримується вектор середніх значень  $Ab^{(k)}$  та рівнем селекції  $\rho$  будується еталонний вектор.

Розглянемо наступні статистичні характеристики, що використовуються в ІЕІТ:

$K_1^{(k)}$  – кількість своїх реалізацій, що потрапили в контейнер.

$K_2^{(k)}$  – кількість своїх реалізацій, що не потрапили в контейнер.

$K_3^{(k)}$  – кількість чужих реалізацій, що потрапили в контейнер.

$K_4^{(k)}$  – кількість чужих реалізацій, що не потрапили в контейнер.

$D_1^{(k)}$  – перша достовірність, або імовірність того, що свої реалізації будуть розпізнані як свої (True Positive Rate).

$\alpha^{(k)}$  – помилка першого роду, або імовірність того, що свої реалізації не будуть розпізнані як свої (False Negative Rate).

$\beta^{(k)}$  – помилка другого роду, або імовірність того, що чужі реалізації будуть розпізнані як свої (False Positive Rate).

$D_2^{(k)}$  – друга достовірність, або імовірність того, що чужі реалізації будуть розпізнані як чужі (True Negative Rate).

Реалізація контейнеру являє собою вектор розміру  $n^{(k)}$ ,  $E$  – вектор виду  $1 \times n^{(k)}$  ( $E = 1, \dots, 1$ ). Введемо наступні функції:

$Belongs(V, R^{(k)}, Ev^{(k)})$  – функція належності реалізації  $V$  до контейнеру радіусом  $R^{(k)}$ , з центром  $Ev^{(k)}$ .

$dist(A, B)$  – відстань між векторами  $A$  та  $B$ .

$Exam(V)$  – функція визначення класу, до якого належить вектор  $V$ .

Ключові терміни та поняття, що використовуються в алгоритмі ІЕІТ зображено на рис.16.

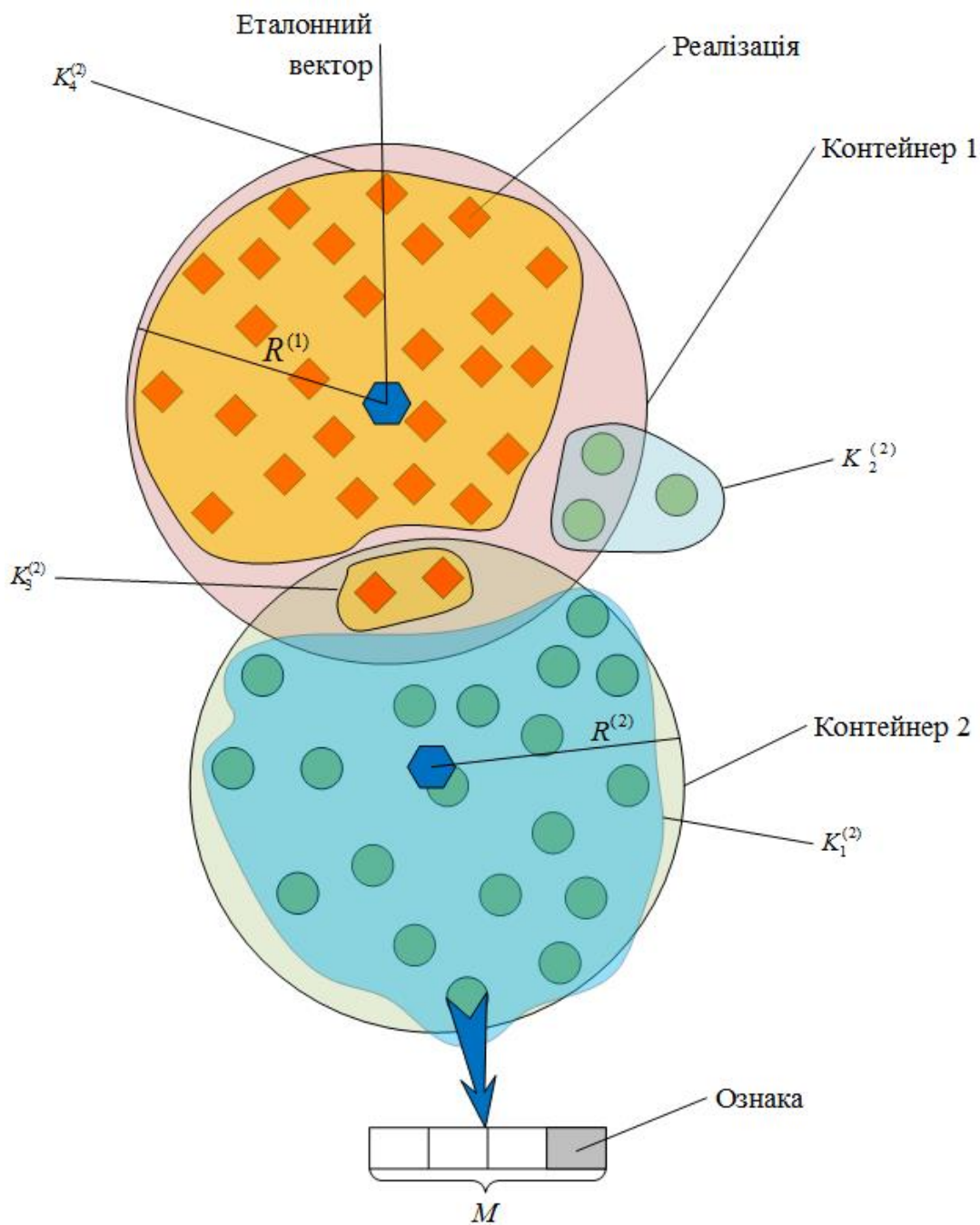


Рис. 16: Візуалізація базових понять ІЕІТ.

Для реалізації алгоритму ІЕІТ проведемо імплементацію розглянутих функцій та операторів перетворення простору ознак розпізнавання.

Відстань між двома векторами в бінарному просторі

$$dist(A, B) = \sum_{i=1}^M xor(A_i, B_i) \quad (3.1)$$

$$\text{xor}(a, b) = \begin{cases} 1, a = b \\ 0, a \neq b \end{cases}$$

Функція належності вектору до контейнеру

$$\text{Belongs}(V, R^{(k)}, Ev^{(k)}) = \begin{cases} 1, \text{dist}(V, Ev^{(k)}) \leq R^{(k)} \\ 0, \text{dist}(V, Ev^{(k)}) > R^{(k)} \end{cases} \quad (3.2)$$

Середні значення ознак

$$Av^{(k)} = \frac{1}{n^{(k)}} EX^{(k)} \quad (3.3)$$

Контрольні допуски для ознак  $k$ -го класу

$$\begin{cases} Ub = Av^{(k)} + \gamma E \\ Lb = Av^{(k)} - \gamma E \end{cases} \quad (3.4)$$

Бінеаризація матриці  $k$ -го класу

$$B_{i,j}^{(k)} = \begin{cases} 1, X_{i,j}^{(k)} \in [Lb_j, Ub_j] \\ 0, X_{i,j}^{(k)} \notin [Lb_j, Ub_j] \end{cases} \quad (3.5)$$

Еталонний вектор  $k$ -го класу

$$Ev^{(k)} = \begin{cases} 1, Ab^{(k)} \geq \rho \\ 0, Ab^{(k)} < \rho \end{cases} \quad (3.6)$$

Точнісні характеристики СППР для  $k$ -го класу

$$\begin{cases} K_1^{(k)} = \sum_{i=1}^{n^{(k)}} \text{Belongs}(V_i^{(k)}, R^{(k)}, Ev^{(k)}) \\ K_2^{(k)} = n^{(k)} - K_1^{(k)} \\ K_3^{(k)} = \sum_{i=1}^{n^{(d^{(k)})}} \text{Belongs}(V_i^{(d^{(k)})}, R^{(k)}, Ev^{(k)}) \\ K_4^{(k)} = n^{(d^{(k)})} - K_3^{(k)} \end{cases} \quad (3.7)$$

Достовірності прийняття рішення СППР для  $k$ -го класу

$$\begin{cases} D_1^{(k)} = \frac{K_1^{(k)}}{n^{(k)}} \\ \alpha^{(k)} = \frac{K_2^{(k)}}{n^{(k)}} \\ \beta^{(k)} = \frac{K_3^{(k)}}{n^{(k)}} \\ D_2^{(k)} = \frac{K_4^{(k)}}{n^{(k)}} \end{cases} \quad (3.8)$$

Робоча область

$$D_1^{(k)} \in [0.5, 1] \wedge D_2^{(k)} \in [0.5, 1] \quad (3.9)$$

Робоча формула на базі ентропійної міри Шеннона

$$\begin{aligned} E = 1 + \frac{1}{2} \left( \frac{\alpha^{(k)}}{\alpha^{(k)} + D_2^{(k)}} \log_2 \frac{\alpha^{(k)}}{\alpha^{(k)} + D_2^{(k)}} + \frac{D_1^{(k)}}{\beta^{(k)} + D_1^{(k)}} \log_2 \frac{D_1^{(k)}}{\beta^{(k)} + D_1^{(k)}} \right. \\ \left. + \frac{\beta^{(k)}}{\beta^{(k)} + D_1^{(k)}} \log_2 \frac{\beta^{(k)}}{\beta^{(k)} + D_1^{(k)}} \right. \\ \left. + \frac{D_2^{(k)}}{\alpha^{(k)} + D_2^{(k)}} \log_2 \frac{D_2^{(k)}}{\alpha^{(k)} + D_2^{(k)}} \right) \end{aligned} \quad (3.10)$$

Робоча формула на базі інформаційної міри Кульбака

$$\begin{aligned} E = \frac{1}{2} \log_2 \left( \frac{D_1^{(k)} + D_2^{(k)}}{\alpha^{(k)} + \beta^{(k)}} \right) \left( (D_1^{(k)} + D_2^{(k)}) - (\alpha^{(k)} + \beta^{(k)}) \right) = \\ \log_2 \left( \frac{2 - \alpha^{(k)} - \beta^{(k)}}{\alpha^{(k)} + \beta^{(k)}} \right) (1 - \alpha^{(k)} - \beta^{(k)}) \end{aligned} \quad (3.11)$$

Одним із параметрів функціонування СППР побудованого в рамках ІЕІТ є радіус контейнеру класу розпізнавання, тоді ітераційна процедура визначення радіусу контейнеру набуває вигляду 3.12.

Обчислення радіусу контейнера  $k$ -го класу

$$R^{(k)} = \min_{1 \leq r \leq P^{(k)}} \left\{ r : E \left( K_1^{(k)}, K_2^{(k)}, K_3^{(k)}, K_4^{(k)} \right) \rightarrow \max, \right. \\ \left. D_1^{(k)} \in [0.5, 1] \wedge D_2^{(k)} \in [0.5, 1] \right\} \quad (3.12)$$

Очевидно, що точнісні характеристики  $K_i^{(k)}$  залежать від радіусу контейнера, що в свою чергу залежить від простору, що створюється за рахунок перетворень (формули 3.4-3.6).

Після обчислення зазначених параметрів проведемо навчання системи за наступною

$$(\forall k = \overline{1, N}) R^{(k)} = \min_{1 \leq r \leq P^{(k)}} \left\{ r : E \left( K_1^{(k)}, K_2^{(k)}, K_3^{(k)}, K_4^{(k)} \right) \rightarrow \max, \right. \\ \left. D_1^{(k)} \in [0.5, 1] \wedge D_2^{(k)} \in [0.5, 1] \right\} \quad (3.13)$$

Алгоритм екзамену СППР

$$Exam(V) = \min_{m=1..N} (m : Belongs(V, R^{(m)}, Ev^{(m)})) \quad (3.14)$$

Проведемо асимптотичний аналіз алгоритму з метою оцінки ресурсів, що необхідні для його роботи. Будемо брати до уваги час роботи алгоритму та задіяну пам'ять, які залежать від розміру вхідних даних. Весь процес роботи системи класифікації складається з трьох етапів: попередня підготовка даних - виконується одноразово для вибірки до початку всіх інших стадій, етап навчання - виконується одноразово після підготовки даних, етап екзамену - може виконуватися безліч разів на системі, що пройшла через перші два етапи. Головною вимогою до системи є її швидкодія на етапі екзамену, бо саме цей етап виконується найбільшу кількість разів. В той же час, система, що довго навчається не може бути швидко перенавчена в режимі експлуатації. Для ІЕІТ отримані наступні асимптотичні оцінки швидкодії.

### Попередня підготовка даних

1. Обчислення середнього значення елементів матриці (3.3):  $O(n^{(k)} * M)$
2. Обчислення контрольних допусків (3.4):  $O(n^{(k)})$
3. Бінеаризація матриці (3.5):  $O(n^{(k)} * M)$
4. Обчислення еталонного вектору (3.6):  $O(M)$
5. Загальна асимптотика (формули 3.3–3.6):  $O(n^{(k)} * M) + O(n^{(k)}) + O(M) = M)$

### Алгоритм навчання

1. Обчислення точнісних характеристик класу (3.7):
  - (а)  $K_1^{(k)} : O(M)$
  - (б)  $K_2^{(k)} : O(1)$
  - (в)  $K_3^{(k)} : O(M)$
  - (г)  $K_4^{(k)} : O(1)$
2. Перевірка належності реалізації до контейнера (3.2):  $O(M)$
3. Відстань між векторами:  $O(M)$
4. Обчислення радіусу контейнера (3.12):  $O(P^{(k)} * M) + O(P^{(k)})$
5. Загальна асимптотика (3.13):
 
$$O(N * n^{(k)} * M) + O(N * n^{(k)}) + O(N * M) + O(N * P^{(k)} * M) + O(N * P^{(k)}) = O(N * n^{(k)} * M) + O(N * P^{(k)} * M)$$

### Алгоритм екзамену

1. Загальна асимптотика алгоритму екзамену(3.14):  $O(N * M)$

### Система правил

Вхідні дані для алгоритму розділені на  $N$  станів, кожен з яких представлений  $n^{(k)}$  реалізаціями, що описуються  $M$  ознаками. При навчанні системи розпізна-

вати клас  $k$ , за відомим вектором значень  $i$ -ї ознаки  $F_i^{(k)}$  будуються інтервали допусків (3.15), (3.16), що характеризуються мінімальним значенням ознаки  $Lb_i^{(k)}$  та максимальним  $Ub_i^{(k)}$ . Процес екзамену (3.18) полягає в знаходженні класу, сума відстаней (3.17) від ознаки  $V_i$  екзаменаційного об'єкту до інтервалу  $[Lb_i^{(k)}, Ub_i^{(k)}]$ , буде мінімальною. Зазначимо, що в процесі прийняття рішення необхідно проводити нормалізацію інтервалів, наприклад, за середнім значенням ознаки  $Mb_i^{(k)}$ .

Визначення інтервалів

$$Lb_i^{(k)} = \min_{i=1..M} (F_i^{(k)}) \quad (3.15)$$

$$Ub_i^{(k)} = \max_{i=1..M} (F_i^{(k)}) \quad (3.16)$$

Відстань до класу

$$D^{(k)}(V) = \sum_{i=1}^M \begin{cases} 0, & Lb_i^{(k)} \leq V_i \leq Ub_i^{(k)} \\ \min\left(\left|\frac{Lb_i^{(k)} - V_i}{Mb_i^{(k)}}\right|, \left|\frac{Ub_i^{(k)} - V_i}{Mb_i^{(k)}}\right|\right), & otherwise \end{cases} \quad (3.17)$$

Етап прийняття рішення

$$Exam(V) = \min_{m=1..N} (m : D^{(m)}(V)) \quad (3.18)$$

Асимптотична оцінка

Алгоритм навчання:  $O(N * M)$

Алгоритм екзамену:  $O(N * M)$

## 3.2 Оцінка достовірності розпізнавання класифікатору

Для оцінки достовірності розпізнавання алгоритмів класифікації використовуються такі метрики як precision та recall[119], що залежать від величин True Positive(TP), False Positive(FP) та False Negative(FN):

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$



$$precision = \frac{true\ positives}{true\ positives + false\ positives}, \text{ де}$$

*true positives* та *false negatives* - кількість правильно класифікованих об'єктів, що належать та не належать базовому класу; *false positives* - помилка першого роду, що полягає у внесенню в клас об'єкту, що йому не належить. Повнота (*recall*) - відношення кількості правильно класифікованих об'єктів відносно загальної їх кількості. Точність (*precision*) - відношення кількості правильно класифікованих об'єктів відносно кількості об'єктів, які віднесено до даного класу. З використанням метрик *precision* та *recall* можливо виразити метрику *f1 - score*[120] для оцінки достовірності розпізнавання алгоритму:

$$f1 - score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Метрика *f1-score* є гармонійним середнім між метриками *precision* та *recall*, що набуває значень  $[0; 1]$  та прямо пропорційно залежить від достовірності класифікації. Для оцінки достовірності розпізнавання класифікатору незалежно від розбиття вхідної вибірки використовується крос валідація з 5 фолдами. Для розбиття вибірки обрана стратегія, при якій в кожному наборі тестових даних зберігається розподіл кількості елементів між класами.

### 3.3 Виявлення блимання

Очевидно, що під час процесу блимання відбувається поступове зменшення площі видимої частини ока та EAR, знаходження ока в закритому стані під час якого значення мінімальне та поступове відкриття ока. Під блиманням розуміється проміжок часу, під час якого око проходить описані вище стадії. Локалізація інтервалів блимання виконується за допомогою детектору, що базується на зміні характеристик ока під час процесу блимання. Так як, абсолютне значення обох величин прямо пропорційно залежить від ступені відкриття ока, то алгоритми обробки та їх аналізу ідентичні.

В якості вхідних даних використовується експертно розмічена вибірка, що описує значення основних параметрів ока під час спостереження. В якості вхідних даних для класифікаторів використано площу видимої частини склери та EAR. Інтервали, отримані за допомогою класифікатора визначено правильно, якщо площа їх перетину з еталонним інтервалом більша за 50%.

В якості вхідних даних використовується вибірка `eyeblink8`[121]. Для застосування класифікаторів до даної вибірки, побудовано множини даних  $X$  та  $EAR$ , що відповідають видимій площі ока та характеристиці EAR, відповідно.

Для обчислення характеристик блимання використовується інтерполяція величини за допомогою кубічного сплайну[122], що представляє собою поліном виду:

$$f(x) = a + b(x - x_0) + c(x - x_0)^2 + d(x - x_0)^3,$$

де  $a, b, c, d$  - коефіцієнти поліному,  $x_0$  - ліва границя інтервалу інтерполяції. Для побудови сплайну використовується інформація про два суміжні інтервали  $(x_0; x_1)$  та  $(x_1; x_2)$ , для кожного з яких побудовано сплайни  $f_0(x)$  та  $f_1(x)$  відповідно:

$$\begin{cases} f_0(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3 \\ f_1(x) = a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3 \end{cases}$$

Також, в процесі інтерполяції необхідні значення першої та другої похідних даного поліному:

$$f'(x) = b + 2c(x - x_0) + 3d(x - x_0)^2$$

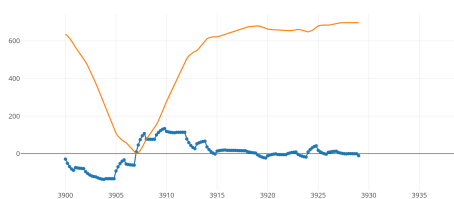
$$f''(x) = 2c + 6d(x - x_0)$$

Шляхом елементарних перетворень отримано наступні співвідношення невідомих

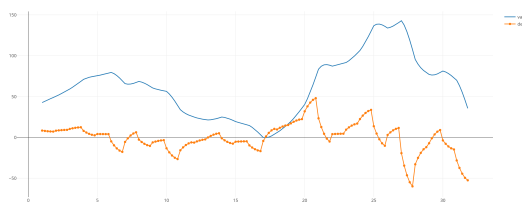
коефіцієнтів:

$$\left\{ \begin{array}{l} a_0 = y_0 \\ a_1 = y_1 \\ c_0 = 0 \\ b_0(x_1 - x_0) + d_0(x_1 - x_0)^3 = y_1 - y_0 \\ b_1(x_2 - x_1) + c_1(x_2 - x_1)^2 + d_1(x_2 - x_1)^3 = y_2 - y_1 \\ b_0 + 3d_0(x_1 - x_0)^2 = b_1 \\ 6d_0(x_1 - x_0) = 2c_1 \\ 2c_1 + 6d_1(x_2 - x_1) = 0 \end{array} \right.$$

Ключовою характеристикою блимання при визначенні поточного стану оператора є тривалість перебування ока в кожній стадії. Для визначення даної характеристики використовується інформація про характер зміни площі ока під час спостереження. Допоміжною характеристикою при визначенні інтервалів спадання та зростання величини є перша похідна, що обчислена для кубічного сплайну на досліджуваному відрізку. Так, при зростанні величини, перша похідна набуває додатніх значень, а при спаданні - від'ємних. Завдяки цьому можливо уникнути похибок, викликаних шумами у вхідних даних.



(а) нормального стану.



(б) сонливого стану.

Рис. 17: Графік і перша похідна для

На рис.17 спостерігається, що для сонливого стану характерна інтенсивна зміна значень першої похідної. Це спровоковано фізіологічними особливостями організму в стані засинання. Таким чином, наявність інтенсивних змін значень першої похідної можливо використовувати в якості однієї з ознак засинання оператора.

Окрім того, простежується взаємозв'язок між першою похідною і стадіями блимання: для закриття ока характерне неперервний інтервал з від'ємними значеннями першої похідної, а для відкриття - з додатними, ці інтервали перетинаються в точці перегину, яка одночасно свідчить про максимально закрите око.

### **Адаптивний класифікатор, що базується на мінімальному значенні ознаки**

Найпростішим з алгоритмів визначення інтервалів, що описують блимання є пошук фрагментів послідовності, значення елементів яких менше заздалегідь визначеного порогового значення  $Th$ :

$$X_i \leq Th$$

Емпіричним шляхом у роботі встановлено, що блимання при нормальному стані оператора триває не менше ніж 20 фреймів, тому результуючий інтервал представляю собою серію з 20 значень, з центром в точці, де досягається мінімум.

Так як, при класифікації використовуються абсолютні значення величини, то даний підхід є чутливим до якості вхідного зображення, умов освітлення, позиції обличчя на зображенні та його повороту. Точка мінімуму є моментом максимального ступеню закритості ока. Період часу, що передує цьому моменту є інтервалом закриття ока, а період, що слідує після мінімуму - інтервал відкриття.

Незалежно від вибору ознаки для використання в якості критерію при визначенні поточного стану ока, можливі колізії внаслідок перетину множин мінімальних значень ознаки  $MN$ . При використанні в якості порогового значення  $Th = MN + \Delta$ , де  $\Delta$  - значення обране емпіричним шляхом, площа перетину збільшується, що значним чином зменшує достовірність класифікації. З метою підвищення достовірності розпізнавання значення  $MN$  коригується під час обробки даних з урахуванням поточних умов. Оптимальною стратегією зміни значення  $MN$ , що враховує ФСО та умови зйомки є оновлення значення за наступною

формулою:

$$MN = \min_{i=0,5000} X_i,$$

що відповідає 4 хвилинам спостереження. Значення критерію f1-score, залежно від значення  $\Delta$  зображено на рис.18.

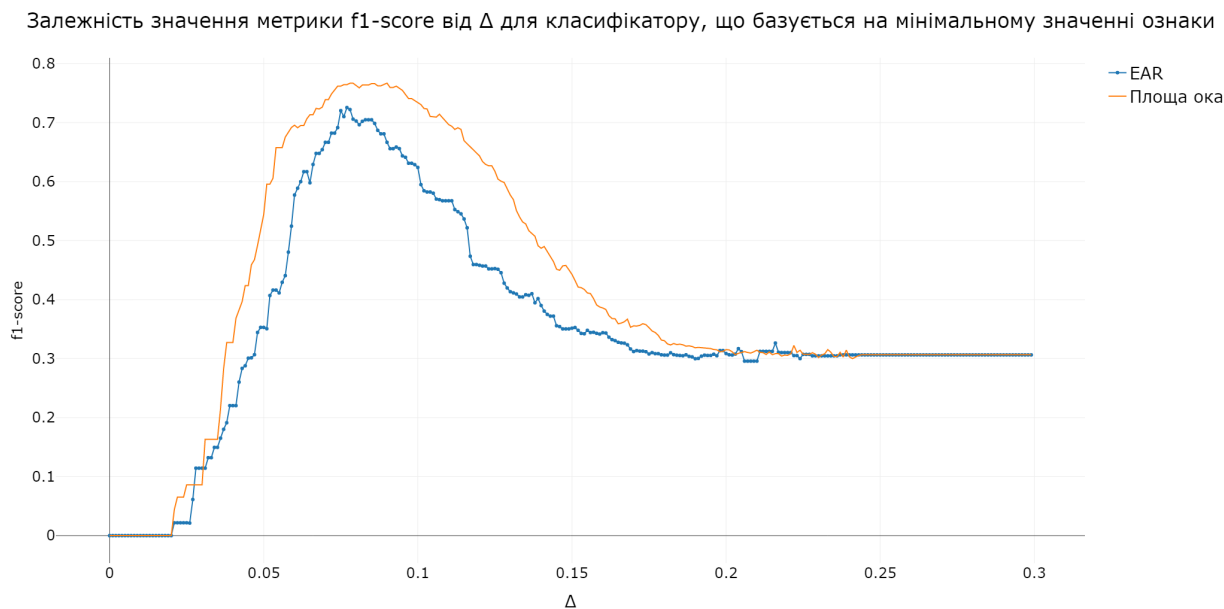


Рис. 18: Залежність значення f1-score від  $\Delta$  для класифікатору, що базується на мінімальному значенні ознаки

З метою приведення значень до єдиного масштабу, значення  $\Delta$  для класифікатору, що базується на площі ока, розділено на 1000. Максимальне значення метрики f1-score = 0.725 досягається при  $\Delta = 0.077$  класифікатором, що побудовано з використанням значення ознаки EAR. При використанні класифікатору, що побудовано з використанням інформації про видиму площу ока, максимальне значення метрики f1-score дорівнює 0.767 та досягається при значенні  $\Delta = 79$ .

При цьому, отримані наступні статистичні характеристики побудованих класифікаторів, що наведено в табл.5.

			Результат класифікатору			
			Площа ока		EAR	
			blinks	timeline	blinks	timeline
			82	-	99	-
Вхідне значення	blinks	87	74	13	78	9
	timeline	-	8	-	21	-

Табл. 5: Статистичні характеристики адаптивного класифікатору, що базується на мінімальному значенні ознаки

З табл.5 зазначимо, що всього у вибірці присутні 87 інтервалів, що містять блимання. При використанні класифікатору, що побудовано з використанням в якості навчальної вибірки інформації про видиму площу ока, правильно класифіковано 85% інтервалів. При цьому, 15% інтервалів блимання не знайдено взагалі, а 10% знайдених інтервалів є хибними спрацюваннями. Таким чином, отримано класифікатор з наступними значеннями метрик:  $Precision = 0.902$ ,  $Recall = 0.851$ .

При побудові класифікатору, з використанням в якості вхідних даних інформації про характеристику EAR, правильно класифіковано 90% інтервалів, а 10% інтервалів відкинуто як хибні. В той же час, на відміну від класифікатору, що базується на площі ока, кількість хибних спрацювань складає 21%. Класифікатор, побудований з використанням параметру EAR має близькі значення параметрів  $Precision$  та  $Recall$  (0.897 та 0.788 відповідно). Більшість інтервалів, що містять блимання були пропущені, про що свідчить низьке значення параметру  $Recall$ . При використанні характеристики EAR кількість хибних спрацювань значно більша за аналогічний показник для класифікатору, що побудований з використанням в якості навчальної вибірки, видимої площі ока.

Таким чином, оптимальними вхідними даними є видима площа ока. Даний класифікатор має достатні статистичні характеристики для його використання з метою виявлення блимань у відеопотоці.

## Класифікатор, що базується на динаміці зміни ознаки

Характерною ознакою блимання є проходження оком трьох стадій: закриття, перебування в закритому стані та відкриття. При цьому, час перебування ока в закритому стані незначний, а через помилки при локалізації очей даний стан визначається однією точкою. Очевидно, що величини EAR та видима площа ока залежать від поточного стану ока. При закритті ока дані величини зменшуються, а при відкритті - збільшуються. Таким чином, задача визначення інтервалу зводиться до пошуку послідовних інтервалів спадання та зростання ознаки, з точкою мінімуму між ними.

Пошук часових проміжків, під час яких відбулося блимання реалізується за допомогою детектору, який для заданої множини значень знаходить інтервали, в яких відбувається монотонне спадання та монотонне зростання величини. Нехай, в інтервалі часу  $[a; b]$  відбувається процес блимання з максимальним закриттям ока в момент часу  $K$ , тоді для величини  $X_i$  в момент часу  $i$  характерна наступна поведінка:

$$\begin{cases} X_i \geq X_{i+1}, \forall i \in [a; K] \\ X_i \leq X_{i+1}, \forall i \in [K; b] \end{cases}$$

Головним недоліком даної моделі є її чутливість до будь-яких інтервалів, які відповідають патерну. Це значним чином впливає на кількість помилок першого роду, окрім того, на достовірність розпізнавання впливають шуми, що локалізовані на інтервалах спадання та зростання. Завдяки використанню рухомого середнього, значення величини згладжується, а вплив шумів мінімізується.

За допомогою фільтрації результатів з урахуванням амплітуди величини площі ока  $S_a$  та EAR зменшується кількість помилок першого роду. В якості інтервалів блимання визнаються лише ті інтервали, для яких виконується умова  $MX - MN > S_a$ , де  $MX$  - максимальне значення величини на інтервалі, а  $MN$  - мінімальне. Залежність метрики f1-score від значення амплітуди  $S_a$  зображено на рис.19.

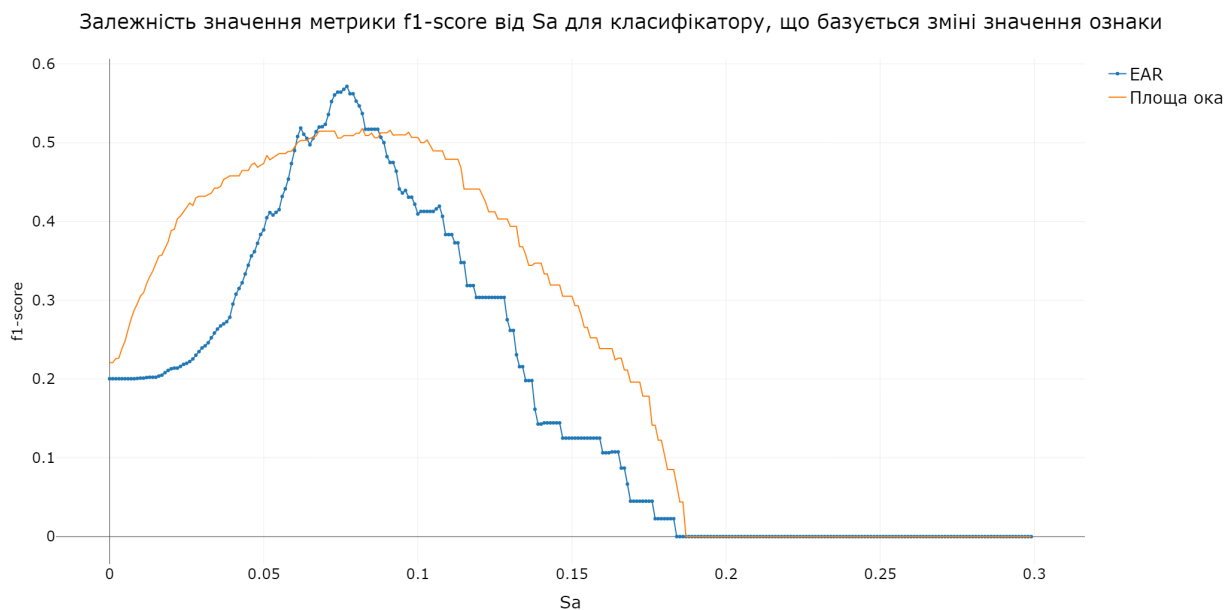


Рис. 19: Залежність значення метрики f1-score від  $S_a$  для класифікатору, що базується зміні значення ознаки.

З метою відображення залежності значення метрики f1-score від параметру  $S_a$  для обох ознак, для класифікатору, що в якості вхідних даних використовує видиму площу ока, значення  $S_a$  розділено на 1000. Максимальне значення f1-score = 0.514 досягається при значенні  $S_a = 73$  при використанні в якості вхідних даних інформації про зміну площі ока. При використанні в якості ознаки величини EAR значення f1-score = 0.571 при значенні EAR = 0.077. Детальні результати класифікації наведено в табл.6.

			Результат класифікатору			
			Площа ока		EAR	
			blinks	timeline	blinks	timeline
			111	-	57	-
Вхідне значення	blinks	87	44	33	44	33
	timeline	-	51	-	33	-

Табл. 6: Статистичні характеристики функціонування класифікатору, що базується на характері зміни ознаки



При використанні даного класифікатору, незалежно від використаної ознаки, досягається низьке значення метрики Recall, що залежить від кількості правильно знайдених об'єктів. Так, при використанні в якості ознаки параметру EAR отримано значення метрики Precision = 0.571, а Recall = 0.571.

Так, при використанні в якості ознаки видимої площі ока правильно класифіковано 50.6% інтервалів. В той же час, 49.4% інтервалів є хибним спрацюванням. Завдяки цьому досягнуто значення метрики Precision = 0.571, а Recall = 0.463. Негативний внесок у достовірність розпізнавання вносить низька кількість знайдених об'єктів. При використанні характеристики EAR отримано показники, аналогічні до класифікатору, вхідні дані для якого базуються на видимій площі ока. Головною відмінністю класифікатору, що базується на ознаці EAR є низька кількість хибних спрацювань. Таким чином, для даного класифікатору метрики Precision та Recall становлять 0.321 та 0.571 відповідно.

Таким чином, незалежно від навчальної вибірки, побудовано класифікатор, що з високою достовірністю відкидає інтервали, що не є блиманням, про що свідчить високе значення метрики Precision. В той же час близько половини інтервалів не було знайдена взагалі. Тобто, даний класифікатор можливо використовувати у поєднанні з класифікатором, що базується на мінімальному значенні ознаки. За допомогою першого класифікатору знаходиться велика кількість інтервалів, що містить значну кількість хибних спрацювань, а за допомогою другого класифікатору отримані результати уточнюються.

### **Класифікатор, що базується на функції непропорційності**

Для пошуку еталонних сигналів  $g_i(x)$ , ( $i = 1, 2, \dots, N$ ) у вхідному сигналі  $f(x)$  використовується функція непропорційності [123, 124, 125]. При цьому, фрагмент вхідного сигналу, що відповідає еталону може бути трансформований шляхом зміщення або масштабування по осі абсцис та ординат. Таким чином, вхідний сигнал набуває виду:

$$f(x) = Ag_i(\alpha x + \beta) + C,$$

де  $A, \alpha$  - масштабні коефіцієнти,  $\beta$  - зміщення по осі абсцис,  $C$  - зміщення по осі ординат.

Так як,  $\frac{dy}{dx}$  є границею відношення  $\frac{\Delta y}{\Delta x}$  при  $\Delta x \rightarrow 0$ , тому, якщо в даній точці існує пропорційний зв'язок між  $x$  та  $y$  в точці  $M(x, y)$ , то має місце наступна числова рівність:

$$\frac{y}{x} = \frac{dy}{dx}$$

Непропорційністю за похідною першого порядку називається різниця між  $\frac{y}{x}$  та  $\frac{dy}{dx}$ :

$$@d_x^{(1)}y = \frac{y}{x} - \frac{dy}{dx}$$

Геометрично непропорційність представляє собою різницю тангенсів двох кутів між додатнім напрямом осі  $Ox$  та двома прямими: пряма, що з'єднує початок координат та точку  $M$  та пряма, що є дотичною до графіку функції  $f(x)$  в точці  $M$  відповідно. Непропорційність за похідною першого порядку для заданого значення  $x$  визначає різницю між двома швидкостями зміни  $y$  від  $x$ : швидкість зміни  $y$  випадку наявності пропорційної залежності між  $y$  та  $x$  та реальна швидкість  $\frac{dy}{dx}$ . За допомогою непропорційності за значенням  $@v_x^{(1)}y$  оцінюється відхилення залежності  $y$  від  $x$  в окремих точках. Непропорційністю за значенням функції  $f(x)$  називається добуток  $x$  та непропорційності за похідною  $@d_x^{(1)}y$ :

$$@v_x^{(1)}y = x * @d_x^{(1)}y = y - x \frac{dy}{dx}$$

Дана непропорційність відображає пропорційну залежність  $y$  та  $x$  в точці  $M$ . З метою аналізу характеру сигналу без урахування абсолютної величини значень сигналу використовується відносна непропорційність. Для отримання відносної непропорційності необхідно значення непропорційності за значенням розділити на  $y$ :

$$@N_x^{(1)}y = 1 - \frac{x}{y} \frac{dy}{dx} \quad (3.19)$$

Дана оцінка є безрозмірною і дорівнює нулю у випадку пропорційної залежності між  $y$  та  $x$ . Якщо функція задана параметрично, то вираз (3.19) набуває

виду:

$$@N_{x(t)}^{(1)}y(t) = 1 - \frac{x(t) y'_t(t)}{y(t) x'_t(t)} \quad (3.20)$$

Так як, в досліджуваній функції наявний масштабний коефіцієнт  $\alpha$ , то для вирішення даної задачі необхідно використовувати відносну непропорційність першого порядку за її першою похідною:

$$@N_{x(t)}^{(1)}y(t) = 1 - \frac{f'(x) f'(x)}{f(x) f''(x)} \quad (3.21)$$

Необхідною умовою застосування даного виду відносної непропорційності є існування першої та другої похідної функції. Для цього, початкова функція інтерполюється за допомогою кубічного сплайну. Внаслідок застосування відносної функції непропорційності до фрагментів графіку, що містять блимання отримано патерн, який описує кожний інтервал(рис.20).

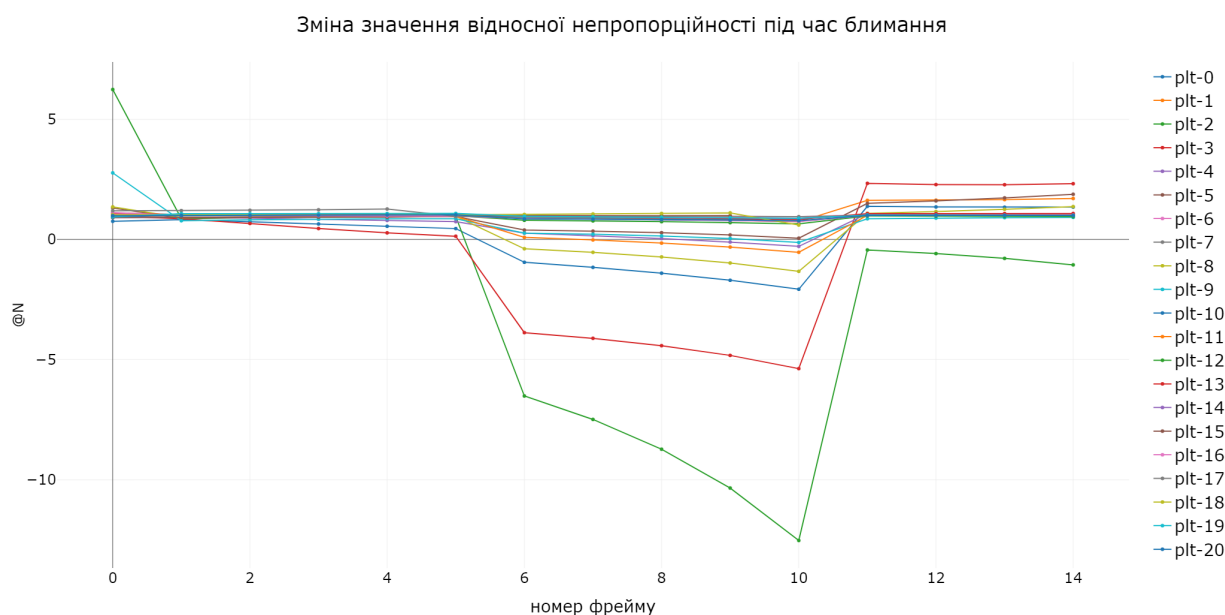


Рис. 20: Зміна значення відносної непропорційності під час блимання при використанні в якості вхідних даних видимої площі ока

При трансформації вхідних даних з використанням функції непропорційності отримано патерн, що описує блимання. При використанні даного патерну в якості ознаки можливо з високою достовірністю класифікувати інтервали блимання. Недоліком даного підходу є велика кількість хибних спрацювань через похибки при

інтерполяції та наявність інтервалів, що за характером зміни величини схожі на блимання. При використанні в якості ознаки характеристики EAR отримано аналогічний патерн, тому побудова класифікатору, що побудовано з використанням в якості вхідних даних EAR не є доцільним.

Статистичні характеристики функціонування побудованого класифікатору, який в якості використовує видиму площу ока, наведено в табл.7.

			Результат класифікатору	
			blinks	timeline
			712	-
Вхідне значення	blinks	87	73	14
	timelinel	-	625	-

Табл. 7: Статистичні характеристики класифікатору, що базується на функції непропорційності

При використанні класифікатору, що базується на функції непропорційності, який побудовано на вхідних даних, що в якості ознаки використовують видиму площу ока правильно класифіковано близько 84% вхідних даних. В той же час, велика загальна кількість хибних спрацювань, що становить 87.7% від загальної кількості спрацювань. Отриманий класифікатор має наступні показники якості розпізнавання: Precision = 0.839 та Recall = 0.105.

### Побудова ансамблю класифікаторів

Проаналізувавши результати роботи описаних вище класифікаторів робимо висновок, що кожен з них має сильні та слабкі сторони. При цьому, очевидно, що жоден з класифікаторів не може використовуватися самостійно. Так, для класифікатори, що побудовані з використанням функції непропорційності та мінімального значення ознаки знаходять більшість інтервалів, що містять блимання. В той же час для них характерна велика кількість хибних спрацювань. Для класифікатору,

що аналізує характер зміни величини ознаки характерна низька кількість знайдених інтервалів та мала кількість хибних спрацювань (табл. 5–7). Статистичні характеристики побудованих класифікаторів наведено в табл.8.

Класифікатор	Precision	Recall	f1-score
Мінімальне значення	0.902	0.851	0.767
Характер зміни ознаки	0.571	0.571	0.571
Фунція непропорційності	0.839	0.105	0.186

Табл. 8: Значення метрик Precision та Recall для побудованих класифікаторів

Таким чином, жоден з розглянутих класифікаторів не досягає достовірності розпізнавання близької до 1 (табл.8). Для підвищення достовірності класифікації доцільно об'єднати дані класифікатори в ансамбль.

З метою збільшення достовірності розпізнавання побудовано ансамбль, що складається з трьох, описаних вище класифікаторів. Клас, до якого належить інтервал визначається ізольовано кожним з класифікаторів. Наступним кроком, за допомогою простої більшості голосів відбувається класифікація стану оператора, якому відповідає даний інтервал. Якщо за вердикт ансамблю приймати рішення лише одного класифікатору, то це відповідає використанню єдиного класифікатору. Як наслідок, присутня значна кількість хибних спрацювань. Якщо, даний інтервал визначено як блимання не менш як двома класифікаторами, то приймаємо, що даний стан відповідає блимання. В іншому випадку даний інтервал визначається як хибне спрацювання. Таким чином вдалося досягти низької кількості хибних спрацювань та високої достовірності розпізнавання. В табл.10 наведено дані про достовірність розпізнавання ансамблю класифікаторів при ізольованому використанні кожної пари класифікаторів.

Базова функція	Мінімальне значення	Функція непропорційності	Динаміка зміни ознаки
Мінімальне значення	-	28.7%	0%
Функція непропорційності	28.7%	-	18.4%
Динаміка зміни ознаки	0%	18.4%	-

Табл. 10: Кількість правильно визначених інтервалів при попарному об'єднання класифікаторів

При використанні ансамблю 3% блимань не було знайдено взагалі, а 30% інтервалів правильно визнано блиманнями одночасно трьома класифікаторами. В той же час, 19.6% інтервалів, що містять блимання правильно знайдено лише класифікатором, що базується на функції непропорційності. Збільшення внеску класифікаторів, що базуються на мінімальному значенні ознаки та динаміці її зміни жодним чином не впливає на зміну достовірності розпізнавання. Збільшення ваги голосу класифікатору, що базується на функції непропорційності призводить до значного росту кількості хибних спрацювань. Таким чином, використання зваженого голосування в якості вирішального правила не впливає на достовірність класифікації.

Ансамблем, що складається з трьох класифікаторів правильно визначено 80.4% інтервалів, а кількість хибних спрацювань становить 4.7%. Отриманий ансамбль має високу достовірність розпізнавання, про що значить наступні значення метрик: Precision = 0.805, Recall = 0.959 f1-score = 0.875. Високі значення метрик Precision та Recall свідчать про можливість використання отриманого класифікатору в системах, що функціонують в режимах реального часу.

## Побудова навчальної вибірки

Згідно з [53] та [72] процес навчання системи триває в середньому від 3 до 6 хвилин. Реалізований програмний додаток в середньому за 1 секунду гарантовано опрацьовує близько 18 фреймів. Таким чином, необхідна кількість фреймів для навчання бінарного класифікатору знаходиться в інтервалі від 3000 до 6000 фреймів.

Побудова навчальної вибірки для бінарного класифікатору, що розв'язує задачу визначення ФСО, виконується за допомогою алгоритмів пошуку аномалій з використанням унарного класифікатору. Емпіричним шляхом встановлено, що для функціонування унарного класифікатору достатньо 4000 фреймів, що відповідає 200 секундам спостереження.

Побудова навчальних вибірок починається разом з роботою програмного додатку. Так як, для навчання бінарного класифікатору необхідна достатня кількість представників кожного з класу, то під час побудови навчальної вибірки введемо поняття коефіцієнту швидкості навчання. Під даним коефіцієнтом розуміється відношення кількості фреймів, що описують сонливий стан до загальної їх кількості. Так як, стан оператора на початок роботи є завжди бадьорим, то збільшення даної величини свідчить про поступове засинання оператора. При досягненні необхідної кількості фреймів для навчання бінарного класифікатору, про що свідчить перевищення актуального значення коефіцієнту навчання над необхідним, визначення ФСО виконується за допомогою нього. Емпіричним шляхом встановлено, що оптимальне значення коефіцієнту швидкості навчання становить 0.85. Дане значення свідчить про те, що мінімум 85% фреймів сгенерованої вибірки описують сонливий стан. При такому значення генерація навчальної вибірки відбувається за мінімальний проміжок часу. Зміна значення даного коефіцієнту прямо пропорційна часу побудови навчальної вибірки. При цьому, в кожен момент часу аналізується інформації лише про останні 4000 фреймів, що становить 200 секунд спостереження.

## Алгоритми виявлення аномалій вхідних даних

Виходячи з опису системи, задача побудови навчальної вибірки для визначення функціонального стану оператора зводиться до задачі пошуку аномалій у вхідних даних. Аномалії можуть бути викликані: помилками вхідних даних, наявністю шумів або зміною функціонального стану. Під час побудови навчальної вибірки, найбільше значення мають аномалії, що викликані зміною стану оператора. Аномалії такого типу називаються новизною та використовуються при побудові бінарного класифікатора. Завдяки статичним умовам зйомки кількість аномалій, що викликана шумами буде незначною та не впливатиме на загальну достовірність класифікації.

Методи виявлення аномалій у вхідних даних поділяються на наступні категорії:

1. статистичні тести (методи, що базуються на міжквартильній відстані, критерії Граббса, критерії Пірса, критерії Діксона);
2. модельні тести;
3. ітераційні методи;
4. метричні методи (алгоритм локального рівня викиду);
5. методи підміни задачі;
6. методи машинного навчання.

Оптимальними методами для виявлення аномалій в реальному часі є представники статистичних тестів та метричних методів. Підходи, що належать до інших груп є ресурсоемкими та не можуть використовуватися в рамках розв'язання даної задачі. Головною вимогою до алгоритму виявлення аномалій є досягнення високої достовірності виявлення аномалій у вхідних даних при якомога меншому розмірі навчальної вибірки з мінімальною кількістю операцій. Завдяки цьому



досягається мінімальний час навчання системи та швидкий її перехід до режиму визначення ФСО.

Для оцінки достовірності класифікації алгоритму, використано вибірку, що складається з 35502 об'єктів, які рівномірно розподілені між класами. Для зменшення впливу шумів на роботу алгоритмів, з вхідних даних було виключено об'єкти, значення будь-якої ознаки  $X$  яких не потрапляє до діапазону  $[\bar{X} - 2 * \sigma; \bar{X} + 2 * \sigma]$ .

Для тестування вхідні дані розбиваються на 2 класи, що описують сонливий та бадьорий стан. Навчання класифікатора відбувається на даних, що описують бадьорий стан. Оцінка достовірності класифікації відбувається на вибірці, що складається з усіх об'єктів, що присутні у вхідних даних. Таким чином оцінюється достовірність виявлення аномалій та відсутність перенавчання класифікатора.

### Міжквартильна відстань

Найпростішим методом для пошуку викидів вибірки є алгоритм, що базується на міжквартильній відстані. Так, викидами класифікуються значення всіх величини, що не потрапляють в діапазон  $[q_{25} - 1.5 * (q_{75} - q_{25}); q_{75} + 1.5 * (q_{75} - q_{25})]$ , де  $q_{25}$  та  $q_{75}$  - 25-й та 75-й перцентилі, відповідно. Дані, до яких застосовується пошук аномалій з використанням міжквартильної відстані мають задовольняти нормальному розподілу, бути унімодальним та одномірними. При обробці багатомірних даних враховується кількість викидів по кожному з вимірів.

Результати тестування алгоритму виявлення аномалій, що базується на міжквартильній відстані для описаної вище навчальної вибірки, наведено в табл.11.

	Precision	Recall	f1-score
Сонливий стан	0.71	0.83	0.77
Бадьорий стан	0.78	0.65	0.71

Табл. 11: Результати класифікації для алгоритму, що базується на міжквартильній відстані

Для навчальної вибірки параметр  $recall$  цільового класу складає  $recall = 0.83$ , а для класу, на якому виконувалося навчання  $recall = 0.65$ , що свідчить про відсутність перенавчання у побудованого класифікатору. Значення параметру  $precision = 0.71$  для об'єктів, що описують сонливий стан свідчить про високу достовірність розпізнавання побудованого класифікатору, але недостатню для його використання в якості єдиного алгоритму виявлення аномалій.

### Система обмежень

При використанні класифікатору, що базується на системі обмежень необхідні мінімальні його модифікації для використання його в якості унарного класифікатору. Так, за замовчуванням для побудови класифікатору використовується лише ізольована інформація про клас, не враховуючи інформацію його сусідів. Об'єкт класифікується як викид, якщо сума відстаней для всіх ознак, до еталонного класу перевищує 0.2. Результати тестування алгоритму виявлення аномалій, що базується на системі обмежень наведено в табл.12.

	Precision	Recall	f1-score
Сонливий стан	1.00	0.61	0.76
Бадьорий стан	0.71	1.00	0.83

Табл. 12: Результати класифікації для класифікатору, що базується на системі обмежень

Згідно табл.12 класифікатор правильно визначив всі об'єкти навчальної вибірки для бадьорого стану. При цьому спостерігається перенавчання, про що свідчить значення метрики  $precision = 0.71$ . Тобто, кількість об'єктів, що класифіковані як такі, що описують бадьорий стан - завелика. В той же час, знайдено не всі об'єкти сонливого стану ( $recall = 0.61$ ), але всі об'єкти класифіковані як аномаліями виявлено правильно ( $precision = 1.00$ ). Таким чином, для даного класифікатору характерна наявність незначної помилки першого роду, вплив якої може бути

зменшеним при використанні його у складі ансамблю класифікаторів.

### Локальний рівень викиду

Локальний рівень викиду - алгоритм, що використовується для розпізнавання новизни та аномалій у вибірках. Підхід базується на локальній концентрації точок в околиці. Для оцінки концентрації точок використовується інформація про  $k$  найближчих сусідів. Стверджується, що аномалії мають менше значення локальної густини за аналогічний показник для об'єктів, що належать до вибірки  $D$ .

Для будь-якого позитивного  $k$  величина  $k$ -distance до об'єкту  $p$  обчислюється як відстань  $d(p, o)$  до всіх об'єктів  $o$ , що задовольняють наступним умовам:

- існує не менше  $k$  об'єктів  $u \in D$   $p : d(p, u) \leq d(p, o)$
- існує не більше ніж  $k - 1$  об'єктів  $u \in D$   $p : d(p, u) < d(p, o)$

Відстань досяжності  $reachDist_k(p, o)$  не є симетричною величиною та визначається як  $reachDist_k(p, o) = \max(d(p, o), k - distance(o))$ . Дана величина має єдине значення для всіх об'єктів, що належать до множини  $k$  найближчих сусідів ( $N_k$ ). Локальна густина досяжності характеризує середню відстань, на якій об'єкт  $p$  знаходиться відносно його сусідів обчислюється наступним чином:

$$lrd_k(p) = 1 / \left( \frac{\sum_{o \in N_k(p)} reachDist_k(p, o)}{|N_k(p)|} \right)$$

Значення локального рівня викиду є середньою локальною густиною досяжності об'єкту та обчислюється за наступною формулою:

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|}$$

Параметр  $LOF$  є величиною відносною і залежно від вибірки його значення може коливатися, тому для коректного пошуку аномалій необхідно використовувати відносні значення.

Для оцінки достовірності класифікації за допомогою алгоритму локального рівня викиду вихідна вибірка розбита на 2 частини: нормальний стан (timeline) та стан засинання(drowsiness). Так, всі реалізації, що належать до вибірки, що отримана під час засинання мають бути класифіковані як аномалії. Для цього вибірка timeline була обрана в якості навчальної, а drowsiness у якості екзаменаційної. Для тестування обрана реалізація алгоритму з кількістю сусідів  $k = 5$ , а для обчислення відстані використовується метрика Мінковського другого порядку. Розподіл значень  $LOF$  для об'єктів вибірки, залежно від їх належності до класу, що описує ФСО наведено на рис.21.

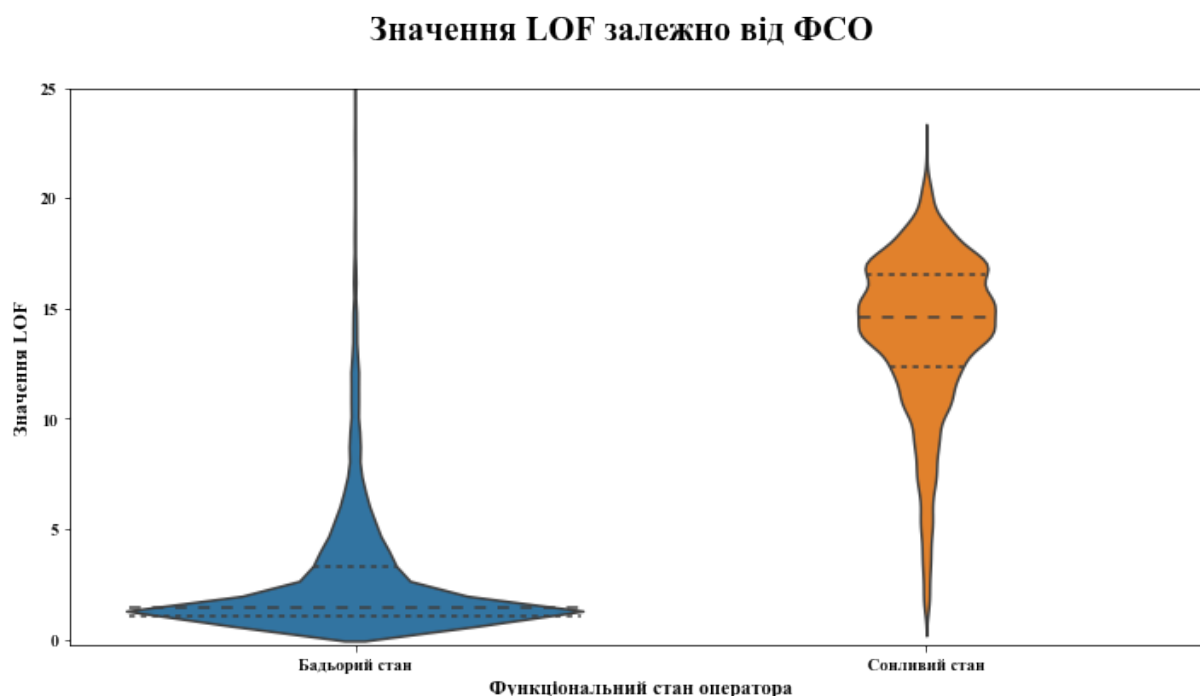


Рис. 21: Значення LOF залежно від класу об'єкту

Всі значення параметру  $LOF$  знаходяться в інтервалі  $[0; 25]$ . Для вибірки timeline характерне більше середнє значення параметру  $LOF$ , в той же час спостерігається незначний перетин частини значень з вибіркою drowsiness, але більшість значень  $LOF$  для вибірок не перетинаються за діапазонами. Так, для вибірки timeline основна частина значень знаходиться в інтервалі  $[0; 8]$ , а для вибірки drowsiness - в діапазоні  $[4; 30]$ , що свідчить про віддаленість об'єктів екзаменаційної вибірки від об'єктів навчальної вибірки.

Результати тестування алгоритму виявлення аномалій, що базується на локальному рівні викиду, наведено в табл.13.

	Precision	Recall	f1-score
Сонливий стан	0.93	0.74	0.83
Бадьорий стан	0.78	0.94	0.85

Табл. 13: Результати класифікації для алгоритму, що базується на локальному рівні викиду

Високі значення метрик *precision* та *recall* для обох класів свідчать про високу достовірність класифікації при відсутності перенавчання. Для сонливого стану спостерігається висока достовірність виявлення аномалій ( $precision = 0.93$ ,  $recall = 0.74$ ). Дані показники вищі, аніж для алгоритму, що базується на міжквартильній відстані, але недостатні для повноцінного використання даного алгоритму в якості єдиного класифікатора для виявлення аномалій.

### Використання інтелектуальної екстремальної інформаційної технології для пошуку аномалій у вхідних даних

Для класифікації аномалій у вхідних даних використовується модифікація інтелектуальної екстремальної інформаційної технології. Для коректного визначення статистичних характеристик, в якості базового класу обирається частина вибірки, що не містить викидів. В якості класу-сусіда не обирається жодного класу, а значення статистичних характеристик обираються рівними: помилка другого роду  $\beta^{(k)} = 0$ , друга достовірність  $D_2^{(k)} = 1$ . Дані значення описують взаємне розміщення контейнерів, при якому вхідні класи не перетинаються, та досягається максимальна міра різноманітності. Завдяки цьому, при обчисленні радіусу контейнера використовується лише інформація про поточний клас, без урахування його сусідів.

Емпіричним шляхом отримано оптимальні параметри функціонування алго-

ритму: базовий клас - 0 (що відповідає сонливому стану), рівень селекції  $\rho = 0.5$ , вектор значень ширини інтервалів допусків ознак приймає наступний вигляд  $\gamma = [0.25, 0.2, 250, 250, 250, 250, 0.01, 0.01, 0, 0, 50, 50, 0.3, 1.5, 1.55, 0.02]$ . В якості робочої формули використовується інформаційна міра Кульбака.

При використанні даного алгоритму для пошуку аномалій вхідних даних отримано високу достовірність розпізнавання, що наведено в табл.14.

	Precision	Recall	f1-score
Сонливий стан	0.64	0.77	0.70
Бадьорий стан	0.69	0.54	0.60

Табл. 14: Результати класифікації для інтелектуальної екстремальної інформаційної технології

Результати тестування інтелектуальної екстремальної інформаційної технології для виявлення аномалій близькі до результатів класифікатора, що базується на міжквартильній відстані. При цьому результати класифікації для бадьорого стану близькі до звичайного вгадування, а для сонливого стану свідчать про можливість використання даного класифікатора у складі ансамблю для виявлення аномалій.

Незважаючи на високі результати достовірності класифікації, даному алгоритму притаманні певні недоліки. Головним недоліком даного класифікатора є необхідність адаптувати параметри його функціонування до навчальної вибірки. Так, в рамках задачі, при значній зміні умов спостереження або оператору необхідно обчислити нові значення параметрів. Через велику розмірність вхідних даних та відсутність розміченої вибірки дана задача не може бути вирішена в реальному часі.

## Ансамбль класифікаторів

Для підвищення достовірності визначення переходу оператору до стану дрімоти, використовується ансамбль класифікаторів, що включає в себе класифікатори, що базуються на міжквартильній відстані, локальному рівні викиду, системі обмежень та інтелектуальній екстремальній інформаційній технології. Головною метою об'єднання класифікаторів в ансамбль є покращення властивостей, що збільшують достовірність класифікації для кожного з класифікаторів та зменшення загальної помилки першого роду для цільового класу.

Під час побудови ансамблю параметри функціонування класифікаторів ідентичні до наведених вище, при їх використанні в якості єдиного класифікатора. В якості вирішального правила обрано незважене голосування - не надається перевага результату будь-якого з класифікаторів. Результати тестування ансамблю класифікаторів наведені в табл.15.

	Precision	Recall	f1-score
Сонливий стан	0.77	0.82	0.80
Бадьорий стан	0.80	0.75	0.77

Табл. 15: Результати класифікації для ансамблю класифікаторів

Незважаючи на те, що значення метрики  $f1 - score$  ансамблю класифікаторів менше за аналогічний показник для окремих алгоритмів (алгоритм локального рівня викиду), значення метрик  $precision$  та  $recall$  для цільового класу вищі. Значення аналогічних метрик для об'єктів, що описують бадьорий стан відрізняються незначним чином, що свідчить про відсутність перенавчання отриманого ансамблю. Дане твердження підтвердилося під час крос-валідації. Таким чином, завдяки об'єднанню класифікаторів в ансамбль вдалося підвищити достовірність виявлення аномалій без значного зменшення швидкодії алгоритму.

## Визначення функціонального стану оператора

### Інтелектуальна екстремальна інформаційна технологія

Інтелектуальна екстремальна інформаційна технологія - алгоритм бінарної класифікації, що має найбільшу кількість гіперпараметрів, відносно аналогічних алгоритмів. Максимальна достовірність класифікації досягається при використанні класу, що описує бадьорий стан в якості базового, а значення рівень селекції  $\rho = 0.6$ . Через значну дисперсію значень ознак доцільно використовувати вектор  $\gamma$  значень ширини інтервалів допуску для ознак. Емпіричним шляхом встановлено наступні значення елементів вектору  $\gamma$ :

$$\gamma = [0.25, 0.2, 250, 250, 250, 250, 0.01, 0.01, 50, 50, 50, 50, 0.3, 0.5, 0.55, 0.02]$$

Результати тестування побудованого класифікатора з використанням інтелектуальної екстремальної інформаційної технології наведено в табл.16.

	Precision	Recall	f1-score
Сонливий стан	0.80	0.72	0.76
Бадьорий стан	0.73	0.81	0.77

Табл. 16: Результати класифікації інтелектуальної екстремальної інформаційної технології

Загальні результати класифікації для об'єктів сонливого стану недостатні для його використання в якості єдиного класифікатора. Так, значення метрики *precision* = 0.8 свідчить про те, що загалом 80% об'єктів віднесено до сонливого стану правильно, але з загальної вибірки знайдено лише 72% об'єктів (*recall* = 0.72). Для класу, що описує бадьорий стан результати незначним чином кращі: загалом знайдено 81% об'єктів, але значення метрики *precision* = 0.73 свідчить про наявність помилок першого роду. Таким чином, достовірність класифікації вища за тривіальне вгадування, та недостатня для використання в якості єдиного класифікатора.



Шляхом додавання даного класифікатору до ансамблю, можливо зменшити кількість помилок першого роду при класифікації об'єктів сонливого стану. За рахунок інших класифікаторів, що входять до ансамблю, можливо збільшити загальну кількість об'єктів, що ймовірно відносяться до цільового класу. Таким чином, теоретично, повинна збільшитися достовірність класифікації реалізацій, що описують сонливий стан. Значний вплив на метрики, що описують результати класифікації об'єктів бадьорого стану не має сенсу, адже дана інформація використовується лише на етапі навчання системи.

### Наївний баєсів класифікатор

Наївний баєсів класифікатор - алгоритм класифікації, що базується на теоремі Байеса[126] для обчислення належності об'єкту  $x_i$  до певного класу  $y$ . Оцінка параметрів побудованого класифікатору виконується за допомогою методу максимальної правдоподібності[127], що базується на максимізації значення функції правдоподібності. Основною формулою, що реалізує наївний баєсів класифікатор є:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right),$$

де  $P(x_i|y)$  - ймовірність належності реалізації  $x_i$  до класу  $y$ . Для обчислення значення параметрів математичного очікування  $\mu_y$  та середньоквадратичного відхилення  $\sigma_y$  використовується метод максимальної правдоподібності.

Результати тестування наївного баєсового класифікатора на розміченій вибірці, що описує ФСО наведено в табл.17.

	Precision	Recall	f1-score
Сонливий стан	0.86	0.65	0.74
Бадьорий стан	0.71	0.89	0.79

Табл. 17: Результати класифікації наївного баєсового класифікатору

При використанні наївного баєсового класифікатора для визначення ФСО отримано достовірність класифікації вищу за середню. Для цільового класу отримано значення метрики  $f1 - score = 0.74$ . Так як, для даного алгоритму відсутні будь-які параметри, зміною яких можливо вплинути на достовірність класифікації, то єдиним шляхом для збільшення достовірності класифікації є трансформація вхідної вибірки. При цьому, відсутність гіперпараметрів алгоритму дозволяє спростити процес навчання класифікатора за рахунок відсутності етапу оптимізації гіперпараметрів. Таким чином, враховуючи переваги побудованого класифікатора, його доцільно використовувати для визначення ФСО, але лише у складі ансамблю з метою збільшення достовірності класифікації.

## Логістична регресія

Логістична регресія - бінарний класифікатор, котрий базується на ідеї лінійного розділення реалізацій на класи. Для задачі класифікації багатомірних даних, класи розділяються між собою за допомогою площини, що називається лінійним дискримінантом. Найвища достовірність класифікації для логістичної регресії досягається для вхідних даних, кожна ознака яких лінійно розділяється на два класи, що не перетинаються. В рамках даної задачі використовується класичний бінарний класифікатор, що не потребує значних модифікацій логіки функціонування, а побудовані правила класифікації є лінійними класифікаторами.

Результати тестування логістичної регресії на навчальній вибірці наведено в табл.18.

	Precision	Recall	f1-score
Сонливий стан	0.96	0.67	0.79
Бадьорий стан	0.73	0.97	0.83

Табл. 18: Результати класифікації логістичної регресії

Незважаючи на простоту реалізації та високу швидкість навчання, даний кла-

сифікатор має високий показник метрики  $precision = 0.96$  для цільового класу. Це свідчить про низьку кількість хибних спрацювань. В той же час, метрика  $recall = 0.67$  свідчить про наявність значної кількості помилок першого роду. Для класу, що описує бадьорий стан значення метрик  $recall = 0.97$  та  $precision = 0.73$  свідчить про той факт, що більшість реалізацій навчальної вибірки віднесено саме до цього класу. Так як, в навчальній вибірці більша частина реалізацій належить до бадьорого стану, то для даного класифікатору спостерігається факт незначного перенавчання, що загалом незначним чином зменшує достовірність класифікації для цільового класу.

### **Класифікатор, що базується на системі обмежень**

Побудова класифікатору, що базується на системі обмежень полягає в обчисленні мінімального та максимального значень кожної з ознак. Окрім того, для зменшення впливу викидів на достовірність класифікації додатково враховується середнє значення ознаки. Таким чином, для даного класифікатору відсутні будь-які гіперпараметри, що потребують підбору значень залежно від вхідних даних. Завдяки цьому спрощується процес побудови класифікатора.

Для кожної ознаки вхідної вибірки були обчислені описані вище характеристики та побудовані довірчі інтервали, що будуть використовуватися при класифікації. Сума відстаней між інтервалами, що описують відповідні класи, для кожної з ознак прямо пропорційна достовірності класифікації. Таким чином - для даного класифікатора в якості попередньої підготовки даних рекомендується відкинути всі викиди для кожного з класів. Завдяки цьому зменшиться площа перетину відповідних довірчих інтервалів, що позитивно вплине на достовірність класифікації. Результати тестування класифікатору, що базується на системі обмежень наведено в табл.19.

	Precision	Recall	f1-score
Сонливий стан	0.73	0.90	0.81
Бадьорий стан	0.87	0.65	0.74

Табл. 19: Результати класифікації класифікатору, що базується на системі обмежень

В результаті тестування виявлено, що побудований класифікатор має приблизно однакові значення метрик  $f1 - score$  для обох класів. Окрім того, значення метрики  $precision$  значно більше за 0.5, що свідчить про те, що отриманий класифікатор функціонує значно краще, аніж тривіальне вгадування. Високе значення метрики  $precision = 0.87$  та  $precision = 0.73$  для бадьорого та сонливого станів, відповідно, свідчить про незначну площу перетину довірчих інтервалів. Значення метрики  $recall = 0.9$  для сонливого стану свідчить про доцільність використання даного класифікатору для визначення ФСО, адже для даної вибірки знайдено 90% реалізацій, що описують сонливий стан. Кількість хибних спрацювань відображена метрикою  $precision = 0.73$ , яка свідчить про те, що 73% об'єктів класифіковано правильно, а 27% реалізацій бадьорого стану класифіковано як сонливий стан. Отже, для даного класифікатору характерна незначна кількість хибних спрацювань, що не дозволяє використовувати його як єдиний класифікатор. Даний недолік може бути скомпенсований при функціонуванні даного класифікатору у складі ансамблю.

## Нейронна мережа

Нейронна мережа представляє собою набір шарів, перехід між якими виконується за допомогою функції активації. В якості функції активації для задачі визначення ФСО обрано наступні функції: випрямляч (ReLU)[128], його гладке наближення (softplus) та сигмоїда (sigmoid)[129]. Робочі формули для кожної з

функцій наведено нижче:

$$ReLU : f(x) = \max(x, 0)$$

$$softplus : f(x) = \log(1 + \exp(x))$$

$$sigmoid : f(x) = \frac{1}{1 + e^{-x}}$$

В якості нейронної мережі використовується багатошаровий перцептрон, що складається з семи шарів:

1. Повнозв'язний шар, що складається з 30 нейронів, а в якості функції активації - softplus;
2. Повнозв'язний шар, що складається з 60 нейронів, а в якості функції активації - ReLU;
3. шар, що зі ймовірністю 0.3 відкидає значення, отримані з попереднього шару;
4. Повнозв'язний шар, що складається з 120 нейронів, а в якості функції активації - ReLU;
5. шар, що зі ймовірністю 0.5 відкидає значення, отримані з попереднього шару;
6. Повнозв'язний шар, що складається з 50 нейронів, а в якості функції активації - ReLU;
7. Повнозв'язний шар, що складається з одного нейрону, а в якості функції активації виступає сигмоїд. Даний шар є вихідним шаром нейронної мережі.

Емпіричним шляхом отримано оптимальну конфігурацію, що дозволяє отримати найвищу достовірність класифікації при найменшій алгоритмічній складності.

При цьому, доповнення нейронної мережі додатковими шарами або зміна функцій активації не дозволяє збільшити достовірність розпізнавання. Результати тестування нейронної мережі на вхідних даних наведено в табл.20.

	Precision	Recall	f1-score
Сонливий стан	0.97	0.70	0.81
Бадьорий стан	0.75	0.98	0.85

Табл. 20: Результати класифікації нейронної мережі

При класифікації об'єктів сонливого класу отримано низький показник метрики  $recall = 0.7$ , але в той же час, достатньо високий показник метрики  $precision = 0.97$  свідчить про низьку кількість хибних спрацювань. Даний фактор є негативним при використанні нейронної мережі як незалежного класифікатора, але може значним чином підвищити достовірність класифікації при введенні даного класифікатора в ансамбль.

Високе значення метрики  $recall = 0.98$  та відносно низьке значення метрики  $precision = 0.75$  для реалізацій, що описують бадьорий стан, може свідчити про незначне перенавчання системи внаслідок нерівномірного розподілу об'єктів між класами.

Таким чином, якщо реалізація описує об'єкт сонливого стану, то с високою достовірністю його буде класифіковано правильно. Дану властивість доцільно використовувати при побудові ансамблю класифікаторів.

## Дерево прийняття рішень

Так як, вхідні дані містять невелику кількість ознак, то доцільно перевірити можливість використання дерева прийняття рішень в якості алгоритму класифікації. Максимальна глибина дерева обмежена 15 рівнями, а мінімальна кількість об'єктів в одній вершині - не менше 5. Дерево прийняття рішень побудовано за допомогою алгоритму CART[130]. Вплив кожної ознаки при класифікації об'єкту

наведено в табл.21.

Назва ознаки	Вага
eye_square_running_mean_1000	0.4659
mouth_square_running_mean_1000	0.3094
head_incline	0.0759
head_vertical_incline	0.0414
head_movement_running_mean_1000	0.0413
mouth_square	0.0219
blinks_frequency	0.0218
head_vertical_incline_running_mean_1000	0.0141

Табл. 21: Вплив ознак на достовірність класифікації

Для всіх інших ознак, вага становить менше за 0.01, що свідчить про їх малу значимість при класифікація. Дана інформація може бути використана при зменшенні об'єму навчальної вибірки за рахунок вибору ознак класифікації. Зміна конфігурації дерева прийняття рішень не призводить до значної зміни розподілу значимості ознак.

В результаті застосування побудованого дерева прийняття рішень, для наведеної вище навчальної вибірки отримано значення метрик *precision*, *recall*, *f1 – score*, що наведено в табл.22.

	Precision	Recall	f1-score
Сонливий стан	1.00	0.75	0.85
Бадьорий стан	0.79	1.00	0.88

Табл. 22: Результати класифікації дерева прийняття рішень

З результатів, описаних в табл.22 приходимо до висновку, що для дерева прийняття рішень, на даній навчальній вибірці без її модифікації спостерігається перена-

вчання. Даного факту неможливо уникнути шляхом зміни гіперпараметрів класифікатору. Про факт перенавчання свідчить високе значення метрики  $recall = 1$  при класифікації об'єктів, що описують бадьорий стан, та значно нижче значення аналогічної метрики для об'єктів, що описують сонливий стан ( $recall = 0.75$ ). В той же час, значення метрики  $precision = 1$  для об'єктів сонливого стану свідчить про значні перспективи використання даного класифікатору у складі ансамблю.

## Ансамбль класифікаторів

Кожен з описаних вище класифікаторів не досягає достовірності класифікації, необхідної для його використання в якості єдиного класифікатору для визначення ФСО. Саме тому доцільно об'єднати їх в ансамбль для взаємної компенсації недоліків кожного з алгоритмів.

Так, для інтелектуальної екстремальної інформаційної технології характерне високе значення метрики  $precision = 0.80$  при класифікації реалізацій, що належать до сонливого стану. При введенні даного класифікатору в ансамбль очікується збільшення значення метрики  $recall = 0.72$  для сонливого стану без зменшення достовірності класифікації в цілому.

Через алгоритмічну схожість алгоритму ІЕІТ та наївного баєсового класифікатору - результати їх тестування доволі близькі. Так, для цільового класу значення метрики  $recall$  для наївного баєсового класифікатору складає  $recall = 0.65$ , а метрика  $precision$  для цього ж класу  $precision = 0.86$ . Тобто, порівняно з алгоритмом ІЕІТ, наївний баєсів класифікатор правильно позначив меншу частину реалізацій сонливого стану, але при цьому має меншу частоту виникнення помилок другого роду.

Для класифікатору, що базується на системі обмежень притаманне високе значення метрики  $recall = 0.9$  для цільового класу, але значення метрики  $precision = 0.73$  значно менше за аналогічний показник у його конкурентів. Саме тому, при включенні даного алгоритму до ансамблю очікується загальне підвищення значення метрики  $recall$  всього ансамблю за рахунок класифікатору, що базується на



системі обмежень, а підвищення метрики *precision* має бути виконано за рахунок інших представників ансамблю.

Для логістичної регресії характерне низьке значення метрики  $recall = 0.67$  для цільового класу, але значення метрики  $precision = 0.96$  свідчить про те, що 96% реалізацій, що визначені класифікатором як сонливий стан, класифіковані правильно. При додаванні логістичної регресії до ансамблю класифікаторів дана властивість може значним чином підвищити достовірність класифікації. Так, наприклад, значна частина реалізацій сонливого стану може бути знайдена за допомогою класифікатора, що базується на системі обмежень, а за допомогою логістичної регресії серед них будуть відкинуті реалізації, що відносяться до бадьорого стану.

Результати тестування нейронної мережі на навчальній вибірці незначним чином кращі за аналогічні результати для логістичної регресії. Так, для нейронної мережі метрики, що описують достовірність класифікації об'єктів сонливого стану становлять  $recall = 0.7$  та  $precision = 0.97$ . Тобто, повністю перекривають результати логістичної регресії. Під час побудови ансамблю класифікаторів доцільно перевірити чи перетинаються результати класифікації відповідних об'єктів. Можлива ситуація, коли значення метрики *recall* для логістичної регресії та нейронної мережі досягнуто за рахунок класифікації різних об'єктів. У такому випадку, дані класифікатори доповнюють один одного.

При класифікації об'єктів навчальної вибірки з використанням дерева прийняття рішень, отримано найвищий показник метрики  $precision = 1$  для реалізацій, що описують сонливий стан. Даний результат незначним чином вищий за аналогічний показник для нейронної мережі. В той же час, значення метрики  $recall = 0.75$  вище за значення цієї ж метрики для нейронної мережі, що, скоріше за все, свідчить про те, що доцільно в склад ансамблю внести дерево прийняття рішень замість нейронної мережі, за умови, що дані значення метрик досягнуті при правильній класифікації множин об'єктів, що не перетинаються. Беручи до уваги наведені вище факти, приходимо до висновку, що сильною стороною

дерева прийняття рішень є висока достовірність класифікації об'єктів сонливого стану ( $precision = 1$ ). Попередній розподіл об'єктів між класами може бути виконано, наприклад, за допомогою класифікатору, що базується на системі обмежень ( $recall = 0.9$ ) або сукупності інших класифікаторів, що доповнюють один одного.

Головною задачею під час побудови ансамблю класифікаторів є досягнення максимальної достовірності класифікації об'єктів навчальної вибірки при мінімальній асимптотичній складності. Оптимальне рішення даної задачі полягає у мінімізації загальної кількості класифікаторів, з яких складається ансамбль. Окрім того, однією з ключових задач є вибір стратегії формування вердикту ансамблю класифікаторів на основі результатів класифікації окремих класифікаторів.

Загальна схема функціонування ансамблю класифікаторів, що складається з класифікаторів, що наведені вище, зображена на рис.22.

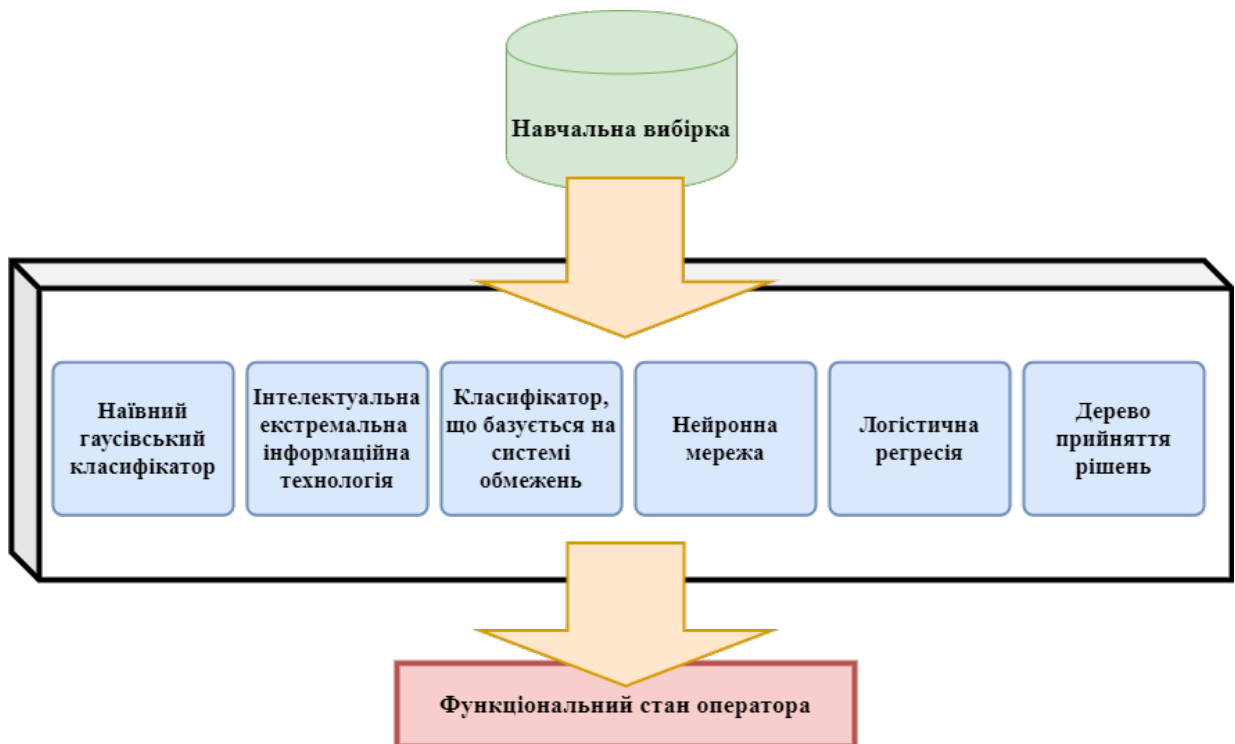


Рис. 22: Структура ансамблю класифікаторів для визначення ФСО

Життєвий цикл ансамблю класифікаторів, як і життєвий цикл одного класифікатору, представлений двома етапами: навчання та екзамен. На етапі навчання

кожен класифікатор навчається на розмічених даних навчальної вибірки. Під час екзамену, кожним класифікатором, що входить до складу ансамблю, визначається клас даної реалізації. Використовуючи вихідні дані кожного класифікатора, за допомогою зваженого або незваженого голосування, формується вердикт ансамблю класифікаторів в цілому.

Найбільш примітивний ансамбль класифікаторів будується шляхом навчання кожного з базових класифікаторів на навчальній вибірці та об'єднання результатів їх класифікації шляхом зваженого голосування. Результати роботи класифікаторів, що побудований за алгоритмом, описаним вище, у наведено в табл.23.

	Precision	Recall	f1-score
Сонливий стан	0.96	0.81	0.88
Бадьорий стан	0.82	0.97	0.89

Табл. 23: Результати класифікації з використанням ансамблю класифікаторів

Найвища достовірність класифікації досягається для ансамблю, що складається з наступних класифікаторів: класифікатор, що базується на системі обмежень, інтелектуальна екстремальна інформаційна технологія та дерево прийняття рішень. Подальше збільшення кількості класифікаторів не призводить до збільшення достовірності класифікації, а в деяких конфігураціях і навпаки - значним чином її знижує. Таким чином, композиція класифікаторів в примітивний ансамбль не мала очікуваного збільшення достовірності розпізнавання. Незначне збільшення достовірності класифікації компенсується збільшенням алгоритмічної складності отриманої системи.

Для збільшення достовірності класифікації ансамблю класифікаторів використовуємо бустинг. Даний підхід базується на послідовному навчанні класифікаторів на частинах вибірки, де попереднім класифікатором отримано неправильний результат. В якості ансамблю класифікаторів використовуємо ансамбль, отриманий на попередньому етапі. Спочатку класифікація виконується за допомогою класи-

фікатору, що базується на системі обмежень. Після цього, об'єкти, що були класифіковані неправильно використовуються в якості навчальної та тестової вибірки для інтелектуальної екстремальної інформаційної технології. Аналогічні операції виконуються для дерева прийняття рішень.

Достовірність класифікації ансамблю класифікаторів, який побудовано з використанням бустингу наведено в табл.24.

	Precision	Recall	f1-score
Сонливий стан	1.00	0.90	0.95
Бадьорий стан	0.91	1.00	0.95

Табл. 24: Результати класифікації з використанням ансамблю класифікаторів та бустингу

Побудований таким чином класифікатор має вищу достовірність класифікації за звичайний ансамбль без зменшення швидкодії алгоритму класифікації.

### 3.4 Формування набору доречних ознак

Головною вимогою до побудованого ансамблю класифікаторів, окрім високої достовірності класифікації, є висока швидкість класифікації. Так як процес навчання класифікатору виконується під час функціонування системи, то необхідно мінімізувати час навчання системи без зниження достовірності класифікації. Одним з варіантів вирішення даної задачі є зменшення розмірності вхідних даних за рахунок виключення з навчальної вибірки ознак, що корелюють між собою або не впливають на результат класифікації.

Для зниження розмірності вхідних даних та відкидання ознак, що є надлишковими та недоречними використовується відбір ознак. Для відбору ознак використовується група обгорткових методів. При даному підході використовується тренування класифікатору на всіх можливих наборах ознак та обирається такий набір, при якому достовірність класифікації максимальна.

При використанні реалізації алгоритму ІЕІТ, що базується на векторі значень  $\beta$  можливо оцінити вплив кожної ознаки на достовірність класифікації шляхом коригування відповідних компонентів вектору  $\beta$  та впливу на достовірність класифікації. В результаті тестування алгоритму ІЕІТ на різноманітних наборах вхідних ознак, отримано, що на достовірність класифікації впливають наступні ознаки: `mouth_square_running_mean_1000`, `left_eye_upper`, `right_eye_upper`, `head_incline`, `head_incline_running_mean_1000`, `head_vertical_incline`, `head_vertical_incline_running_mean_1000`, `pupil_movement`, `head_movement_running_mean_1000`, `blinks_frequency`.

Для вибору множини незалежних величин серед ознак, що описують функціональний стан ока використовується коефіцієнт кореляції Пірсона. Взаємну кореляцію даних величин, обчислену за допомогою коефіцієнту кореляції Пірсона, зображено на рис.23.

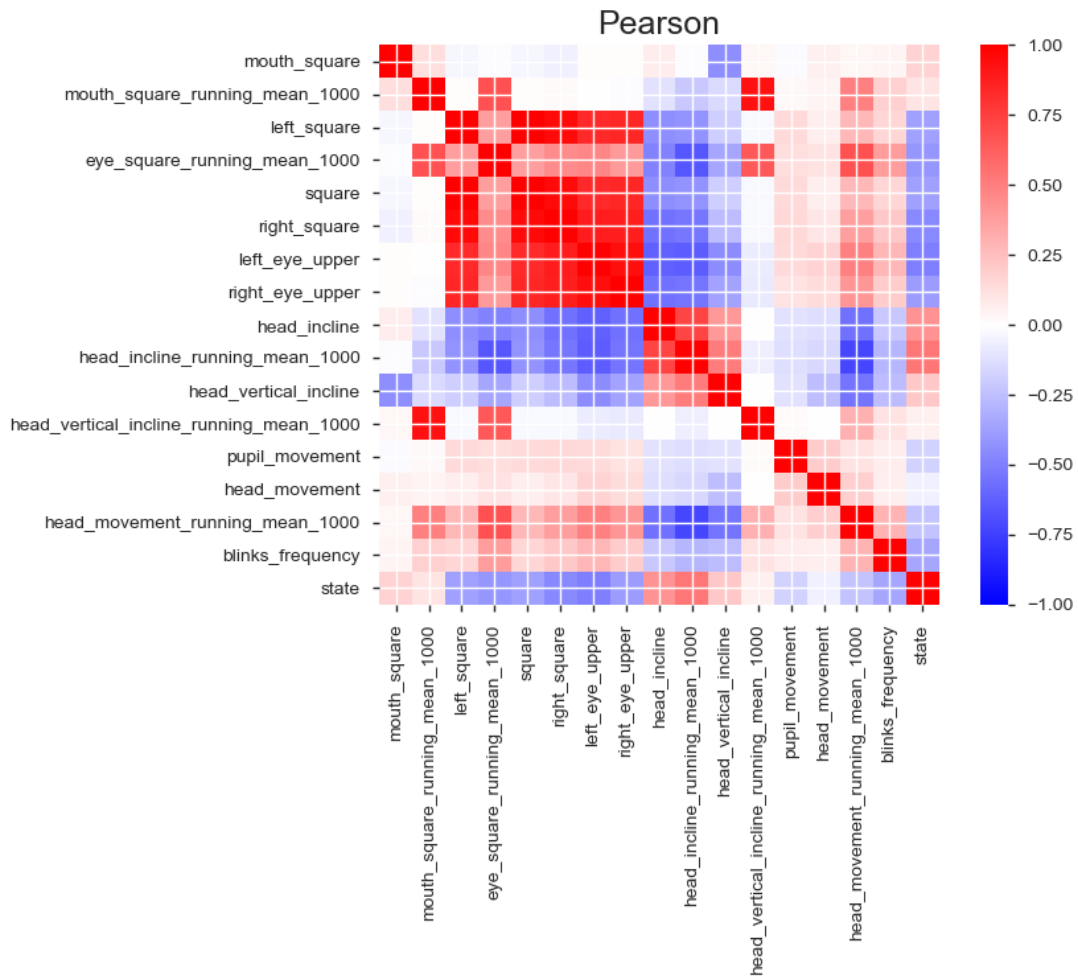


Рис. 23: Коефіцієнт кореляції Пірсона для ознак, що описують функціональний стан оператора

На рис.23 спостерігається висока кореляція між ознаками `head_vertical_incline_run` та `mouth_square_running_mean_1000` ( $\rho = 0.93$ ). Тобто, при виключенні будь-якої з наведених ознак із навчальної вибірки можливо досягнути підвищення швидкості роботи алгоритму без значного зменшення достовірності класифікації. Окрім того, спостерігається висока кореляція між відповідними ознаками для лівого та правого очей. Так, недоцільно використовувати два окремих значення ККП верхньої повіки для лівого та правого очей, так як вони корелюють між собою зі значенням  $\rho = 0.94$ . Аналогічно, розглянемо видиму площу лівого та правого очей. Значення ознака `square` повністю дублюють значення ознаки `left_square`, а тому може бути відкинута. Також, існує висока кореляція між видимою площею правого (`right_square`) та лівого (`left_square`) очей ( $\rho = 0.96$ ).

Таким чином, з навчальної вибірки можуть бути виключені наступні ознаки: left\_square, right\_square та mouth\_square\_running\_mean\_1000 без значного зниження достовірності класифікації.

З використанням дерев прийняття рішень можливо оцінити вплив кожної ознаки на загальну достовірність розпізнавання. Важливість ознаки прямо пропорційно залежить від відношення кількості об'єктів вибірки, що потрапили до вершини відносно загального обсягу вибірки. Так як головною метою при побудові дерева прийняття рішень є мінімізація його висоти, то таке розбиття є оптимальним. Детальну інформація про використання кожної ознаки в процентному співвідношенні та її вплив на достовірність класифікації наведено в табл.25.

Назва ознаки	Вага
eye_square_running_mean_1000	0.4659
mouth_square_running_mean_1000	0.3094
head_incline	0.0759
head_vertical_incline	0.0414
head_movement_running_mean_1000	0.0413
mouth_square	0.0219
blinks_frequency	0.0218
head_vertical_incline_running_mean_1000	0.0141
head_movement	0.002631
right_eye_upper	0.001991
square	0.001316

Табл. 25: Вплив ознак навчальної вибірки на достовірність класифікації

В табл.25 наведено лише ті ознаки, які хоча б один раз використовувалися під час розбиття вершини дерева прийняття рішень. Проаналізувавши табл.25 приходимо до висновку, що найбільшу вагу під час класифікації мають наступні ознаки: eye\_square\_running\_mean\_1000, mouth\_square\_running\_mean\_1000, що не-

значним чином суперечить результатам, що отримані з використанням критерію Пірсона. Порядок ознак, наведений в табл.25 є відсортованим за їх впливом на достовірність класифікації. Використовуючи отримані дані можливо отримати оптимальний набір ознак, що забезпечуватиме максимальну достовірність класифікації при мінімально можливій алгоритмічній складності. Для побудови результуючого набору ознак розпізнавання необхідно поступово виключати з навчальної вибірки ознаки у порядку зростання їх впливу на достовірність класифікації.

Проаналізувавши дані про важливість ознак, отримані з використанням різноманітних підходів, отримуємо, що оптимальний набір ознак можливо побудувати шляхом виключення з навчальної вибірки ознак, отриманих за допомогою дерева прийняття рішень. Даний підхід не суперечить результатам, що отримані за допомогою критерію Пірсона та алгоритму ІЕІТ. При цьому, дерево прийняття рішень дозволяє отримати порядок виключення ознак, що неможливо отримати за допомогою алгоритму ІЕІТ.

Приймаючи до уваги інформацію про вплив кожної ознаки на достовірність класифікації побудовано навчальну вибірку, з якої виключено ознаки, вплив яких мінімальний: `left_square`, `square`, `right_square`, `left_eye_upper`, `right_eye_upper`, `head_vertical_incline_running_mean_1000`. Результати класифікації побудованого раніше ансамблю класифікаторів на отриманій вибірці наведено в табл.26.

	Precision	Recall	f1-score
Сонливий стан	1.00	0.94	0.97
Бадьорий стан	0.94	1.00	0.97

Табл. 26: Результати класифікації ансамблю класифікаторів з використанням навчальної вибірки після відбору ознак

Таким чином, завдяки процедурі відбору ознак вдалося незначною мірою підвищити достовірність класифікації побудованого ансамблю. Окрім того, зменшення кількості ознак навчальної вибірки значним чином підвищує швидкодію си-



стеми за рахунок зменшення кількості обчислень необхідних для побудови математичного опису цих ознак із вхідного зображення та зменшення об'єму вхідних даних для класифікаторів.

Отже, в результаті об'єднання класифікаторів в ансамбль та зменшення об'єму вхідних даних за допомогою відбору ознак, отримано систему з високою швидкістю та високою достовірністю класифікації. Отримана система може використовуватися в режимі реального часу для визначення функціонального стану оператора.

## 4 Апаратне та програмне забезпечення адаптивної системи визначення функціонального стану оператора

### 4.1 Апаратна частина

В якості вхідних даних для системи визначення функціонального стану оператора виступає відеопотік. Джерелом даного потоку може бути будь-яка вебкамера або відеофайл. Для функціонування даної системи в реальних умовах, необхідно досягти максимальної її швидкодії при мінімальних розмірах додаткового обладнання. При цьому, якщо дана система використовується в якості інформаційної системи контролю пильності водія транспортного засобу, то необхідно досягти мінімальних розмірів вебкамери. В той же час, для оператора ПК дана умова не є необхідною.

В загальному випадку система розпізнавання функціонального стану оператора складається з вебкамери та комп'ютеру, за допомогою якого оброблюються вхідні дані. Низька асимптотична складність застосованих алгоритмів дозволяє використовувати в якості платформи для запуску програмного коду одноплатний комп'ютер на базі Raspberry Pi або Orange Pi. В якості веб-камери використовується офісна веб-камера Gemix T21 з роздільною якістю не менше ніж 640x480. Збільшення роздільної якості значним чином не впливає на достовірність класифікації, але значно знижує швидкість системи. Окрім того можливо використання штатної веб-камери Raspberry Pi.

### 4.2 Структура програмного додатку

Для визначення функціонального стану оператора розроблено програмний додаток мовою Python 3.7. Формування навчальної вибірки виконується за допомогою сторонніх бібліотек: для отримання вхідного відеопотоку та формування по-

слідовності фреймів з нього використовується бібліотека *OpenCV*, а для пошуку обличчя та ключових точок використовується бібліотека *dlib*.

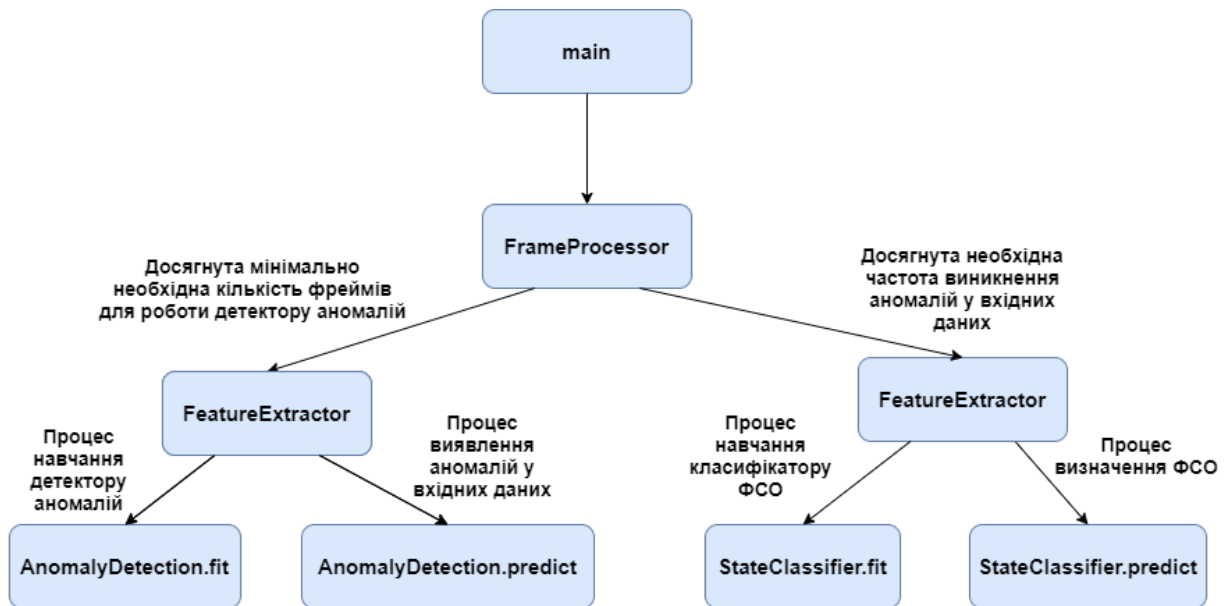


Рис. 24: Процес обробки вхідного фрейму

На вхід програмному додатку передається відеопотік, який в файлі *main.py* розбивається та аналізується пофреймово. Для кожного фрейму з використанням класу *FrameProcessor* будується об'єкт *FrameDescription*, що є числовим описом вхідного фрейму. Незалежно від подальших дій, для кожного фрейму відбувається побудова математичного опису за допомогою класу *FeatureExtractor*. Результатом роботи даного класу є набір ознак, що містять інформацію про зміну числових характеристик видимих частин обличчя. На наступному етапі, залежно від поточного стану системи, виконується або побудова навчальної вибірки або визначення функціонального стану оператора.

Для побудови навчальної вибірки, що містить представників обох класів використовується детектор аномалій. Одразу після початку роботи системи будується навчальна вибірка, що складається з 4000 фреймів та призначена для навчання детектору аномалій. Після того, як досягнуто необхідний мінімальний об'єм навчальної вибірки виконується навчання детектору аномалій. В подальшому даний класифікатор переходить у режим екзамену. Завдяки цьому будується навчальна

вибірка, що буде використана при навчанні ансамблю класифікаторів для визначення ФСО. Для кожного фрейму, що отримано з вхідного потоку виконується визначення його належності до викидів. У випадку, якщо даний об'єкт є аномалією, то у навчальній вибірці він класифікується як об'єкт, що описує сонливий стан оператора. Всі інші об'єкти описують бадьорий стан оператора. Так як, до аномалій відносяться об'єкти, що описують блимання, то для зменшення їх кількості у побудованій вибірці, ключовим параметром, що свідчить про завершення процесу побудови вибірки є частота виникнення аномалій за останні 4000 фреймів. Емпіричним шляхом встановлено, що оптимальним значенням відношення кількості аномалій до загальної кількості фреймів на інтервалі довжиною в 4000 фреймів є число 0.85.

Наступним етапом виконується навчання ансамблю класифікаторів, з використанням навчальної вибірки, що отримана на попередньому кроці. Після навчання ансамблю система переходить у режим екзамену та виконує визначення функціонального стану оператора для кожного з фреймів, що отримані з вхідного потоку.

Схематично робота програмного додатку зображена на рис.25.

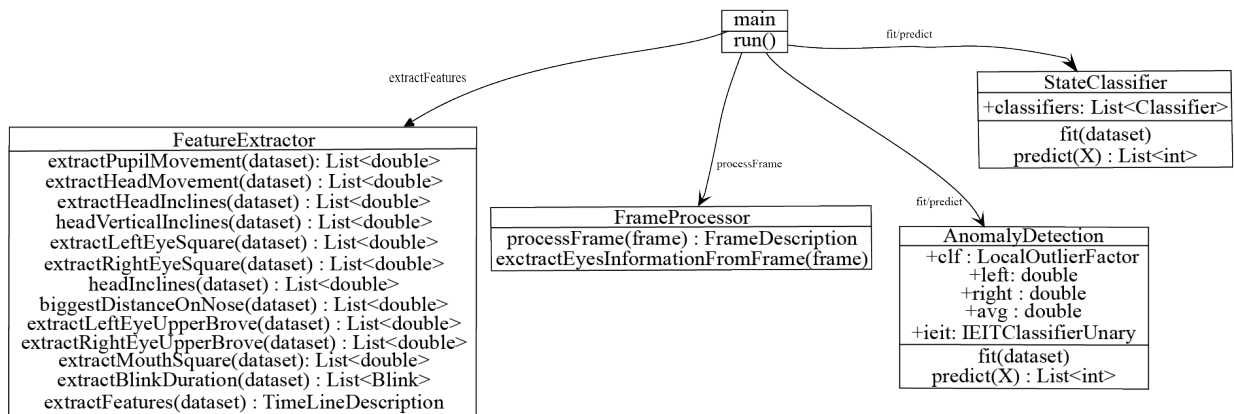


Рис. 25: Структура програми

### 4.3 Побудова навчальної вибірки

Першим етапом роботи програми є побудова навчальної вибірки шляхом обробки вхідного фрейму та побудови математичної моделі елементів обличчя. Для отримання

мання фрейму з відеокамери використовується бібліотека OpenCV. Для локалізації ключових точок обличчя використовується функція `get_frontal_face_detector` з бібліотеки `dlib`. Після отримання ключових точок виконується побудова математичної моделі обличчя оператора.

## Побудова математичної моделі обличчя

Обробка кожного фрейму виконується за допомогою класу `FrameProcessor`, структура якого зображена на рис.26.

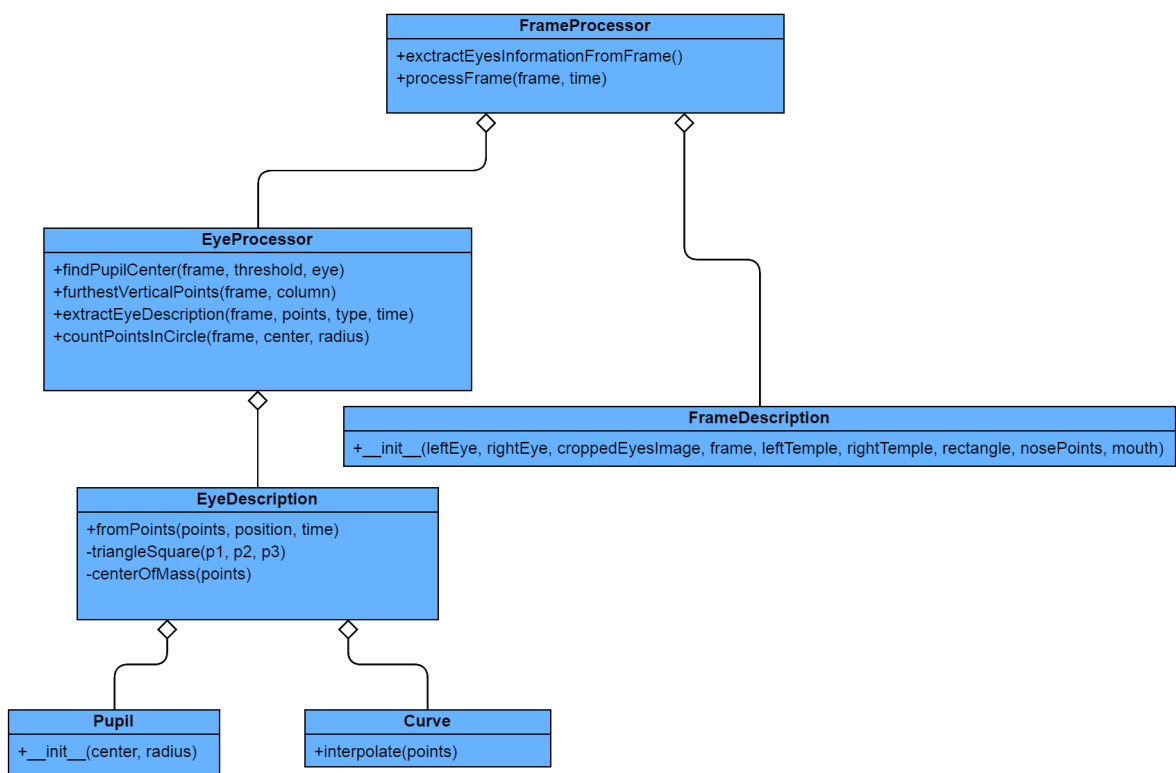


Рис. 26: Структура класу `FrameProcessor`

Клас `FrameProcessor` призначений для обробки фрейму та отримання даних про складові частини людського обличчя в визначений момент часу. Даний функціонал використовується кожний раз, коли отримано новий фрейм з відеопотоку. Результати роботи використовуються в подальшому аналізі даних. В якості вхідних даних виступає фрейм, що отриманий з відеопотоку, а результатом роботи є математична модель, що представляє собою чисельний опис складових частин

обличчя, локалізованого на фреймі. Для збільшення швидкодії та достовірності розпізнавання облич, для локалізації облич та ключових точок використовується алгоритм, який базується на діаграмі направлених градієнтів, що поєднано з лінійним методом опорних векторів. Побудова діаграми направлених векторів полягає у обробці кожного пікселя та порівняння його яскравості з яскравістю його сусідів та визначення напрямку її найбільшого зменшення. Завдяки використанню такого підходу зменшується вплив умов освітлення на достовірність класифікації. Для зменшення кількості ознак, що описують фрейм, виконується розбиття вхідного зображення на квадрати розміром 16 x 16 пікселів та обчислення домінуючого градієнту у ньому. Розпізнавання функціонального стану оператора досягається шляхом локалізації ключових елементів обличчя. За допомогою алгоритму, описаного в [131] виконується пошук 68 ключових точок, що присутні на будь-якому обличчі та є специфічними для кожної людини. Пошук обличчя на зображенні виконується за допомогою алгоритму *get\_frontal\_face\_detector* з пакету *dlib*, а ключові точки локалізуються алгоритмом *shape\_predictor* з того ж пакету. Результатом роботи алгоритму локалізації ключових точок є вектор координат, в якому кожній комірці відповідає ключова точка обличчя. Так, елементи з індексами [36; 41] описують ключові точки, що розміщені навколо лівого ока, а елементи з індексами [42; 47] - правого, точки, що розміщені вздовж спинки носу - [21; 31], а навколо роту - [48 : 70]:

```

1 leftEyePoints = (36, 41)
2 rightEyePoints = (42, 47)
3 detector = dlib.get_frontal_face_detector()
4 predictor = dlib.shape_predictor('cascades/shape_predictor_68_face_landmarks.dat'
5 )

```

Для отримання інформації про ключові точки обличчя на фреймі використовується метод *extractEyesInformationFromFrame*. За допомогою оголошеного вище детектору обчислюється масив координат ключових точок *shape*. Наступним кроком виконується отримання необхідних в подальшому ознак, серед яких:

точки носу *nosePoints*, точки навколо роту *mouthPoints*, точка на лівій та правій скронях *leftCornerPoint* та *rightCornerPoint* відповідно. За допомогою класу *EyeProcessor* виконується обчислення необхідних характеристик лівого та правого очей.

```

1 def extractEyesInformationFromFrame(self, frame, time):
2     grayscaleImage = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
3     landmarks = self.detector(grayScaleImage, 1)
4     leftEye = None
5     rightEye = None
6     leftCornerPoint = None
7     rightCornerPoint = None
8     face = None
9     nosePoints = None
10    mouthPoints = None
11    if len(landmarks) > 0:
12        face = landmarks[0]
13        if face.tl_corner().x > 0 \
14            and face.tl_corner().y > 0 \
15            and face.br_corner().x \
16            and face.br_corner().y:
17
18            shape = self.predictor(grayScaleImage, face)
19            shape = face_utils.shape_to_np(shape)
20
21            nosePoints = shape[27:31]
22            mouthPoints = shape[48:70]
23            leftCornerPoint = shape[0]
24            rightCornerPoint = shape[16]
25
26            rightEye = self.eyeProcessor.extractEyeDescription(frame,
27                shape[self.rightEyePoints[0]: self.rightEyePoints
28                    [1] + 1],
29                    'right_eye', time)
30            leftEye = self.eyeProcessor.extractEyeDescription(frame,
31                shape[self.leftEyePoints[0]: self.leftEyePoints
32                    [1] + 1],
33                    'left_eye', time)
34
35    return (leftEye, rightEye, face, leftCornerPoint,

```

```
rightCornerPoint, nosePoints, mouthPoints)
```

Аргументами методу *processFrame* є вхідне зображення та час його отримання. Використовуючи описаний вище метод *extractEyesInformationFromFrame* виконується отримання інформації про ключові точки обличчя та створення об'єкту *FrameDescription*, що містить опис обличчя. Інформація про межі обличчя заповнюється з використанням границь, обчислених за допомогою детектору облич.

```

1 def processFrame(self, frame, time):
2     eyes = self.extractEyesInformationFromFrame(frame, time)
3     leftEye = None
4     rightEye = None
5
6     croppedEyesImage = None
7     frameWithLandmarks = None
8
9     face = None
10    leftTemple = None
11    rightTemple = None
12    nosePoints = None
13    mouthPoints = None
14    if eyes[0] is not None and eyes[1] is not None:
15        leftEye = eyes[0]
16        rightEye = eyes[1]
17        leftTemple = Point(eyes[3][0], eyes[3][1])
18        rightTemple = Point(eyes[4][0], eyes[4][1])
19        nosePoints = [Point(point[0], point[1]) for point in eyes[5]]
20        mouthPoints = eyes[6]
21
22        face = (Point(eyes[2].tr_corner().y, eyes[2].bl_corner().y),
23              Point(eyes[2].bl_corner().x, eyes[2].tr_corner().x))
24    return FrameDescription(leftEye, rightEye, croppedEyesImage,
25                            frameWithLandmarks,
26                            leftTemple, rightTemple, face, nosePoints, mouthPoints)

```

За допомогою класу *EyeProcessor* виконується агрегація даних про кожне око та заповнення відповідних полів класу *EyeDescription*. Даний процес ініціюється



класом *FrameProcessor*. Метод *extractEyeDescription* в якості аргументів приймає зображення *frame*, на якому виконано локалізацію ока; вектор точок *points*, які описують верхню та нижню повіки; відомості про положення ока на обличчі *type* та час отримання фрейму *time*.

Локалізація зіниці з використанням кольорової характеристики зображення виконується за допомогою методу Оцу[117] та його реалізація з бібліотеки *skimage*. Для його застосування необхідно викоремити зображення ока з загального зображення. Регіон, що містить око описується наступним чином:

$$x_{left} = \min_x points_i$$

$$y_{left} = \min_y points_i$$

$$x_{right} = \max_x points_i$$

$$y_{right} = \max_y points_i$$

Після знаходження прямокутника, що обмежує око, наступним кроком для зменшення впливу якості вхідного зображення виконано розмиття фрейму за допомогою фільтра Гаусса. Реалізація описаної моделі наведена нижче:

```

1 xmin = min([point.x for point in (eye.lowerEyeBrove.points + eye.upperEyeBrove.
      points)])
2 xmax = max([point.x for point in (eye.lowerEyeBrove.points + eye.upperEyeBrove.
      points)])
3 ymin = min([point.y for point in (eye.lowerEyeBrove.points + eye.upperEyeBrove.
      points)])
4 ymax = max([point.y for point in (eye.lowerEyeBrove.points + eye.upperEyeBrove.
      points)])
5
6 gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
7 gray = cv2.GaussianBlur(gray, (5, 5), 0)
8 eye.eyeImage = gray[ymin: ymax, xmin: xmax]
9

```

З метою чіткого розділення пікселів ока на кластери, один з яких повністю представляє собою пікселі, що описують зіницю ока, виконується бінарнізація вхідного

зображення. Пошук центру зіниці реалізується в методі *findPupilCenter*, в якій для бінеаризації зображення використовується порогове значення, отримане за допомогою методу Оцу на попередньому кроці.

```
1 transformedImage = np.array([(y < threshold) for y in x] for x in frame])
2
```

Пошук зіниці на бінарному зображенні виконується шляхом пошуку регіону з пікселями одного кольору. Для цього, в кожному стовпці виділяються неперервні регіони з пікселями одного кольору, серед яких знаходиться мінімальне та максимальне значення. Зменшення впливу шумів, що виражаються інтервалами довжиною в 1 піксель відбувається шляхом об'єднання інтервалів. Пошук границь, що обмежують зіницю в заданому стовпці виконується в методі *furthestVerticalPoints*:

```
1 def furthestVerticalPoints(self, frame, column):
2     pointsInColumn = frame[:, column]
3     whitePointIndices = []
4     for i in range(0, len(pointsInColumn)):
5         if pointsInColumn[i]:
6             whitePointIndices.append(i)
7     if len(whitePointIndices) == 0:
8         return (0, 0)
9
10    intervals = [whitePointIndices[0]]
11    for i in range(1, len(whitePointIndices)):
12        if (whitePointIndices[i] != whitePointIndices[i - 1] + 1):
13            intervals.append(whitePointIndices[i - 1])
14            intervals.append(whitePointIndices[i])
15
16    intervals.append(whitePointIndices[len(whitePointIndices) - 1])
17
18    lowest = min(whitePointIndices)
19    highest = max(whitePointIndices)
20
21    for i in range(1, len(intervals), 2):
22        if intervals[i] != intervals[i - 1]:
23            lowest = min(lowest, intervals[i - 1])
24            highest = max(highest, intervals[i])
25
```

```

26     return (lowest, highest)
27

```

Для пошуку центру зіниці знаходяться стовпці з найбільшою кількістю пікселів білого кольору, що відповідають зіниці. При наявності декількох таких стовпців, в якості центру зіниці обирається середнє значення серед їх координат:

```

1  columnNumber = transformedImage.shape[1]
2  maxLen = 0
3  xCenter = []
4  for col in range(0, columnNumber):
5      points = self.furthestVerticalPoints(frame = transformedImage, column = col)
6      if points[1] - points[0] > maxLen:
7          maxLen = points[1] - points[0]
8          xCenter = [col]
9      elif maxLen == points[1] - points[0]:
10         xCenter.append(col)
11 xCenter = int(np.mean(xCenter))
12

```

Пошук координати на осі ординат кола та його радіусу виконується з припущенням, що для істинного центру зіниці кількість пікселів білого кольору буде максимальною.

```

1  yCenter = 0
2  rad = 0
3  maxPointsInCircle = 0
4  for y in range(0, transformedImage.shape[0]):
5      for radius in range(int(maxLen / 2 - 10), int(maxLen / 2 + 3)):
6          inCircle = self.countPointsInCircle(frame = transformedImage, \
7              center = (xCenter, y), radius = radius)
8          if inCircle > maxPointsInCircle:
9              maxPointsInCircle, yCenter, rad = inCircle, y, radius
10
11 for point in (eye.upperEyeBrove.points + eye.lowerEyeBrove.points):
12     rad = min(rad, utils.distance(Point(xCenter, yCenter), Point(point.x, point.y
13     )))
14 return Pupil(Point(xCenter, yCenter), rad)

```

Метод *countPointsInCircle* за заданими координатами центру кола та його радіусом обчислює кількість пікселів білого кольору, які належать даному колу:

```

1 def countPointsInCircle(self, frame, center, radius):
2     result = 0
3     for x in range(max(0, center[0] - radius), min(center[0] + radius, frame.
4     shape[1])):
5         for y in range(max(0, center[1] - radius), min(center[1] + radius, frame.
6         shape[0])):
7             if (frame[y, x]):
8                 result = result + 1
9
10    return result

```

Клас *EyeDescription* призначений для зберігання інформації про око. В даному об'єкті зберігається така інформація як: точки, що представляють верхню та нижню повіки, їх представлення у вигляді поліномів, площа ока, центр зіниці та її радіус. Для ініціалізації класу реалізовано метод *fromPoints*, який в якості аргументів приймає список точок *points*, що описують контури ока, відомості про положення ока на обличчі *position* та час отримання фрейму *time*:

```

1 def fromPoints(self, points, position, time):
2     pts = np.array(points, np.int32)
3     self.square = 0
4     for i in range(0, len(pts)):
5         p1 = pts[i]
6         p2 = pts[(i + 1) % len(pts)]
7         self.square = self.square + self.triangleSquare(p1, p2, (0, 0))
8
9     self.upperEyeBrove = Curve(1, 2, 3, [])
10    self.lowerEyeBrove = Curve(1, 2, 3, [])
11    self.upperEyeBrove.fromPoints([pts[0], pts[1], pts[2], pts[3]])
12    self.lowerEyeBrove.fromPoints([pts[3], pts[4], pts[5], pts[0]])
13
14    self.center = self.centerOfMass(pts)
15    self.type = position
16    self.time = time
17    self.pupil = Pupil(self.center, 5)
18

```

Обчислення орієнтованої площі трикутника реалізовано в методі `triangleSquare(p1, p2, p3)`, де  $p1, p2, p3$  - вершини трикутника.

```

1 def triangleSquare(self, p1, p2, p3):
2     return (p2[0] - p1[0]) * (p3[1] - p1[1]) - (p2[1] - p1[1]) * (p3[0] - p1[0])
3

```

За центр ока приймається точка, що є центром мас багатокутника, що є випуклою оболонкою для точок, які розміщені на вздовж повік. Так як маса даного багатокутника є однорідною на всій його площині, то в якості робочої формули доцільно використовувати формулу для обчислення центроїду, тобто середнього арифметичного координат вершин:

$$\vec{r}_m = \frac{\sum_{i=1}^n \vec{r}_i}{n}, \text{ де}$$

$\vec{r}_i$  - радіус-вектор, що задає положення точки відносно початку системи координат,  $n$  - кількість точок у багатокутнику,  $\vec{r}_m$  - результуючий радіус-вектор. Обчислення центру масу виконується в методі `centerOfMass(points)`, який в якості аргументів приймає список точок `points` заданого багатокутника:

```

1 def centerOfMass(self, points):
2     x = 0
3     y = 0
4     for point in points:
5         x = point[0] + x
6         y = point[1] + y
7
8     return Point(int(x / len(points)), int(y / len(points)))
9

```

Клас `Pupil` призначений для зберігання інформації про око, а саме його центру та радіусу в полях `center` та `radius` відповідно. Окрім того, для серіалізації об'єкту реалізовано методи `__str__` та `__repr__`:

```

1 class Pupil:
2     center = None
3     radius = None

```

```

4
5     def __init__(self, center, radius):
6         self.radius = radius
7         self.center = center
8
9     def __str__(self):
10        return '{} {}'.format(self.center, self.radius)
11
12    def __repr__(self):
13        return '{} {}'.format(self.center, self.radius)
14

```

Клас *MouthDescription* зберігає точки, що описують рот та площа рота, що обчислюється за наступною формулою:

$$MAR = \frac{||p_2 - p_1|| + ||p_3 - p_2|| + ||p_5 - p_9|| + ||p_6 - p_8||}{4 * ||p_1 - p_7||}$$

Лістинг коду класу *MouthDescription*:

```

1 class MouthDescription:
2     points = []
3     square = None
4
5     def __init__(self, points):
6         if points is not None:
7             self.points = [Point(point[0], point[1]) for point in points]
8             self.square = (utils.distance(self.points[1], self.points[12]) +
9 utils.distance(self.points[2], self.points[11]) + utils.distance(self.points
10 [4], self.points[8]) + utils.distance(self.points[5], self.points[7])) / (4 *
11 utils.distance(self.points[0], self.points[6]))

```

Результат побудови математичної моделі зберігається в класі *FrameDescription*, структура якого наступна:

```

1 class FrameDescription:
2     leftEye = None
3     rightEye = None
4     frame = None
5     faceRectangle = None
6     leftTemple = None
7     rightTemple = None

```

```

8     nose = None
9     mouth = None

```

Кожне поле класу зберігає інформацію про відповідний елемент обличчя:

- *leftEye* - об'єкт типу *EyeDescription* в якому зберігається опис лівого ока;
- *rightEye* - об'єкт типу *EyeDescription* в якому зберігається опис правого ока;
- *frame* - вхідний фрейм;
- *faceRectangle* - пара точок, що описує лву нижню та праву верхню точки, що обмежують окреслений навколо обличчя прямокутник;
- *leftTemple* точка, що описує ліву скроню;
- *rightTemple* - точка, що описує праву скроню;
- *nose* - масив точок, що знаходяться на спинці носу
- *mouth* - об'єкт типу *MouthDescription*, в якому зберігається опис роту.

Приклад опису вхідного фрейму навчальної вибірки наведено нижче:

```

▼ frameInfo = (FrameDescription) <frame.FrameDescription.FrameDescription object at 0x000001DC88357E30>
  10 croppedEyesImage = (NoneType) None
  > faceRectangle = (tuple: 2) ((194;657), (553;1016))
  10 frame = (NoneType) None
  > leftEye = (EyeDescription) Unable to get repr for <class 'frame.EyeDescription.EyeDescription'>
  > leftTemple = (Point) (560;344)
  ▼ mouth = (MouthDescription) <frame.MouthDescription.MouthDescription object at 0x000001DC88356F90>
  > points = (list: 20) [(704;534), (737;519), (769;512), (789;515), (810;510), (842;516), (872;528), (845;552), (813;563), (791;566), (769;565), (736;556),
  10 square = (float) 0.2504609407345331
  > Protected Attributes
  > nose = (list: 4) [(781;319), (783;354), (786;388), (788;424)]
  > rightEye = (EyeDescription) Unable to get repr for <class 'frame.EyeDescription.EyeDescription'>
  > rightTemple = (Point) (985;334)
  > Protected Attributes

```

Рис. 27: Приклад опису вхідного фрейму

Результатом роботи сервісу *FrameProcessor* є список об'єктів типу *FrameDescription*, що представляють собою математичну модель обличчя в кожен момент часу спостереження.

## Побудова навчальної вибірки

Клас *FeatureExtractor* призначений для побудови навчальної вибірки з використанням даних, що отримані за допомогою класу *FrameDescription*. Навчальна вибірка будується при досягненні необхідної, заздалегідь визначеної кількості фреймів. Обмеження на мінімальну кількість фреймів обумовлене видом ознак, що обчислюються. Так, як одна з навчальних вибірок містить лише інформацію про блимання та їх інтенсивність, то при малому об'єму вхідних даних результуюча вибірка матиме об'єм, недостатній для навчання класифікаторів. Таким чином, результатом роботи даного класу є пара вибірок, одна з яких описує послідовність фреймів та зміну характеристики з плином часу, а друга - інформацію про блимання та їх інтенсивність.

За допомогою методу *extractPupilMovement(dataset)* обчислюється переміщення зіниці ока відносно її попереднього положення. Припускається, що центр зіниці не змінює свого положення відносно зіниці, тому за переміщення зіниці приймається евклідова відстань між поточним та попереднім положеннями центру зіниці.

```

1 def extractPupilMovement(dataset):
2     result = [0]
3     for i in range(1, len(dataset)):
4         cur = dataset[i].leftEye.pupil.center
5         prev = dataset[i - 1].leftEye.pupil.center
6
7         result.append(cur.distance(prev))
8     return result
9

```

Визначення переміщення голови у площині, що перпендикулярна осі спостереження виконується аналогічно обчисленню переміщення зіниці. За базову точку, положення якої незмінне відносно голови прийнято кінчик носу. Рухами обличчя, що викликають переміщення носу в рамках даного методу знехтувано. Для обчислення переміщення голови реалізовано метод *extractHeadMovement(dataset)*.

```

1 def extractHeadMovement(dataset):

```



```

2     points = [item.nose[-1] for item in dataset]
3     result = [0]
4     for i in range(1, len(points)):
5         cur = points[i]
6         prev = points[i - 1]
7
8         result.append(cur.distance(prev))
9
10    return result
11

```

Нахил голови у площині, що перпендикулярна осі спостереження обчислюється з використанням інформації про ліву та праву скроні. Так, кут між прямою, що паралельна горизонтальній осі на фреймі та прямою, що проходить через дві точки наскронях описує нахил голови. Даний метод реалізовано базуючись на рівнянні прямої з кутовим коефіцієнтом:

$$y = kx + b, \text{ де}$$

$k$  - тангенс шуканого кута. Для обчислення коефіцієнту  $k$  використовується наступна формула:

$$k = \frac{y_2 - y_1}{x_2 - x_1}, \text{ де}$$

$x_1, x_2, y_1, y_2$  - відповідні координати точок. Обчислення кутового коефіцієнту  $k$  реалізовано в методі *extractHeadIncline(p1, p2)*, який в якості аргументів приймає точки  $p1, p2$ , що розміщені на прямій. Обчислення нахилу голови для вибірки виконується в методі *headInclines* з використанням методу *extractHeadIncline*.

```

1 def extractHeadIncline(p1, p2):
2     return (p2.y - p1.y) / (p2.x - p1.x)
3 def headInclines(dataset):
4     return [extractHeadIncline(item.leftTemple, item.rightTemple) for item in
5         dataset]

```

Обчислення нахилу голови у площині, що паралельна осі спостереження виконується виходячи з припущення про залежність довжини відрізка від кута спо-

стереження. Так вздовж спинки носу розміщено відрізок, що починається на кінчику носа і закінчується на переніссі. При нахилі голови у будь-якому напрямі та при стаціонарному розміщенні точки спостереження довжина даного відрізка зменшується або збільшується. Зміна довжини відрізка свідчить про нахил голови. даний функціонал реалізовано в методі *biggestDistanceOnNose*. Даний метод, серед всіх точок, що розміщені на носі знаходить дві найбільш віддалені. Ця інформація в подальшому використовується в методі *headVerticalInclines* при побудові множини ознак для навчальної вибірки.

```

1 def biggestDistanceOnNose(points):
2     dist = 0
3     for p1 in points:
4         for p2 in points:
5             dist = max(p1.distance(p2), dist)
6
7     return dist
8
9 def headVerticalInclines(dataset):
10    return [biggestDistanceOnNose(item.nose) for item in dataset]
11

```

Метод *extractEyeSquare* призначений для екстракції інформації про площу ока з вхідних даних:

```

1 def extractEyeSquare(dataset):
2     return [item.leftEye.square for item in dataset]
3

```

Метод *square\_under\_upper\_brove* використовується для обчислення площі ока під верхньою повікою за формулою:

$$\int_{x_1}^{x_2} (Ax^2 + Bx + C) - \frac{1}{2} |y_2 - y_1| * |x_2 - x_1|$$

```

1     def square_under_upper_brove(eye):
2         brove = eye.upperEyeBrove.integrate(eye.leftPoint.x, eye.rightPoint.x
3         )
4         under_center = 0.5 * (eye.leftPoint.y + eye.rightPoint.y) * (abs((eye
5         .leftPoint.x - eye.rightPoint.x)))

```

```
4     return brove - under_center
5
```

За допомогою методу *extractBlinkDuration* виконується обчислення характеристик блимання для фреймів, що представлені набором вхідних даних. Для пошуку інтервалів блимання використовується ансамбль класифікатор, що реалізований в класі *BlinkDetection*. З метою зменшення впливу нахилу голови на достовірність розпізнавання, поточний нахил голови враховується при обчисленні інтервалів блимання. Пошук інтервалів блимання виконується шляхом перевірки кожних 15 послідовних фреймів(розд. 3.3) за допомогою відповідного класифікатора. Якщо для даного інтервалу знайдено більше ніж одне блимання, то обирається блимання з мінімальним значенням при закритому оці. Емпіричним шляхом встановлено, що для класифікатора, що базується на мініальному значенні ознаки параметр порогового значення залежить від мінімального та максимального значення ознаки за період спостереження з урахуванням поправки на нахил голови. А для класифікатора, що базується на динаміці зміни величини ознаки значення амплітуди блимання залежить лише від поправки на нахил голови. Результатом пошуку інтервалів блимання є вектор, кожен елемент якого відповідає точці з мінімальним значенням для кожного з блимань.

```
1 blinkDetector = BlinkDetection()
2 nose = headVerticalInclines(dataset)
3 square = extractEyeSquare(dataset)
4 blinksMinimalValues = []
5 windowSize = 15
6 mn = 400
7 mx = 400
8 maxIncline = -1
9 lastMinPos = 0
10 minimalValue = 250
11
12 for i in range(len(square) - windowSize):
13     curIncline = nose[i]
14     coeff = maxIncline / curIncline
15     maxIncline = max(maxIncline, curIncline)
```

```

16     cur = square[i: i + windowSize]
17     mn = min(mn, min(cur))
18     mx = max(mx, max(cur))
19     threshold = mn + 0.2 * (mx - mn)
20     if i - lastMinPos >= 4000:
21         minimalValue = min(square[i - 4000: i])
22         lastMinPos = i
23     blinks = blinkDetector.detect(cur, threshold * coeff, 220 * coeff,
minimalValue)
24     total = sum([int(item > 0) for item in blinks])
25
26     if total >= 2:
27         mpos = 0
28         for pos in range(len(cur)):
29             if cur[pos] < cur[mpos]:
30                 mpos = pos
31
32     blinksMinimalValues.append(mpos + i)
33

```

Після знаходження інтервалів блимань, виконується об'єднання точок, що описують один і той же інтервал:

```

1 result = [blinksMinimalValues[0]]
2 for i in range(1, len(blinksMinimalValues)):
3     if abs(blinksMinimalValues[i] - result[-1]) > 1:
4         result.append(blinksMinimalValues[i])
5

```

Наступним кроком значення вхідної величини інтерполюється за допомогою кубічного сплайну з метою пошуку його похідної. За допомогою інформації про знак першої похідної можливо з високою точністю знаходити інтервали спадання та зростання величини. В результаті обробки вхідної послідовності з урахуванням інформації про інтервали блимання в ній будується вектор *found*, який містить інформацію про початок блимання, його кінець та точку в якій досягається мінімальне значення величини, що аналізується.

```

1 coeffs = utils.cubicSpline([i for i in range(len(square))], square)
2 derivs = []

```

```

3 for i in np.arange(0.5, len(coeffs), 1):
4     cur = coeffs[int(i)]
5     h = i - cur[4]
6     ffx = cur[1] + 2 * cur[2] * h + 3 * cur[3] * h ** 2
7     derivs.append(ffx)
8
9 found = []
10 for point in result:
11     i = point
12     while (i > 0 and derivs[i] <= 0) or (i > 1 and derivs[i - 1] <= 0):
13         if (i > 0 and derivs[i] < 0):
14             i -= 1
15         else:
16             i -= 2
17     left = i
18     i = point + 1
19     while (i < len(derivs) and derivs[i] >= 0) or (i < len(derivs) - 1 \
20         and derivs[i + 1] >= 0):
21         if (i < len(derivs) and derivs[i] > 0):
22             i += 1
23         else:
24             i += 2
25     if abs(point - left) >= 4 or abs(i - point) >= 4:
26         found.append((point, point - left, i - point))
27
28 res = [found[0]]
29 for i in range(1, len(found)):
30     cur = found[i]
31     if utils.intervalIntersection((res[-1][0] - res[-1][1], res[-1][0] + res
32 [-1][2]),
33                                     (cur[0] - cur[1], cur[0] + cur[2])) > 0:
34         res[-1] = (int((res[-1][0] + cur[0]) / 2.0), min(res[-1][1], cur[1]),
35                                                         max(res[-1][2], cur[2]))
36     else:
37         res.append(cur)

```

Фінальним кроком при обчисленні характеристик блимання є побудова вектору, що агрегує отримані дані і трансформує їх у загальноприйнятий формат. Результатом роботи даного методу є вектор, кожен елемент якого містить наступну

інформацію: початок блимання, кінець блимання, точку з мінімальним значенням величини, амплітуду зміни площі ока та амплітуду зміни значення ККП.

```

1 upperBroveA = [item.leftEye.upperEyeBrove.a for item in dataset]
2 for i in range(len(res)):
3     interval = res[i]
4     left = interval[0] - interval[1]
5     right = interval[0] + interval[2]
6     mnSquare = min(square[left:right])
7     mxSquare = max(square[left:right])
8
9     mnA = min(upperBroveA[left:right])
10    mxA = max(upperBroveA[left:right])
11
12    res[i] = (interval[0], interval[1], interval[2], mxSquare - mnSquare, mxA -
13    mnA)
14 return res
15

```

Точкою входу для даного класу є метод *extractFeatures(dataset)*, якому на вхід передається множина даних про значення ознак під час спостереження. Зі вхідних даних, за допомогою описаних вище методів відбувається обчислення значень відповідних ознак навчальної вибірки. Результати роботи методів групуються у вектори *timelineDescription* для опису характеру зміни значень ознак з плином часу та в вектор *blinksDescription* для опису характеристик блимання. Вибірку *timelineDescription* характеризують наступні ознаки: *square* - площа ока для поточного фрейму, *head\_incline* - нахил голови у площині, перпендикулярній осі спостереження, *head\_vertical\_incline* - нахил голови у площині, паралельній осі спостереження, *pupil\_movement* - переміщення зіниці відносно попереднього фрейму, *head\_movement* - переміщення голови відносно попереднього фрейму. Вибірку *blinksDescription* складається з наступних ознак: *desc* - тривалість закриття ока у фреймах, *asc* - тривалість відкриття ока у фреймах, *previous* - тривалість у фреймах від попереднього блимання, *square\_amplitude* - амплітуда зміни площі ока, *curve\_amplitude* - амплітуда зміни коефіцієнту кривизни пові-

ки. Результатом роботи даного методу є пара векторів, кожен елемент якої описує навчальну вибірку, побудовану, базуючись інформації про фрейми вхідного відео-поток.

```

1 def extractFeatures(dataset):
2     square = extractEyeSquare(dataset)
3     headIncline = headInclines(dataset)
4     headVert = headVerticalInclines(dataset)
5     pupilMovement = extractPupilMovement(dataset)
6     headMovement = extractHeadMovement(dataset)
7     blinkDuration = extractBlinkDuration(dataset)
8     timelineDescription = {
9         'square': square,
10        'head_incline': headIncline,
11        'head_vertical_incline': headVert,
12        'pupil_movement': pupilMovement,
13        'head_movement': headMovement
14    }
15
16    blinksDescription = {
17        'desc': [x[1] for x in blinkDuration],
18        'asc': [x[2] for x in blinkDuration],
19        'previous': [0] + [blinkDuration[i][0] -
20                          blinkDuration[i - 1][0] for i in range(1, len(blinkDuration))],
21        'square_amplitude': [x[3] for x in blinkDuration],
22        'curve_amplitude': [x[4] for x in blinkDuration]
23    }
24
25    return (timelineDescription, blinksDescription)
26

```

## Ансамбль класифікаторів для виявлення блимання

Клас *BlinkDetection* представляє собою ансамбль класифікаторів, який призначений для визначення моментів блимання. Даний ансамбль містить наступні класифікатори: адаптивний класифікатор, що базується на мінімальному значенні ознаки, класифікатор, що базується на динаміці зміни ознаки, класифікатор, що базується на функції непропорційності. На відміну від класичних класифіка-

торів, даний ансамбль не містить окремого методу, що відповідає етапу навчання. В класі *BlinkDetection* етап навчання і екзаммену поєднано в один метод *detect*. Метод *detect* є єдиним публічним методом даного класу, вхідними даними для якого є розмічені дані, а вихідними - набір з трьох векторів - результатів роботи відповідного класифікатору, кожен з яких складається з пар точок - моментів початку та закінчення інтервалів блимання. Точкою виклику даного ансамблю є процес обробки послідовності фреймів при досягненні кількості фреймів, достатньої для знаходження блимань. Даний клас є точкою входу для класифікаторів, що є його складовими: *BlinkIntervalDetector*, *MinimalValueBlinkDetector*, *UnpropDetector*. Структура ансамблю зображена на рис.28.

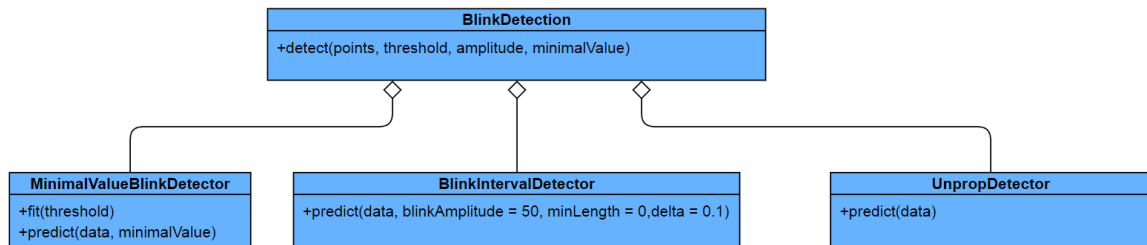


Рис. 28: Структура класу *BlinkDetection*

Класифікатор, що базується на мінімальному значенні ознаки представлений класом *MinimalValueBlinkDetector* та представляє собою примітивне порівняння вхідної величини з визначеним заздалегідь значенням. Якщо значення величини менше за порогове, то інтервал довжиною 20 фреймів (разд. 3.3), центром якого є дана точка визначається як блимання. Даний класифікатор є одним з алгоритмів з яких складається ансамбль для визначення інтервалів блимання, що реалізований в класі *BlinkDetection*. На вхід даному класифікатору передається вектор значень ознаки, що характеризує поточний стан ока. Вихідними даними є вектор, кожен елемент якого описує початок та кінець блимання, знайденого класифікатором. Даний функціонал реалізовано в функції *predict(data, minimalValue)*, яка приймає вектор вхідних значень *data* та мінімальне значення ознаки *minimalValue*:



```

1     def predict(self, data, minimalValue):
2         res = []
3         length = len(data)
4         i = 0
5         while i < length:
6             if data[i] <= minimalValue + self.threshold:
7                 res.append((max(0, i - 10), min(len(data), i + 10)))
8                 i = i + 20
9             i = i + 1
10        return res
11

```

В класі *BlinkIntervalDetector* реалізовано класифікатор, що визначає інтервали блимання з використанням інформації про динаміку зміни величини ознаки. Даний класифікатор викликається ансамблем, реалізованим в класі *BlinkDetection*. Клас *BlinkIntervalDetector* складається з єдиного методу, що призначений для пошуку інтервалів, що задовольняють певним умовам. Вхідними даними є вектор значень ознаки *data*, мінімальне значення величини амплітуди блимання *blinkAmplitude*, мінімальна довжина інтервалу блимання *minLength* та параметр *delta*, що використовується при порівнянні двох послідовних значень. Для пошуку інтервалів спадання та зростання використовуються функції *findDescendingIntervals* та *findAscendingIntervals* відповідно. За допомогою функції *concatIntervals* виконується об'єднання послідовних інтервалів спадання та зростання величини у один інтервал. Інтервал визнається таким, що містить блимання, якщо він задовольняє наступним умовам: різниця між максимальним та мінімальним значенням ознаки не менша за значення *blinkAmplitude*; довжина інтервалу більша за значення *minLength*. Параметр *delta* призначений для оцінки взаємного розміщення значень ознаки під час пошуку інтервалів спадання та зростання.

Даний функціонал реалізовано в методі *predict(data, blinkAmplitude = 50, minLength = 0, delta = 0.1)* класу *BlinkIntervalDetector*:

```

1     def predict(self, data, blinkAmplitude = 50, minLength = 0, delta = 0.1):
2         first = utils.findDescendingIntervals(data, delta)
3         second = utils.findAscendingIntervals(data, delta)

```

```

4
5     res = utils.concatIntervals(first, second)
6
7     ans = []
8     for interval in res:
9         start = interval[0][0]
10        end = interval[1][1]
11        values = data[start:end]
12        mn = min(values)
13        mx = max(values)
14        if (mx - mn > blinkAmplitude and end - start >= minLength):
15            ans.append(interval)
16    return ans

```

Аналогічно описаних вище алгоритмів, класифікатор, що базується на функції непропорційності, який входить в ансамбль з класу *BlinkDetection*, реалізовано в класі *UnpropDetector* та представлено методом *predict(data)*, де *data* - Масив вхідних даних, які використовуються при побудові функції непропорційності. Як і для класифікаторів, що реалізують аналогічний функціонал на вхід передається послідовність значень ознаки, а результатом роботи є список інтервалів, що визнані як блимання.

Для застосування функції непропорційності до дискретної величини виконується її інтерполяція за допомогою кубічного сплайну. Збільшення точності обчислення функції непропорційності досягається шляхом збільшення частоти дискретизації. Використовуючи коефіцієнти поліному обчислюємо першу та другу похідну в точці, що необхідні при визначенні значення функції непропорційності.

```

1     y = data
2     x = [i + 1 for i in range(len(data))]
3     coeffs = utils.cubicSpline(x, y)
4
5     splineData = []
6     for coef in coeffs:
7         for i in np.arange(coef[4], coef[4] + 1, 0.15):
8             h = i - coef[4]
9             fx = coef[0] + coef[1] * h + coef[2] * h ** 2 + coef[3] * h ** 3
10            ffx = coef[1] + 2 * coef[2] * h + 3 * coef[3] * h ** 2

```

```

11     fffx = 2 * coef[2] + 3 * coef[3] * h
12     splineData.append((i - 1, fx, ffx, fffx))

```

Використовуючи формулу (3.21) для кожної отриманої точки та значень функції і її похідних обчислено значення функції непропорційності.

```

1     unprop = []
2     for point in splineData:
3         val = 1 - (point[2] ** 2) / (point[1] * point[3])
4         unprop.append(val)

```

Патерн, що зображено на рис.20 математично виражається наступним чином:

$$U_i \leq U_{i-1} \leq U_{i-2} \leq U_{i-3} \leq U_{i-4} \leq U_{i-5}$$

$$U_i \leq U_{i+1}$$

Емпіричним шляхом встановлено, що значення точок, що відповідають патерну знаходяться в інтервалі  $[-1; 1.5]$ , що математично виражається наступним чином:

$$\min(U) \leq -1 \wedge \max(U) \leq 1.5$$

Для зменшення кількості хибних спрацювань додатково введено наступні умови, що визначені експериментальним шляхом:

$$||U_{i+2} - U_{i+1}| - |U_{i+2} - U_{i+3}|| \leq 0.03$$

$$||U_{i+3} - U_{i+2}| - |U_{i+4} - U_{i+3}|| \leq 0.03$$

$$||U_{i+4} - U_{i+3}| - |U_{i+5} - U_{i+4}|| \leq 0.03$$

$$|U_{i-5} - U_{i-6}| \leq 0.08$$

Отримана множина умов записується наступним чином у програмній реалізації:

```

1     found = []
2     for i in range(7, len(unprop) - 3):
3         mx = max(unprop[i - 7: i + 7])
4         mn = min(unprop[i - 7: i + 7])

```

```

5
6     if unprop[i] < unprop[i - 1] < unprop[i - 2] \
7         < unprop[i - 3] < unprop[i - 4] < unprop[i - 5] \
8         and abs(unprop[i - 5] - unprop[i - 6]) < 0.08 \
9         and unprop[i] < unprop[i + 1] \
10        and mx < 1.5 and mn > -1 \
11        and i < len(splineData) and abs(mx - mn) > 0.08 * mx \
12        and abs(abs(unprop[i + 2] - unprop[i + 1]) \
13            - abs(unprop[i + 2] - unprop[i + 3])) <= 0.03 \
14        and abs(abs(unprop[i + 3] - unprop[i + 2]) \
15            - abs(unprop[i + 4] - unprop[i + 3])) <= 0.03 \
16        and abs(abs(unprop[i + 4] - unprop[i + 3]) \
17            - abs(unprop[i + 5] - unprop[i + 4])) <= 0.03:
18        found.append(splineData[i][0])

```

Отриманий масив *found* представляє собою моменти часу, де досягається мінімальне значення патерну.

В результаті побудовано додаткову навчальну вибірку, що представляє собою послідовність пар, які описують інтервали блимання оператора. Інтервали коли око повністю відкрите в вибірку не були додані через недоцільність.

## 4.4 Визначення функціонального стану оператора

### Виявлення аномалій

Знаходження аномалій, що відбувається на етапі побудови навчальної вибірки реалізовано за допомогою класу *AnomalyDetection*, що має два методи: *fit(dataset)* для створення класифікатора з урахуванням вхідних даних *dataset* і *predict(X)* для визначення чи є *X* викидом. Точкою виклику даного класу є етап, на якому зі вхідного потоку отримано фрейм та виконується його обробка. Навчання даного класифікатора виконується кожен раз, коли в вибірці досягнута достатня кількість нових фреймів. Кожен раз, коли класифікатор навчено з використанням нових даних виконується процес пошуку аномалій, що відповідають функціональному стану, відмінному від звичайного.

Клас *AnomalyDetection* представляє собою ансамбль, що складається з трьох класифікаторів: локального рівня викиду, однокласового методу опорних векторів та IsolationForest. Ініціалізація ансамблю відбувається в методі *fit*. Для обчислення квантилей використовується функція *quantile* з пакету *numpy*. В якості класифікатору локального рівня викиду обрана реалізація *LocalOutlierFactor* з пакету *sklearn*, яка в якості параметрів приймає кількість сусідів *n\_neighbors*, що використовуються при визначенні класу та кількість аномалій *contamination* у вхідних даних, що використовується для обчислення порогового значення функції прийняття рішень.

Процес навчання ансамблю класифікаторів для визначення аномалій у вхідних даних реалізовано в методі *fit*:

```

1 def fit(data):
2     model = OneClassSVM(kernel = 'rbf', gamma = 0.1, nu=0.5).fit(data)
3     model_LOF = LocalOutlierFactor(n_neighbors=25, contamination=0.1, novelty=
4     True, algorithm = 'brute', leaf_size=10)
5     model_LOF.fit(data)
6     forest = IsolationForest(contamination=0.1, random_state=42)
7     forest.fit(data)
8     return [model, model_LOF, forest]

```

Параметри кожного з класифікаторів підбрані емпіричним шляхом, методом тестування на різноманітних вхідних даних.

Сформований ансамбль використовується для визначення функціонального стану оператора шляхом виявлення аномалій вхідних даних. Виявлення аномалій виконується шляхом голосування ансамблю класифікаторів. Так як кількість класифікаторів непарна, то ансамбль завжди досягає консенсусу.

Визначення класу для заданої величини *data* на основі ансамблю *classifiers* реалізовано в методі *predict*:

```

1 def most_common(lst):
2     return max(set(lst), key=lst.count)
3
4 def predict(classifiers, data):

```

```

5     res = [clf.predict(data) for clf in classifiers]
6     return most_common(res)
7

```

Результатом роботи даного ансамблю є значення параметрів організму оператора для вибраного фрейму. В подальшому ці дані використовуються для визначення функціонального стану оператора на інтервалі часу.

Для визначення поточного стану оператора необхідно в кожен момент часу швидко опрацювати запит наступного вигляду: "скільки аномалій було серед останніх N фреймів". З метою зберігання інформації про останні N фреймів було реалізовано клас циклічної черги *CyclicQueue*:

```

1 class CyclicQueue:
2     data = []
3     head = 0
4     tail = 0
5     filled = 0
6     def __init__(self, size):
7         self.data = [None] * size
8
9     def put(self, data):
10        self.filled += 1
11        self.data[self.tail] = data
12        self.tail = (self.tail + 1) % len(self.data)
13
14    def get(self):
15        return self.data[self.head]
16
17    def pop(self):
18        res = self.data[self.head]
19        self.data[self.head] = None
20        self.head = (self.head + 1) % len(self.data)
21        return res
22
23    def size(self):
24        if self.head <= self.tail:
25            return self.tail - self.head
26        else:
27            return len(self.data) + self.tail - self.head

```

```

28
29     def elements(self):
30         if self.head < self.tail:
31             return self.data[self.head : self.tail]
32         else:
33             return self.data[self.head:] + self.data[:self.tail]
34
35     def getFilled(self):
36         return self.filled
37
38     def getValueCount(self, value):
39         return len([1 for val in self.data if val == value])
40

```

Наведена вище структура даних дозволяє зберігати дані про останні  $N$  фреймів. При цьому, у випадку переповнення, інформація про початкові фрейми видаляється. Метод *getValueCount* дозволяє обчислювати кількість елементів з заданим значенням. Таким чином, для визначення кількості аномалій необхідно виконати наступний код:

```

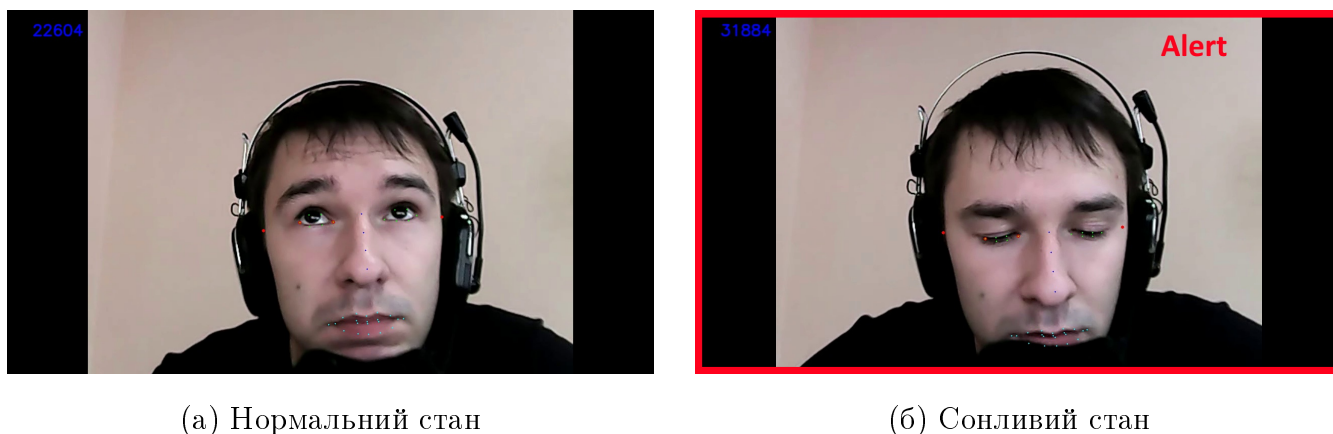
1 density = frameDensity.getValueCount(-1) / DENSITY_QUEUE_LENGTH
2

```

У випадку, якщо отримане значення перевищує порогове, то система класифікує поточний стан оператора як сонливий і застосовує відповідний алгоритм дій.

## 4.5 Результати фізичного моделювання

Для тестування програмного додатку побудовано пристрій на базі Raspberry Pi 3 зі штатно веб-камерою. Спостереження велося за оператором персонального комп'ютера протягом 20 хвилин, з яких 15 хвилин оператор перебував у бадьорому стані, а протягом 5 хвилин відбувалося чергування станів дрімоти та сну. Загальна кількість фреймів, з яких складається вибірка - 37800. Приклад



(а) Нормальний стан

(б) Сонливий стан

Рис. 29: Візуалізація роботи програмного додатку

Перші 17 хвилин спостереження система визначала стан оператора який нормальний, але після цього було перевищене порогове значення кількості фреймів, що описують сонливий стан і система класифікувала цей стан як сонливий, що зображено на рис.29 (б). Під час процесу моделювання побудовано наступний графік зміни характеристики EAR з часом:

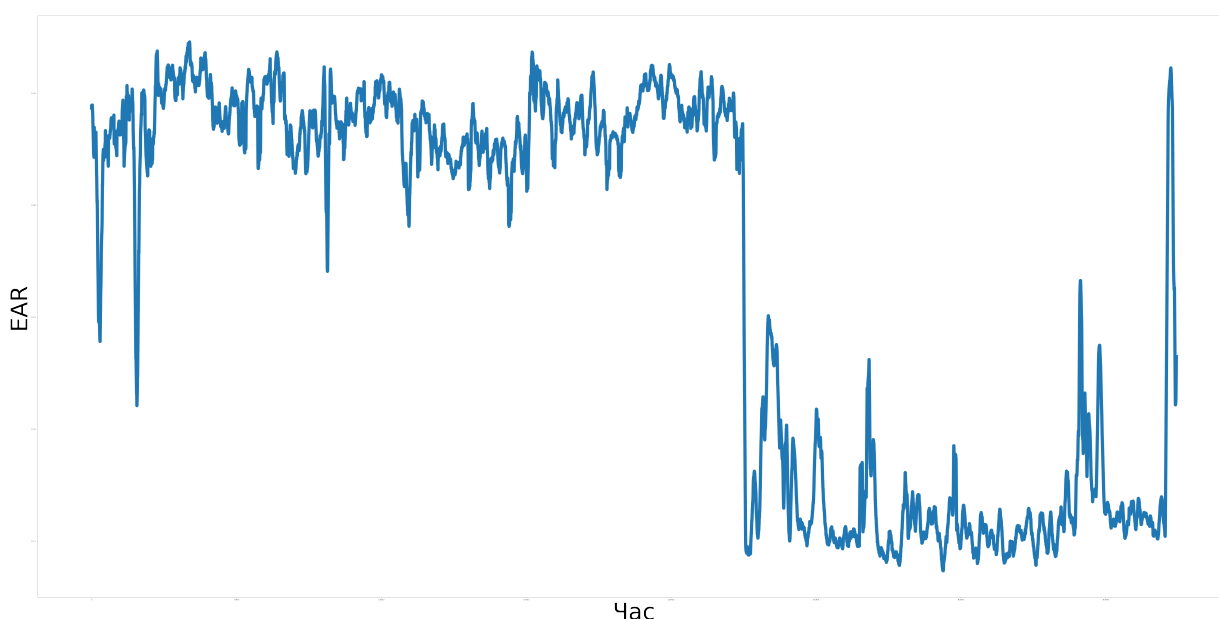


Рис. 30: Зміна характеристики EAR з часом

Даний графік підтверджує припущення про взаємозв'язок характеристики EAR та функціонального стану оператора.

Під час тестування отримано задовільний результат: функціональний стан оператора було визначено точно, були відсутні хибні спрацювання, вхідні дані опра-



цьовувалися швидко і вимагали малих обчислювальних ресурсів. В той же час для визначення поточного стану як сонливого вимагає від системи значної кількості спостережень.

## Висновки

В ході виконання роботи було проаналізовано існуючі технології та підходи до визначення функціонального стану оператора. Виконано огляд і детальний аналіз параметрів організму, що дозволяють визначити функціональний стан оператора. Дані параметри було розділено на групи і виділено лише ті, які можливо отримати шляхом візуального спостереження.

Побудовано систему, що здатна до самонавчання та адаптації до персональних характеристик оператора. Дана система не є ресурсоємкою може базуватися на більшості існуючих платформ. Отримана інформаційна система є нечутливою до умов спостереження і не потребує специфічного обладнання.

Побудова навчальної вибірки для функціонування системи виконується безпосереднім спостереженням за станом оператора протягом перших 4-х хвилин роботи. Визначення функціонального стану оператора відбувається шляхом виявлення аномалій вхідних.

Побудована система має наступні точнісні характеристики функціонування:

- $Precision = 0.975$
- $Accuracy = 0.926$
- $Recall = 0.886$
- $f1-score = 0.928$

Таким чином створена інформаційна система з визначеними оптимальними параметрами дозволяє проводити визначення функціонального стану оператора з високою точністю, та може бути використана в подальшому як додатковий елемент для безпосереднього контролю за пильністю оператору.

## Список використаних джерел

- [1] Sleep state switching / Clifford B Saper, Patrick M Fuller, Nigel P Pedersen [и др.] // *Neuron*. 2010. Т. 68, № 6. С. 1023–1042.
- [2] A molecular mechanism regulating rhythmic output from the suprachiasmatic circadian clock / Xiaowei Jin, Lauren P Shearman, David R Weaver [и др.] // *Cell*. 1999. Т. 96, № 1. С. 57–68.
- [3] Diekelmann Susanne, Born Jan. The memory function of sleep // *Nature Reviews Neuroscience*. 2010. Т. 11, № 2. с. 114.
- [4] Williams Robert L, Agnew Harman W, Webb Wilse B. Sleep patterns in young adults: an EEG study // *Electroencephalography and clinical neurophysiology*. 1964. Т. 17, № 4. С. 376–381.
- [5] Hauri Peter J, Wisbey Joyce. Wrist actigraphy in insomnia // *Sleep*. 1992. Т. 15, № 4. С. 293–301.
- [6] Rowley Jason T, Stickgold Robert, Hobson J Allan. Eyelid movements and mental activity at sleep onset // *Consciousness and cognition*. 1998. Т. 7, № 1. С. 67–84.
- [7] Jiao Yingying, Lu Bao-Liang. Detecting slow eye movement for recognizing driver's sleep onset period with EEG features // *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the / IEEE*. 2016. С. 4658–4661.
- [8] Митькин Александр Александрович. Электроокулография в инженерно-психологических исследованиях. Наука, 1974.
- [9] Meyer Jay J. Herpes zoster ophthalmicus. 1929.
- [10] Marmor Michael F. Clinical electrophysiology of the retinal pigment epithelium // *Documenta ophthalmologica*. 1991. Т. 76, № 4. С. 301–313.

- [11] A method of EOG signal processing to detect the direction of eye movements / Manuel Merino, Octavio Rivera, Isabel Gómez [и др.] // Sensor Device Technologies and Applications (SENSORDEVICES), 2010 First International Conference on / IEEE. 2010. С. 100–105.
- [12] Гришин Олег Витальевич, Гришин Виктор Григорьевич, Смирнов Сергей Валерьевич. Актиграфия в диагностике засыпания человека-оператора // Сибирский научный медицинский журнал. 2012. Т. 32, № 2.
- [13] Sleep and cardiac rhythm in healthy men / Matti Viitasalo, Liisa Halonen, Markku Partinen [и др.] // Annals of medicine. 1991. Т. 23, № 2. С. 135–139.
- [14] Different heart rate patterns in obstructive apneas during NREM sleep / Maria R Bonsignore, Salvatore Romano, Oreste Marrone [и др.] // Sleep. 1997. Т. 20, № 12. С. 1167–1174.
- [15] Hori Tadao, Hayashi Mitsuo, Morikawa Toshio. Topographical EEG changes and the hypnagogic experience. 1994.
- [16] Blake Helen, Gerard Ralph W, Kleitman Nathaniel. Factors influencing brain potentials during sleep // Journal of Neurophysiology. 1939. Т. 2, № 1. С. 48–60.
- [17] White DAVID P, Weil JOHN V, Zwillich CLIFFORD W. Metabolic rate and breathing during sleep // Journal of Applied Physiology. 1985. Т. 59, № 2. С. 384–391.
- [18] Respiration during sleep in normal man. / N J Douglas, D P White, C K Pickett [и др.] // Thorax. 1982. Т. 37, № 11. С. 840–844. URL: <https://thorax.bmj.com/content/37/11/840>.

- [19] Breathing during sleep in normal middle-aged subjects / Jean Krieger, Nelly Maglasiu, Emilia Sforza [и др.] // Sleep. 1990. Т. 13, № 2. С. 143–154.
- [20] Musculoskeletal symptoms and non-REM sleep disturbance in patients with "fibrositis syndrome" and healthy subjects. / Harvey Moldofsky, Phillip Scarisbrick, Robert England [и др.] // Psychosomatic medicine. 1975.
- [21] Hobson J Allan, Pace-Schott Edward F, Stickgold Robert. Dreaming and the brain: toward a cognitive neuroscience of conscious states // Behavioral and brain sciences. 2000. Т. 23, № 6. С. 793–842.
- [22] Tart Charles T. Patterns of basal skin resistance during sleep // Psychophysiology. 1967. Т. 4, № 1. С. 35–39.
- [23] Monroe Lawrence J. Psychological and physiological differences between good and poor sleepers. // Journal of Abnormal Psychology. 1967. Т. 72, № 3. с. 255.
- [24] LACK LEON, Lushington Kurt. The rhythms of human sleep propensity and core body temperature // Journal of sleep research. 1996. Т. 5, № 1. С. 1–11.
- [25] Leproult Rachel, Van Cauter Eve. Role of sleep and sleep loss in hormonal release and metabolism // Pediatric Neuroendocrinology. Karger Publishers, 2010. Т. 17. С. 11–21.
- [26] A novel pupil detection algorithm for infrared eye image / H. Qin, X. Wang, M. Liang [и др.] // 2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013). 2013. Aug. С. 1–5.
- [27] Szczepański Adam, Misztal Krzysztof, Saeed Khalid. Pupil and iris detection algorithm for near-infrared capture devices // IFIP International Conference on Computer Information Systems and Industrial Management / Springer. 2014. С. 141–150.

- [28] Wang Jianzhong, Zhang Guangyue, Shi Jiadong. Pupil and glint detection using wearable camera sensor and near-infrared LED array // *Sensors*. 2015. T. 15, № 12. C. 30126–30141.
- [29] Johns Murray W. A New Method for Measuring Daytime Sleepiness: The Epworth Sleepiness Scale // *Sleep*. 1991. 11. T. 14, № 6. C. 540–545. URL: <https://doi.org/10.1093/sleep/14.6.540>.
- [30] Stanford Sleepiness Scale (SSS) / Azmeh Shahid, Kate Wilkinson, Shai Marcu [и др.] // *STOP, THAT and One Hundred Other Sleep Scales* / под ред. Azmeh Shahid, Kate Wilkinson, Shai Marcu [и др.]. New York, NY: Springer New York, 2012. C. 369–370. URL: [https://doi.org/10.1007/978-1-4419-9893-4\\_91](https://doi.org/10.1007/978-1-4419-9893-4_91).
- [31] Åkerstedt Torbjörn, Gillberg Mats. Subjective and Objective Sleepiness in the Active Individual // *International Journal of Neuroscience*. 1990. T. 52, № 1-2. C. 29–37.
- [32] Development of a fatigue scale / Trudie Chalder, G Berelowitz, Teresa Pawlikowska [и др.] // *Journal of psychosomatic research*. 1993. T. 37, № 2. C. 147–153.
- [33] Goldberg David P. User's guide to the General Health Questionnaire // Windsor. 1988.
- [34] Likert Rensis. A technique for the measurement of attitudes. // *Archives of psychology*. 1932.
- [35] Choi In-Ho, Tran Thi Bao Hanh, Kim Yong-Guk. Real-time categorization of driver's gaze zone and head pose using the convolutional neural network // *Proceedings of HCI Korea / Hanbit Media, Inc.* 2016. C. 417–422.
- [36] Visual object tracking using adaptive correlation filters / David S Bolme,

- J Ross Beveridge, Bruce A Draper [и др.] // Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on / IEEE. 2010. С. 2544–2550.
- [37] Cyganek Bogusław, Gruszczyński Sławomir. Hybrid computer vision system for drivers' eye recognition and fatigue monitoring // Neurocomputing. 2014. Т. 126. С. 78–94.
- [38] Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012. С. 1097–1105.
- [39] Sun Xinghua, Xu Lu, Yang Jingyu. Driver fatigue alarm based on eye detection and gaze estimation // MIPPR 2007: Automatic Target Recognition and Image Analysis; and Multispectral Image Acquisition / International Society for Optics and Photonics. Т. 6786. 2007. с. 678612.
- [40] of Motor Carrier Research United States. Federal Highway Administration. Office, Standards. PERCLOS, a Valid Psychophysiological Measure of Alertness as Assessed by Psychomotor Vigilance. Federal Highway Administration, Office of Motor Carriers, Office of Motor Carrier Research and Standards, 1998. URL: [https://books.google.com.ua/books?id=\\_fFXugEACAAJ](https://books.google.com.ua/books?id=_fFXugEACAAJ).
- [41] Efficient PERCLOS and gaze measurement methodologies to estimate driver attention in real time / Sanka Darshana, Dileepa Fernando, Sadari Jayawardena [и др.] // Intelligent Systems, Modelling and Simulation (ISMS), 2014 5th International Conference on / IEEE. 2014. С. 289–294.
- [42] Zivkovic Zoran, Verbeek Jakob. Transformation invariant component analysis for binary images // Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on / IEEE. Т. 1. 2006. С. 254–259.

- [43] Петров Сергій Олександрович, Марченко ІО, Дібров БО. Дослідження застосування операторів згортки в задачах виділення границь на зображенні // Восточно-Европейский журнал передовых технологий. 2015. № 6 (4). С. 27–31.
- [44] Drowsy driver detection through facial movement analysis / Esra Vural, Mujdat Cetin, Aytul Ercil [и др.] // International Workshop on Human-Computer Interaction / Springer. 2007. С. 6–18.
- [45] Automated drowsiness detection for improved driving safety / Esra Vural, Müjdat Çetin, Aytül Erçil [и др.]. 2008.
- [46] Friesen E, Ekman P. Facial action coding system: a technique for the measurement of facial movement // Palo Alto. 1978.
- [47] Методи комп'ютерної обробки зображень / Михайл Валерьевич Гашников, Николай Иванович Глумов, НЮ Ильясова [и др.]. Общество с ограниченной ответственностью Издательская фирма "Физико-математическая литература 2003.
- [48] Driver drowsiness detection system based on feature representation learning using various deep networks / Sanghyuk Park, Fei Pan, Sunghun Kang [и др.] // Asian Conference on Computer Vision / Springer. 2016. С. 154–164.
- [49] Deep face recognition. / Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman [и др.] // BMVC. Т. 1. 2015. с. 6.
- [50] Long-term recurrent convolutional networks for visual recognition and description / Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama [и др.] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. С. 2625–2634.
- [51] Vision-based drowsiness detector for real driving conditions / I Garcia,



- Sebastian Bronte, Luis Miguel Bergasa [и др.] // Intelligent Vehicles Symposium (IV), 2012 IEEE / IEEE. 2012. С. 618–623.
- [52] Цыплаков АА. Введение в моделирование в пространстве состояний // Квантиль. 2011. Т. 9. С. 1–24.
- [53] Detection and prediction of driver drowsiness using artificial neural network models / Charlotte Jacobé de Naurois, Christophe Bourdin, Anca Stratulat [и др.] // Accident Analysis & Prevention. 2017.
- [54] Driver fatigue detection based on eye tracking and dynamk, template matching / Wen-Bing Horng, Chih-Yuan Chen, Yi Chang [и др.] // Networking, Sensing and Control, 2004 IEEE International Conference on / IEEE. Т. 1. 2004. С. 7–12.
- [55] Driver sleepiness detection system based on eye movements variables / Lisheng Jin, Qingning Niu, Yuying Jiang [и др.] // Advances in Mechanical Engineering. 2013. Т. 5. с. 648431.
- [56] Picot Antoine, Caplier Alice, Charbonnier Sylvie. Comparison between EOG and high frame rate camera for drowsiness detection // 2009 Workshop on Applications of Computer Vision (WACV) / IEEE. 2009. С. 1–6.
- [57] Dingus Thomas A, Hardee H Lenora, Wierwille Walter W. Development of models for on-board detection of driver impairment // Accident Analysis & Prevention. 1987. Т. 19, № 4. С. 271–283.
- [58] Spindler Christiane, Driewer Frauke, Schäfers Lorenz. Analysis of the Robustness of Steering Pattern Based Drowsiness Detection // 24th International Technical Conference on the Enhanced Safety of Vehicles (ESV). 2015. С. 1–11.
- [59] Takei Yoshihiro, Furukawa Yoshimi. Estimate of driver's fatigue through steering motion // Systems, Man and Cybernetics, 2005 IEEE International Conference on / Ieee. Т. 2. 2005. С. 1765–1770.

- [60] Friedrichs Fabian, Yang Bin. Drowsiness monitoring by steering and lane data based features under real driving conditions // Signal Processing Conference, 2010 18th European / IEEE. 2010. С. 209–213.
- [61] Kecklund Göran, Åkerstedt Torbjörn. Sleepiness in long distance truck driving: an ambulatory EEG study of night driving // Ergonomics. 1993. Т. 36, № 9. С. 1007–1017.
- [62] Eoh Hong J, Chung Min K, Kim Seong-Han. Electroencephalographic study of drowsiness in simulated driving with sleep deprivation // International Journal of Industrial Ergonomics. 2005. Т. 35, № 4. С. 307–320.
- [63] Campagne Aurelie, Pebayle Thierry, Muzet Alain. Correlation between driving errors and vigilance level: influence of the driver's age // Physiology & behavior. 2004. Т. 80, № 4. С. 515–524.
- [64] Development of an algorithm for an EEG-based driver fatigue countermeasure / Saroj KL Lal, Ashley Craig, Peter Boord [и др.] // Journal of safety Research. 2003. Т. 34, № 3. С. 321–328.
- [65] Введение в психологию / РЛ Аткинсон, РС Аткинсон, ЭЕ Смит [и др.] // М.: Прайм-Еврознак. 2003.
- [66] Monitoring sleepiness with on-board electrophysiological recordings for preventing sleep-deprived traffic accidents / Christos Papadelis, Zhe Chen, Chrysoula Kourtidou-Papadeli [и др.] // Clinical Neurophysiology. 2007. Т. 118, № 9. С. 1906–1922.
- [67] Jasper Herbert H. The ten-twenty electrode system of the International Federation // Electroencephalogr. Clin. Neurophysiol. 1958. Т. 10. С. 370–375.
- [68] Shabani Hamed, Mikaili Mohammad, Noori Seyed Mohammad Reza. Assessment

- of recurrence quantification analysis (RQA) of EEG for development of a novel drowsiness detection system // *Biomedical Engineering Letters*. 2016. Aug. T. 6, № 3. С. 196–204. URL: <https://doi.org/10.1007/s13534-016-0223-5>.
- [69] Wireless and Wearable EEG System for Evaluating Driver Vigilance / C. Lin, C. Chuang, C. Huang [и др.] // *IEEE Transactions on Biomedical Circuits and Systems*. 2014. April. T. 8, № 2. С. 165–176.
- [70] Li G., Lee B., Chung W. Smartwatch-Based Wearable EEG System for Driver Drowsiness Detection // *IEEE Sensors Journal*. 2015. Dec. T. 15, № 12. С. 7169–7180.
- [71] A vehicle active safety model: vehicle speed control based on driver vigilance detection using wearable EEG and sparse representation / Zutao Zhang, Dianyuan Luo, Yagubov Rasim [и др.] // *Sensors*. 2016. T. 16, № 2. с. 242.
- [72] A real-time wireless brain–computer interface system for drowsiness detection / Chin-Teng Lin, Che-Jui Chang, Bor-Shyh Lin [и др.] // *IEEE transactions on biomedical circuits and systems*. 2010. T. 4, № 4. С. 214–222.
- [73] Brain dynamics in predicting driving fatigue using a recurrent self-evolving fuzzy neural network / Yu-Ting Liu, Yang-Yin Lin, Shang-Lin Wu [и др.] // *IEEE transactions on neural networks and learning systems*. 2016. T. 27, № 2. С. 347–360.
- [74] Multivariate autoregressive models and kernel learning algorithms for classifying driving mental fatigue based on electroencephalographic / Chunlin Zhao, Chongxun Zheng, Min Zhao [и др.] // *Expert Systems with Applications*. 2011. T. 38, № 3. С. 1859–1865.
- [75] Lee Boon-Giin, Lee Boon-Leng, Chung Wan-Young. Mobile healthcare for automatic driving sleep-onset detection using wavelet-based EEG and respiration signals // *Sensors*. 2014. T. 14, № 10. С. 17915–17936.

- [76] Webber Charles, Marwan Norbert. Recurrence Quantification Analysis – Theory and Best Practices. 2015. 01. с. 421.
- [77] Eckmann J-P, Kamphorst S Oliffson, Ruelle David. Recurrence plots of dynamical systems // EPL (Europhysics Letters). 1987. Т. 4, № 9. с. 973.
- [78] Support vector machines / Marti A. Hearst, Susan T Dumais, Edgar Osuna [и др.] // IEEE Intelligent Systems and their applications. 1998. Т. 13, № 4. С. 18–28.
- [79] Russell Stuart J, Norvig Peter. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited,, 2016.
- [80] Altman Naomi S. An introduction to kernel and nearest-neighbor nonparametric regression // The American Statistician. 1992. Т. 46, № 3. С. 175–185.
- [81] Cohen Patricia, West Stephen G, Aiken Leona S. Applied multiple regression/correlation analysis for the behavioral sciences. Psychology Press, 2014.
- [82] Franc Vojtech, Zien Alexander, Schölkopf Bernhard. Support Vector Machines as Probabilistic Models. // ICML. 2011. С. 665–672.
- [83] Fourier Joseph. Theorie analytique de la chaleur, par M. Fourier. Chez Firmin Didot, père et fils, 1822.
- [84] Welch Peter. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms // IEEE Transactions on audio and electroacoustics. 1967. Т. 15, № 2. С. 70–73.
- [85] K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation / Michal Aharon, Michael Elad, Alfred Bruckstein [и др.] // IEEE Transactions on signal processing. 2006. Т. 54, № 11. с. 4311.

- [86] Mallat Stéphane. A wavelet tour of signal processing. Elsevier, 1999.
- [87] Elad Michael. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. 1st изд. Springer Publishing Company, Incorporated, 2010.
- [88] Penny W, Harrison L. Multivariate autoregressive models // Statistical Parametric Mapping: The Analysis of Functional Brain Images, eds K. J. Friston, JT Ashburner, SJ Kiebel, TE Nichols, and WD Penny (London: Elsevier). 2007. C. 534–540.
- [89] Pearson Karl. LIII. On lines and planes of closest fit to systems of points in space // The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1901. T. 2, № 11. C. 559–572.
- [90] Li Pinyi, Jiang Wenhui, Su Fei. Single-channel EEG-based mental fatigue detection based on deep belief network // Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the / IEEE. 2016. C. 367–370.
- [91] Pincus Steven M, Gladstone Igor M, Ehrenkranz Richard A. A regularity statistic for medical data analysis // Journal of clinical monitoring. 1991. T. 7, № 4. C. 335–345.
- [92] Coifman Ronald R, Wickerhauser M Victor. Entropy-based algorithms for best basis selection // IEEE Transactions on information theory. 1992. T. 38, № 2. C. 713–718.
- [93] Mobile ECG-based drowsiness detection / Nipun Gupta, Dina Najeeb, Victor Gabrielian [и др.] // Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual / IEEE. 2017. C. 29–32.
- [94] Elsenbruch Sigrid, Harnish Michael J, Orr William C. Heart rate variability

during waking and sleep in healthy males and females // *Sleep*. 1999. T. 22, № 8. С. 1067–1071.

- [95] Drowsiness detection using heart rate variability / José Vicente, Pablo Laguna, Ariadna Bartra [и др.] // *Medical & biological engineering & computing*. 2016. T. 54, № 6. С. 927–937.
- [96] The integral pulse frequency modulation model with time-varying threshold: application to heart rate variability analysis during exercise stress testing / Raquel Bailón, Ghailen Laouini, César Grao [и др.] // *IEEE transactions on biomedical engineering*. 2011. T. 58, № 3. С. 642–652.
- [97] Mateo Javier, Laguna Pablo. Analysis of heart rate variability in the presence of ectopic beats using the heart timing signal // *IEEE Transactions on Biomedical Engineering*. 2003. T. 50, № 3. С. 334–343.
- [98] Rahim Herlina Abdul, Dalimi Ahmad, Jaafar Haliza. Detecting drowsy driver using pulse sensor // *J. Technol.* 2015. T. 73, № 3. С. 5–8.
- [99] Drowsiness detection based on the analysis of breathing rate obtained from real-time image recognition / José Solaz, José Laparra-Hernández, Daniel Bande [и др.] // *Transportation Research Procedia*. 2016. T. 14. С. 3867–3876.
- [100] Detecting driver drowsiness using wireless wearables / Brandy Warwick, Nicholas Symons, Xiao Chen [и др.] // *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on / IEEE*. 2015. С. 585–588.
- [101] Synchronizing Index for Detecting Drowsiness Based on the Respiratory Effort Signal / N Rodríguez-Ibáñez, MA García-González, Mireya Fernández-Chimeno [и др.] // *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013 / Springer*. 2014. С. 965–968.

- [102] Bundele Mahesh M, Banerjee Rahul. Detection of fatigue of vehicular driver using skin conductance and oximetry pulse: a neural network approach // Proceedings of the 11th International Conference on Information Integration and web-based applications & services / ACM. 2009. С. 739–744.
- [103] Телемеханическая система контроля бодрствования машиниста / Александр Степанович Бугаёв, Сергей Валерьянович Герус, Валерий Васильевич Дементиенко [и др.] // Бюллетень Объединенного ученого совета ОАО РЖД. 2017. № 2. С. 21–41.
- [104] Дементиенко ВВ, Дорохов ВВ. Оценка эффективности систем контроля уровня бодрствования человека-оператора с учетом вероятностной природы возникновения ошибок при засыпании // Журнал высшей нервной деятельности им. ИП Павлова. 2013. Т. 63, № 1. С. 24–24.
- [105] Viola Paul, Jones Michael. Rapid object detection using a boosted cascade of simple features // Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on / IEEE. Т. 1. 2001. С. I–I.
- [106] Марченко Ігор Олександрович, Петров СО, Лисак НВ. Визначення впливу кольорової характеристики об'єкта на оперативність роботи алгоритму Віоли–Джонса на прикладі локалізації облич // Вісник Вінницького політехнічного інституту. 2016. № 1. С. 108–114.
- [107] Singular value decomposition / L De Lathauwer, B De Moor, J Vandewalle [и др.] // Proc. EUSIPCO-94, Edinburgh, Scotland, UK. Т. 1. 1994. С. 175–178.
- [108] Soukupová Tereza, Cech Jan. Real-time eye blink detection using facial landmarks // 21st Computer Vision Winter Workshop. 2016.

- [109] Nakayama M., Yamamoto K., Kobayashi F. Estimation of Sleepiness Using Pupillary Response and Its Frequency Components // *Int. J. Bioinformatics Res. Appl.* 2012. T. 8, № 5/6. C. 342–365. URL: <http://dx.doi.org/10.1504/IJBRA.2012.049621>.
- [110] LOWENSTEIN OTTO, FEINBERG RICHARD, LOEWENFELD IRENE E. Pupillary Movements During Acute and Chronic Fatigue :A New Test for the Objective Evaluation of Tiredness // *Investigative Ophthalmology and Visual Science.* 1963. T. 2, № 2. c. 138. URL: + <http://dx.doi.org/>.
- [111] Kazemi Vahid, Sullivan Josephine. One millisecond face alignment with an ensemble of regression trees // *2014 IEEE Conference on Computer Vision and Pattern Recognition.* 2014. C. 1867–1874.
- [112] Booth E.G. Hydrologic variability of the Cosumnes River floodplain. University of California, Davis, 2006. URL: <https://books.google.com.ua/books?id=vroBGwAACAAJ>.
- [113] Kim H., Cha J., Lee W. D. Eye Detection for Gaze Tracker with Near Infrared Illuminator // *2014 IEEE 17th International Conference on Computational Science and Engineering.* 2014. Dec. C. 458–464.
- [114] Youmaran R., Adler A. Using infrared illumination to improve eye and face tracking in low quality video images // *2008 24th Biennial Symposium on Communications.* 2008. June. C. 391–394.
- [115] Ghosh Sayani, Nandy Tanaya, Manna Nilotpai. Real time eye detection and tracking method for driver assistance system // *Advancements of Medical Electronics.* Springer, 2015. C. 13–25.
- [116] Zhu Zhiwei, Fujimura Kikuo, Ji Qiang. Real-time eye detection and tracking under various light conditions // *Proceedings of the 2002 symposium on Eye tracking research & applications / ACM.* 2002. C. 139–144.



- [117] Otsu N. A Threshold Selection Method from Gray-Level Histograms // IEEE Transactions on Systems, Man, and Cybernetics. 1979. Jan. Т. 9, № 1. С. 62–66.
- [118] Краснополюсовський АС. Інформаційний синтез інтелектуальних систем керування: Підхід, що ґрунтується на методі функціонально-статистичних випробувань // Суми: Видавництво СумДУ. 2004. Т. 261.
- [119] Powers David. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness and Correlation. 2008. 01. Т. 2.
- [120] Sasaki Yutaka. The truth of the F-measure. 2007. 01.
- [121] Drutarovsky Tomas, Fogelton Andrej. Eye Blink Detection Using Variance of Motion Vectors // ECCV Workshops. 2014.
- [122] A practical guide to splines / Carl De Boor, Carl De Boor, Etats-Unis Mathématicien [и др.]. Springer-Verlag New York, 1978. Т. 27.
- [123] Решение задач технической диагностики и сравнения образов с помощью оценки отличия зависимости между двумя функциями от пропорциональной / Виктор Васильевич Авраменко, Анжела Петровна Карпенко, Виктор Васильевич Авраменко [и др.]. 1999.
- [124] Авраменко Віктор Васильович, Авраменко Віктор Васильович. Характеристики непропорциональності числових функцій и их применение при решении задач диагностики. 2000.
- [125] Авраменко Віктор Васильович, Волков Роман Сергеевич, Карпенко Анжела Петровна. Метод розпізнавання сигналу с произвольними масштабами по осям координат относительно соответствующего ему эталона // Автоматизированные системы управления и приборы автоматики. 2012. № 160.
- [126] Joyce James. Bayes' Theorem // The Stanford Encyclopedia of Philosophy / под ред. Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2019.

- [127] Rossi Richard J. Mathematical statistics: an introduction to likelihood based inference. John Wiley & Sons, 2018.
- [128] Nair Vinod, Hinton Geoffrey E. Rectified linear units improve restricted boltzmann machines // ICML. 2010.
- [129] Mira José, Sandoval Francisco. From Natural to Artificial Neural Computation: International Workshop on Artificial Neural Networks, Malaga-Torremolinos, Spain, June 7-9, 1995: Proceedings. Springer Science & Business Media, 1995. T. 930.
- [130] Steinberg Dan, Colla Phillip. CART: classification and regression trees // The top ten algorithms in data mining. 2009. T. 9. c. 179.
- [131] Kazemi V., Sullivan J. One millisecond face alignment with an ensemble of regression trees // 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014. June. C. 1867–1874.

# Додатки

## Приклад опису обличчя

```
1 {
2   "time": "2019-01-30 17:28:08.985424",
3   "eyes": {
4     "left": {
5       "square": 992,
6       "pupil": {
7         "center": "(27;7)",
8         "radius": 9
9       },
10      "broves": {
11        "upper": {
12          "points": [
13            "(281;279)",
14            "(287.5;273.5)",
15            "(294;268)",
16            "(302.5;267.5)",
17            "(311;267)",
18            "(318.5;272.5)",
19            "(326;278)"
20          ],
21          "curve": {
22            "a": 0.02379309429442818,
23            "b": -14.459442002250645,
24            "c": 2463.431968789802
25          }
26        },
27        "lower": {
28          "points": [
29            "(326;278)",
30            "(318.5;280.5)",
31            "(311;283)",
32            "(302.5;283.5)",
33            "(294;284)",
34            "(287.5;281.5)",
35            "(281;279)"
36          ],
```

```

37     "curve":{
38         "a": -0.01066542312782873 ,
39         "b": 6.440400483567758 ,
40         "c": -688.4071518375195
41     }
42 }
43 }
44 },
45 "right":{
46     "square":1006 ,
47     "pupil":{
48         "center": "(30;7)",
49         "radius": 10
50     },
51     "broves":{
52         "upper":{
53             "points": [
54                 "(393;275)",
55                 "(400.5;268.5)",
56                 "(408;262)",
57                 "(416.5;261.5)",
58                 "(425;261)",
59                 "(431.0;266.0)",
60                 "(437;271)"
61             ],
62             "curve":{
63                 "a": 0.025707104081443288 ,
64                 "b": -21.44101005700644 ,
65                 "c": 4731.308325873293
66             }
67         },
68         "lower":{
69             "points": [
70                 "(437;271)",
71                 "(431.5;274.5)",
72                 "(426;278)",
73                 "(418.0;278.0)",
74                 "(410;278)",
75                 "(401.5;276.5)",
76                 "(393;275)"

```

```
77     ],
78     "curve":{
79         "a": -0.011252942018346649 ,
80         "b": 9.27990132880136 ,
81         "c": -1634.6251119726612
82     }
83 }
84 }
85 }
86 },
87 "temple":{
88     "left": "(232;292) " ,
89     "right": "(489;282) "
90 },
91 "face":{
92     "blCorner": "(201;468) " ,
93     "trCorner": "(230;498) "
94 },
95 "nose": [
96     "(360;267) " , "(362;294) " , "(363;319) " , "(364;344) "
97 ] ,
98 "mouth": [
99     "(318;415) " , "(336;407) " , "(352;401) " , "(367;404) " , "(381;400) " , "(398;402) " ,
    (415;406) " , "(401;418) " , "(385;424) " , "(370;426) " , "(355;426) " , "(336;424) " ,
    (324;414) " , "(353;409) " , "(367;410) " , "(382;407) " , "(408;407) " , "(383;410) " ,
    (368;412) " , "(354;412) "
100 ]
101 }
```