

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувачка кафедри,
д-р техн. наук, проф.

_____Ірина ЖУРАВСЬКА

«__» _____ 202__ р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
Моніторингова система для розпізнавання пожеж
у приміщеннях

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія»

Здобувач

_____Олександр ЮРЧЕНКО

підпис

«__» _____ 202__ р.

Керівник д-р техн. наук, професор,
завідувач каф. КІ

_____Ірина ЖУРАВСЬКА

підпис

«__» _____ 202__ р.

Факультет	Комп'ютерних наук
Кафедра	Комп'ютерної інженерії
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	123 Комп'ютерна інженерія
Освітня програма	Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерної інженерії
_____ Ірина ЖУРАВСЬКА
«_____» _____ 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

_____ Юрченка Олександра Валерійовича

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

_____ Моніторингова система для розпізнавання пожеж у приміщеннях

Затверджена наказом по ЧНУ ім. Петра Могили від 30.10.2024 № 297.

2. Строк представлення кваліфікаційної роботи «_____» _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом роботи є: апаратне та програмне забезпечення моніторингової системи для розпізнавання пожеж у приміщеннях. Вхідними даними роботи є специфікація вимог, що описує характеристики зазначеного апаратного та програмного забезпечення

4. Перелік питань, що підлягають розробці:

- 1) дослідження предметної області;
- 2) огляд існуючих систем розпізнавання пожеж;
- 3) розробка апаратної частини системи для розпізнавання пожеж у приміщеннях;
- 4) розробка програмної частини системи для розпізнавання пожеж у приміщеннях.

5. Перелік графічних матеріалів

Презентація

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи

Особистий підпис

Ірина ЖУРАВСЬКА

Власне ім'я ПРИЗВИЩЕ

Здобувач

Особистий підпис

Олександр ЮРЧЕНКО

Власне ім'я ПРИЗВИЩЕ

Дата видачі завдання « _____ » _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної бакалаврської роботи

Тема: Моніторингова система для розпізнавання пожеж у приміщеннях

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КБР	30.10.2024	30.10.2024	Виконано
2	Огляд літератури за темою роботи	01.11.2024	15.01.2025	Виконано
3	Складання календарного плану КБР	16.01.2025	01.02.2025	Виконано
4	Аналіз предметної області	02.02.2025	18.02.2025	Виконано
5	Розробка проєктних рішень	20.02.2025	20.03.2025	Виконано
6	Моделювання та конструювання АПЗ	30.03.2025	20.04.2025	Виконано
7	Перевірка працездатності, тестування та апробація розробленого АПЗ, аналіз результатів тестування	21.04.2025	30.05.2025	Виконано
8	Відгук керівника КБР	03.06.2025	06.06.2025	Виконано
9	Оформлення КБР та презентації	12.05.2025	04.06.2025	Виконано
10	Попередній захист	19.05.2025	07.06.2025	Виконано
11	Рецензування	08.06.2025	09.06.2025	Виконано
12	Завершення оформлення КБР та презентації	10.06.2025	12.06.2025	Виконано
13	Захист кваліфікаційної бакалаврської роботи	25.06.2025	25.06.2025	Виконано

Керівник роботи

Особистий підпис

Ірина ЖУРАВСЬКА

Власне ім'я ПРИЗВИЩЕ

Здобувач

Особистий підпис

Олександр ЮРЧЕНКО

Власне ім'я ПРИЗВИЩЕ

«____» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи

«Моніторингова система для розпізнавання пожеж у приміщеннях»

Студент 405 гр.: Юрченко Олександр Валерійович

Керівник: д-р техн. наук, проф. Журавська Ірина Миколаївна

Ця бакалаврська робота присвячена розробці апаратно-програмної системи для раннього виявлення пожеж у приміщеннях на основі комп'ютерного зору та сенсорного моніторингу повітря. Об'єктом дослідження є процес автоматичного моніторингу ознак займання у закритих приміщеннях. Предмет – апаратно-програмні рішення для детекції диму та полум'я. Метою є підвищення точності і швидкості виявлення пожеж шляхом використання методів відеооброблення та аналізу сенсорних даних. Практична значимість полягає у створенні автономної системи, придатної до інтеграції у реальні умови житлових і промислових об'єктів.

Основні результати роботи:

У першому розділі проведено огляд існуючих методів виявлення пожеж – від традиційних сенсорів до алгоритмів машинного зору. Виділено основні недоліки класичних рішень, зокрема їхню чутливість до помилкових спрацювань та недостатню адаптивність.

У другому розділі змодельовано функціональну структуру системи. Обґрунтовано вибір апаратної платформи (Raspberry Pi 4 Model B), IP-камери з підтримкою RTSP та сенсора MQ-2 для виявлення диму. Запропоновано алгоритм аналізу кадрів (на основі HSV-фільтрації, виявлення контурів, мерехтіння) та математичну модель інтеграції сенсорних і відео ознак у єдину ймовірнісну оцінку. Побудовано блок-схеми роботи системи та розроблено критерії точності, чутливості та F1-міри.

У третьому розділі реалізовано програмну частину системи на Python з використанням бібліотек OpenCV, NumPy, RPi.GPIO, spidev. Описано структуру коду: відеоаналіз, обробка сигналів із MQ-2, логіка прийняття рішень. Сформовано простий консольний інтерфейс та протестовано систему в реальному середовищі. Проведено тестування, що показали стабільну роботу детектора з часом реакції та точністю

Пояснювальна записка містить 66 сторінок (без додатків), 28 рисунків, 8 таблиць, 22 джерел та 2 додатки.

Ключові слова: *пожежна безпека, Raspberry Pi, відеоаналітика, OpenCV, MQ-2, виявлення полум'я, Python, система моніторингу.*

ABSTRACT

of the Bachelor's Thesis

"Monitoring System for Indoor Fire Detection"

Student: Oleksandr Yurchenko

Supervisor: Doctor of Technical Sciences, Professor Iryna Zhuravska

This bachelor's thesis is dedicated to the development of a hardware-software system for early fire detection in indoor environments using computer vision and air quality monitoring sensors. The object of research is the process of automatic monitoring of ignition signs in enclosed spaces. The subject is hardware and software solutions for detecting smoke and flames.

The aim is to improve the accuracy and speed of fire detection by combining video processing techniques with sensor data analysis. The practical significance lies in creating an autonomous system suitable for integration into real residential and industrial settings.

Key results of the work include:

The first chapter provides an overview of existing fire detection methods, ranging from traditional sensors to computer vision algorithms. The main drawbacks of conventional approaches are identified, particularly their sensitivity to false alarms and lack of adaptability.

The second chapter presents the functional structure of the system. The choice of hardware platform (Raspberry Pi 4 Model B), RTSP-compatible IP camera, and MQ-2 smoke sensor is justified. An algorithm for frame analysis (based on HSV filtering, contour detection, and flicker recognition) and a mathematical model for integrating sensor and video features into a single probabilistic metric are proposed. System workflows and performance criteria (accuracy, sensitivity, and F1-score) are developed.

The third chapter describes the software implementation in Python using libraries such as OpenCV, NumPy, RPi.GPIO, and spidev. The system's code structure includes video analysis, MQ-2 signal processing, and decision-making logic. A simple console interface was created, and the system was tested in a real environment. Testing demonstrated stable detector operation with timely response and accurate fire detection.

The thesis contains 66 pages (excluding appendices), 28 figures, 8 tables, 22 references, and 2 appendices.

Keywords: *fire safety, Raspberry Pi, video analytics, OpenCV, MQ-2, flame detection, Python, monitoring system.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ.....	6
1.1 Актуальність системи розпізнавання пожеж	6
1.2 Огляд сучасних систем розпізнавання пожеж	8
1.3 Вимоги до системи.....	13
Висновки до розділу 1	19
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ПОЖЕЖ У ПРИМІЩЕННЯХ.....	21
2.1 Функціональна схема.....	21
2.2 Вибір і обґрунтування апаратних засобів.....	23
2.3 Вимоги до середовища експлуатації системи	30
2.4 Розробка алгоритму виявлення пожежі.....	32
2.5 Математичне моделювання критеріїв виявлення пожежі	37
Висновки до розділу 2	40
3 РОЗРОБКА ТА ДОСЛІДЖЕННЯ АПАРАТНО-ПРОГРАМНОЇ СИСТЕМИ ВИЯВЛЕННЯ ПОЖЕЖ	41
3.1 Вибір технологій і засобів реалізації	41
3.2 Розробка прототипу апаратної частини.....	46
3.3 Реалізація програмної частини системи	50
3.4 Результати тестування	53
3.5 Оцінка якості та методи покращення системи.....	57
Висновки до розділу 3	61
ВИСНОВКИ.....	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	64
ДОДАТОК А Код програми розпізнавання пожеж у приміщеннях	67
ДОДАТОК Б Матеріали апробації роботи	70

ПЕРЕЛІК СКОРОЧЕНЬ

АПЗ	– апаратно-програмне забезпечення
АЦП	аналого-цифровий перетворювач
КБР	– кваліфікаційна бакалаврська робота
HSV	– Hue, Saturation, Value
PoE	– Power over Ethernet
ppm	– parts per million
SPI	– Serial Peripheral Interface

ВСТУП

У сучасному світі проблема пожежної безпеки є однією з найактуальніших, особливо в умовах урбанізованого середовища, де кількість технічних пристроїв, електрообладнання та легкозаймистих матеріалів постійно зростає. Щороку внаслідок пожеж гинуть тисячі людей, знищуються матеріальні цінності, виникають значні екологічні збитки. Особливо небезпечними є пожежі в закритих приміщеннях, де час реагування має критичне значення для порятунку життя та мінімізації збитків. У зв'язку з цим виникає необхідність удосконалення систем виявлення пожеж, що дає змогу оперативно реагувати на загрозу.

Актуальність теми полягає в необхідності впровадження сучасних технічних рішень для автоматичного моніторингу стану повітря в приміщеннях з метою раннього виявлення ознак загоряння. Традиційні системи пожежної сигналізації здебільшого базуються на спрацьовуванні датчиків диму, полум'я або температури, що часто не забезпечує достатню оперативність та точність виявлення. Новітні підходи, зокрема використання комп'ютерного зору, штучного інтелекту та цифрової обробки зображень, відкривають нові можливості для підвищення ефективності та надійності моніторингу.

Практичне значення теми полягає в інтеграції сучасних інформаційних технологій у сферу пожежної безпеки. Запропонована система моніторингу на основі обробки відео з камери спостереження дозволяє виявляти ознаки пожежі на ранній стадії – шляхом аналізу появи диму або полум'я в кадрі. Такий підхід не лише автоматизує процес нагляду, а й зменшує людський фактор, підвищуючи точність і швидкість реагування.

Метою даної роботи є підвищення ефективності виявлення пожеж у приміщеннях шляхом розробки та впровадження моніторингової системи, що здійснює автоматичне розпізнавання ознак пожежі на основі аналізу відеоінформації.

Для досягнення визначеної мети необхідно вирішити такі завдання:

- обґрунтувати актуальність і сучасний стан проблеми виявлення пожеж;

- здійснити аналіз науково-технічної літератури щодо методів розпізнавання вогню та диму;
- обрати апаратну платформу та засоби розробки програмного забезпечення;
- розробити програмно-апаратну систему моніторингу на основі відеоаналітики;
- провести тестування системи в умовах, наближених до реальних;
- оцінити ефективність запропонованого рішення та визначити напрямки подальшого вдосконалення.

Об'єктом дослідження є процес моніторингу стану середовища в приміщенні з метою виявлення пожежної небезпеки.

Предметом дослідження є апаратно-програмні засоби та алгоритми розпізнавання ознак пожежі (полум'я, диму) на основі відеоданих.

Практична цінність розробки полягає у створенні прототипу системи, яку можна інтегрувати в існуючі комплекси безпеки. Таке рішення здатне суттєво підвищити надійність протипожежного захисту у житлових, офісних, промислових і громадських приміщеннях.

Таким чином, обрана тема є актуальною, має значний науково-практичний потенціал і відповідає сучасним вимогам у сфері техногенної безпеки.

Апробація роботи відбулася під час XXVII Всеукраїнської науково-практичної конференції «Могилянські читання – 2024: Досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти» (06–10 листопада 2024 р., Миколаїв). За результатами опубліковано тези доповідей [11].

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ

1.1 Актуальність системи розпізнавання пожеж

Протягом 2024 року в Україні за 11 місяців було зареєстровано 101'437 випадків пожеж у будівлях, що на 56,7 % перевищує показник за аналогічний період 2023 року, причому близько 67 % інцидентів сталися в житловому секторі, включно з квартирами та приватними будинками.

За цей же період підрозділи ДСНС ліквідували 3'513 таких пожеж, унаслідок яких загинуло 215 осіб і постраждало 144 особи; більшість жертв та постраждалих мешкали саме у приватному житлі. Матеріальні втрати лише від внутрішніх пожеж за перші вісім місяців 2024 року сягнули понад 139 млн грн щодня, що вказує на мільярдні збитки за рік [1]. Розподіл питомої ваги пожеж за причинами їх виникнення в Україні за 8 місяців 2024 року наведено на рис. 1.1.

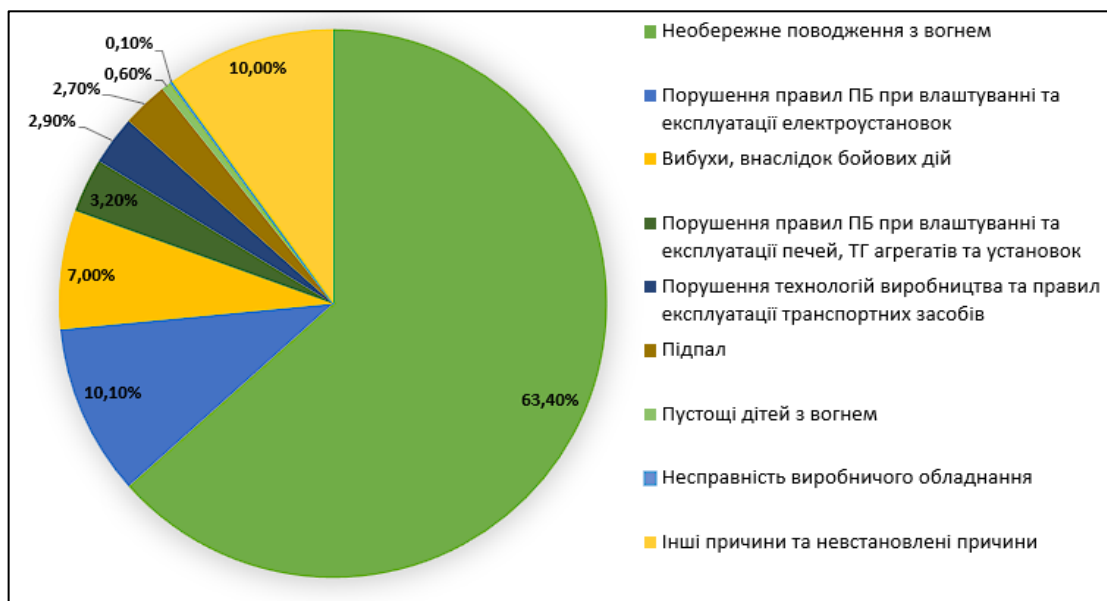


Рисунок 1.1 – Розподіл питомої ваги пожеж за причинами їх виникнення в Україні за 8 місяців 2024 року

У світовому масштабі у 2023 році в США сталося приблизно 352'000 пожеж у житлових будівлях, які призвели до понад 3'070 загиблих та 11'790 поранених, а прямі збитки від цих інцидентів перевищили \$11,4 млрд доларів.

Кожні 95 секунд у Сполучених Штатах реєструється нова пожежа в будівлі, що підкреслює високу частоту й поширеність цього ризику для населення.

Щорічно понад 70 % усіх великих пожеж у світі відбуваються саме в житлових приміщеннях, що підтверджує дані Центру світової статистики пожеж на основі офіційних звітів різних країн [2].

У Європейському Союзі 2023 року зареєстровано понад 120'000 пожеж у житлових будівлях, що спричинило понад 1'800 загиблих та збитки понад 7 млрд євро, а у країнах Східної Європи – понад 30'000 інцидентів із втратами людських життів та значними економічними збитками.

Першим і найважливішим етапом у запобіганні пожежам є проведення комплексної оцінки ризиків з метою виявлення потенційних загроз і факторів, що можуть спричинити виникнення надзвичайної ситуації. Це обов'язкова вимога згідно з нормами HSE (англ. Health, Safety and Environment) перед тим, як дозволити будь-якій особі працювати або перебувати в приміщенні.

До типових джерел небезпеки належать:

- несправні електроприлади, зношена проводка або пошкоджені кабелі;
- перевантажені, несправні або неправильно встановлені електричні розетки;
- побутові пристрої для приготування їжі (електроплити, мікрохвильові печі тощо);
- теплогенерувальне обладнання – обігрівачі, каміни, калорифери;
- легкозаймисті рідини та матеріали, розміщені біля джерел тепла;
- використання відкритого вогню, сигарет або запальничок у недозволених місцях.

Важливу роль у виникненні пожеж відіграє людський фактор. Серед поширених причин – залишені без нагляду електроприлади, порушення правил експлуатації обігрівальної техніки, а також використання несправної електромережі. Особливо високий рівень ризику спостерігається у сільській

місцевості та старому житловому фонді, де часто відсутні сучасні засоби моніторингу та протипожежного захисту.

У зв'язку з цим зростає актуальність впровадження доступних, надійних і високоефективних технологій раннього виявлення полум'я, здатних оперативно реагувати на загрозу й запобігати масштабним наслідкам.

Традиційні димові та теплові сповіщувачі часто спрацьовують із запізненням, особливо в умовах великих приміщень або при недостатній вентиляції. Натомість інтелектуальні системи розпізнавання полум'я, зокрема на базі комп'ютерного зору, дають змогу фіксувати загоряння на ранніх етапах розвитку – ще до появи видимого диму, що значно підвищує шанси на успішне гасіння та мінімізацію збитків.

Висока частота пожеж у житлових будівлях в Україні та світі, значні людські жертви й економічні втрати, а також обмеження традиційних систем виявлення підкреслюють актуальність створення й впровадження спеціалізованих відеомоніторингових рішень для своєчасного розпізнавання полум'я всередині приміщень.

Такі системи не лише підвищують швидкість реагування й точність детекції, а й зменшують кількість хибних тривог, сприяючи підвищенню рівня безпеки мешканців і збереженню матеріальних цінностей.

1.2 Огляд сучасних систем розпізнавання пожеж

Нині в умовах зростання кількості пожеж у житлових та комерційних приміщеннях традиційні димові й теплові сповіщувачі залишаються найпоширенішими рішеннями через простоту встановлення та невисоку вартість, водночас вони мають суттєві обмеження. Перевагою таких сенсорів є здатність фіксувати підвищену концентрацію диму або раптовий стрибок температури й автоматично запускати сигнал тривоги без постійного моніторингу, проте середній час виявлення перевищує 15–20 хвилин, особливо коли джерело вогню віддалене від датчика, що ускладнює ранню евакуацію та локалізацію пожежі. До того ж часті хибні спрацювання через кухонні випари,

пил чи зміну вологості знижують довіру мешканців до системи й нерідко призводять до їх відключення.

Найпоширеніша модель через свою доступність та простоту монтажу – дротовий пожежний сповіщувач Arton СПД-3.0, який наведений на рис. 1.2.



Рисунок 1.2 – Проводовий пожежний сповіщувач Arton СПД-3.0 [14]

Оптичні UV/IR-флейм-детектори здатні швидко виявляти відкритий вогонь за його спектральними характеристиками, реагуючи за лічені секунди навіть без наявності диму, що робить їх ефективними в зоні підвищеного ризику, наприклад у котельнях або машинних відділеннях. Прикладом таких датчиків виступає сповіщувач FlameSpec-UV-IR (рис. 1.3).



Рисунок 1.3 – Сповіщувач FlameSpec-UV-IR [13]

Мультиспектральні моделі (UV+Dual-IR, Triple-IR) можуть значно знижувати кількість хибних тривог порівняно з простішими сенсорами, але ці

2025 р.

рішення коштують удвічі–утричі дорожче, потребують регулярного калібрування та чутливі до змін освітлення на об'єкті. До таких належить Dual IR Flame Detector 55000-064APO, представлений на рис. 1.4.



Рисунок 1.4 – Мультиспектральний детектор Dual IR Flame Detector 55000-064APO [12]

Відеоаналітичні системи на базі комп'ютерного зору і глибинного навчання здатні миттєво розпізнавати полум'я та дим у режимі реального часу, аналізуючи темпорозміри, колірні спектри й текстури кадрів. Згідно з дослідженнями, точність таких алгоритмів перевищує 96 %, а затримка обробки не перевищує 0,5 с на сучасному обладнанні, що істотно скорочує час реакції та мінімізує масштаби пожежі [3].

Перевагою таких систем є можливість інтеграції з існуючими CCTV-мережами, що дозволяє ефективно використовувати вже наявну інфраструктуру для моніторингу. Крім того, вони підтримують архівування відео з автоматичною генерацією звітів, що спрощує аналіз подій, підвищує оперативність реагування та сприяє створенню доказової бази. Водночас впровадження таких систем потребує значних обчислювальних ресурсів для обробки відеопотоків у реальному часі, що може вимагати використання потужних серверів або хмарних рішень. Також вони характеризуються вищою вартістю встановлення та обслуговування порівняно з простішими аналогами, і

вимагають організації захищеного каналу передачі відео, аби уникнути несанкціонованого доступу та забезпечити конфіденційність даних.

Розподілені IoT-мережі сенсорів (рис. 1.5) поєднують переваги кількох джерел даних – температурних, димових, газових та відеодетекторів, і передають інформацію на хмарні платформи для глибокої аналітики та адаптивного калібрування порогів спрацювання [4].



Рисунок 1.5 – Техніки виявлення пожежі

Такі рішення забезпечують гнучкість у масштабуванні великих об'єктів і можуть автоматично видавати комплексні звіти, але залежність від бездротового зв'язку підвищує ризик втрати сигналу в критичний момент, а експлуатаційні витрати на обслуговування численних вузлів можуть перевищувати бюджет невеликих будівель.

Інтеграція перелічених технологій у складі гібридних платформ дозволяє поєднати високу швидкодію UV/IR-детекторів, аналітичну потужність систем відеомоніторингу на основі штучного інтелекту та масштабованість IoT-інфраструктур. Завдяки цьому досягається максимальна ефективність виявлення загроз: такі комплексні системи вирізняються високою точністю, стабільною роботою в реальному часі та надзвичайно низьким рівнем хибних спрацювань – у межах 2–3 %.

Проте впровадження подібних рішень супроводжується низкою викликів: складна архітектура систем вимагає ретельно спланованої реалізації, значних початкових інвестицій, а також кваліфікованого технічного обслуговування на постійній основі.

Порівняльний аналіз різних типів сенсорів і технологій виявлення наведено в табл. 1.1.

Таблиця 1.1 – Порівняння основних типів детекторів

Тип системи	Час виявлення	Точність, %	Хибні спрацювання, %	Вартість	Плюси та мінуси
Димові/теплові сповіщувачі	15–20 хв	70–85	10–25	Низька	Плюси: дешева, проста інсталяція; Мінуси: запізнення, хибні тривоги
UV/IR-флейм-детектори	1–5 с	85–95	5–10	Середня–висока	Плюси: швидке виявлення без диму; Мінуси: висока ціна, чутливі до освітлення
Відеоаналітичні AI-системи	< 0.5 с	> 96	2–5	Висока	Плюси: інтеграція з CCTV, архівація; Мінуси: великі обчислювальні ресурси
ІоТ-мережі сенсорів	5–30 хв	75–90	8–15	Залежить від масштабу	Плюси: масштабованість, адаптивність; Мінуси: залежність від зв'язку
Гібридні платформи	< 5 с	> 98	1–3	Дуже висока	Плюси: оптимальне поєднання; Мінуси: складна архітектура, висока вартість

Сучасні системи виявлення пожеж дозволяють ефективно реагувати на загоряння на різних етапах його розвитку. Вони забезпечують як базовий рівень захисту, так і високоточне раннє виявлення з мінімальною кількістю хибних тривог, що дає змогу своєчасно вжити заходів для мінімізації шкоди та забезпечення безпеки людей і майна.

1.3 Вимоги до системи

Моніторингова система розпізнавання пожеж у приміщеннях призначена для цілодобового автоматичного виявлення ознак займання всередині будівель із мінімальним часом реагування. Основною метою є забезпечення максимально швидкої реакції – не більше ніж за 5 секунд – при високому рівні точності, з хибними спрацюваннями, що не перевищують 5 %. Такий підхід дозволяє своєчасно виявити загрозу займання, передати сигнал тривоги на пристрої оповіщення або інтегрувати систему з автоматизованими комплексами пожежогасіння для швидкої ліквідації джерела займання.

Система реалізована за принципом комбінованого підходу, де поєднується аналіз відео з цифрових камер та алгоритми комп'ютерного зору. Основним апаратним елементом є одноплатний комп'ютер Raspberry Pi 4 (рис. 1.6), який завдяки своїй енергоефективності, компактним розмірам та достатній обчислювальній потужності забезпечує локальну обробку відеопотоку [6]. IP-камера, підключена до одноплатного комп'ютера Raspberry Pi, передає відеопотік у режимі реального часу, після чого окремі кадри виділяються й аналізуються за допомогою бібліотеки OpenCV [7]. У процесі обробки додатково застосовуються попередньо навчені моделі глибокого навчання, які дозволяють ефективно виявляти характерні ознаки полум'я – такі як динамічні контури, кольорні характеристики, мерехтіння, а також специфічні рухові патерни.



Рисунок 1.6 – Одноплатний комп'ютер Raspberry Pi 4

Алгоритм передбачає попередню обробку кадрів, виявлення динамічних контурів, фільтрацію за кольірними характеристиками (відтінки червоного, жовтого, помаранчевого), а також аналіз нестабільності структури полум'я (його мерехтіння та випадкових коливань).

Для підвищення точності розпізнавання застосовуються попередньо навчені моделі у форматі XML на основі каскадів Гаара [5]. Ці моделі дозволяють виявляти характерні просторово-часові ознаки полум'я у відео та зіставляти їх із заздалегідь заданими шаблонами. У результаті система здатна реагувати не лише на зміну кольору чи яскравості, а й на поведінкові особливості полум'я, що дає змогу зменшити кількість хибних тривог від джерел світла (наприклад, ламп, телевізора або сонячних відблисків).

Додатково до відеоаналізу, система підтримує інтеграцію з класичними сенсорними засобами виявлення пожежі, такими як аналогові або цифрові датчики диму. Серед них особливо популярними є сенсори типу MQ-2 (рис. 1.7), які реагують на наявність диму, пропану, метану, водню та інших горючих газів, а також фотометричні сенсори, що оцінюють концентрацію диму за оптичними характеристиками повітря [9]. Дані з цих сенсорів використовуються для підтвердження візуального виявлення полум'я або можуть самостійно ініціювати сигнал тривоги у разі фіксації аномальних параметрів повітря.



Рисунок 1.7 – Датчик газу MQ-2

Система розпізнавання полум'я призначена для використання у внутрішніх приміщеннях житлового та комерційного призначення, зокрема в

квартирах, приватних будинках, офісах, навчальних закладах, дитячих садках, бібліотеках, лікарнях, лабораторіях та інших установах, де необхідне постійне відеоспостереження з автоматичним виявленням ознак займання.

Особлива ефективність системи проявляється в умовах, коли немає постійного нагляду з боку персоналу – наприклад, у нічний час, під час вихідних або свят, а також у закритих просторах з обмеженим доступом. Завдяки використанню комп'ютерного зору й алгоритмів глибокого навчання, система здатна виявляти навіть незначні прояви диму чи полум'я ще до того, як вони спричинять масштабну пожежу.

Крім цього, її можна інтегрувати з існуючими системами відеонагляду, пожежогасіння, оповіщення й управління евакуацією, що робить її універсальним інструментом для побудови багаторівневої системи безпеки. Це особливо актуально для об'єктів з великою кількістю людей або цінним обладнанням, де затримка з реагуванням на займання може мати критичні наслідки.

Основні вимоги до апаратних компонентів, які необхідні для реалізації функціоналу мобільної системи розпізнавання пожежі у приміщенні:

а) Raspberry Pi 4 Model B (4 GB RAM):

- процесор: Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz;
- оперативна пам'ять: 4 GB LPDDR4-3200 SDRAM;
- інтерфейси: 2 × micro-HDMI, 2 × USB 3.0, 2 × USB 2.0, Gigabit Ethernet, GPIO (40 пінів);
- призначення: керування системою, обробка відео, запуск скриптів виявлення полум'я, взаємодія з датчиками;

б) карта пам'яті microSD (рис. 1.8) (UHS-I, 32 Гбайт або більше):

- тип: UHS-I, клас швидкості 10 або вищий;
- мінімальна швидкість читання/запису: 80–100 Мбіт/с;
- призначення: завантаження ОС (Raspberry Pi OS), зберігання відеоархівів, журналів подій, моделей та коду;

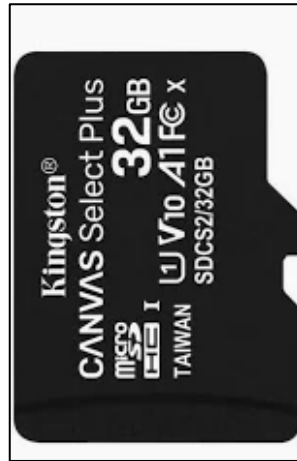


Рисунок 1.8 – Карта пам'яті microSD 32 Гбайт

в) IP-камера:

- роздільність: мінімум 1920×1080 при 30 кадрах/с;
- підключення: Ethernet або Wi-Fi;
- підтримка протоколів: RTSP, HTTP, ONVIF;
- опції: інфрачервоне підсвічування для нічного бачення, широкий кут огляду ($\geq 90^\circ$);
- призначення: передача відео до Raspberry Pi для аналізу в реальному часі;

г) датчик диму MQ-2:

- виявляє: дим, LPG, бутан, метан, водень, спирт;
- напруга живлення: 5 V;
- сигнал на виході: аналоговий (для точного контролю) або цифровий (через компаратор);
- діапазон виявлення: 300–10 000 ppm;
- призначення: додатковий контроль за наявністю диму/газу в приміщенні;

д) ADC-модуль (аналого-цифровий перетворювач)

- тип: MCP3008 (рис. 1.9) (10-бітний, 8 каналів) [8];
- інтерфейс: SPI (Serial Peripheral Interface), підключається до GPIO Raspberry Pi;

- живлення: 3.3 V або 5 V;
- призначення: перетворює аналоговий сигнал від MQ-2 у цифровий для коректного зчитування Raspberry Pi, оскільки вбудованого ADC на платі немає;



Рисунок 1.9 – Аналого-цифровий перетворювач MCP3008

е) блок живлення 5 V, 3 A (рис. 1.10)

- забезпечує стабільне електроживлення для Raspberry Pi та датчиків;
- може бути реалізований як адаптер з розеткою або як модуль живлення на основі акумулятора;
- важливо передбачити захист від перенапруги й стабілізацію.



Рисунок 1.10 – Блок живлення 5 V

Основні програмні вимоги, які необхідні для реалізації функціоналу мобільної системи розпізнавання пожежі у приміщенні:

а) операційна система:

- назва: Raspberry Pi OS;
- сумісність: оптимізована для ARM-архітектури Raspberry Pi;
- призначення: базове середовище для запуску скриптів, обслуговування пристроїв, мережевої взаємодії;

б) мова програмування:

- тип: Python 3.9;
- особливості: підтримує швидко розробку, великий набір бібліотек для роботи з відео, сигналами, мережею;
- призначення: написання головного скрипту системи, обробка відео, робота з датчиками, логіка сповіщення;

в) бібліотека комп'ютерного зору OpenCV:

- версія: 4.5 або вище;
- функції: зчитування відео з камери, обробка кадрів, виявлення ознак полум'я або диму;
- інтеграція: підтримує моделі у форматі XML, дозволяє виконання детекції у реальному часі;
- призначення: реалізація основного алгоритму розпізнавання пожежонебезпечних ситуацій.

г) XML-модель виявлення (каскади Гаара):

- формат: .xml;
- джерело: натренована модель за допомогою OpenCV або інших інструментів машинного навчання;
- призначення: виявлення полум'я на основі візуальних ознак у відео потоці;
- завантаження: через cv2.CascadeClassifier() у Python-скрипті;

д) системні бібліотеки та середовище:

- python-бібліотеки: opencv-python, numpy, datetime, os, time, csv, RPi.GPIO, spidev, requests;

- призначення: підтримка повноцінного циклу виявлення, збереження, сповіщення та взаємодії з апаратною частиною;

е) мережеве забезпечення:

- протоколи: RTSP/HTTP – для прийому відео з IP-камер;
- підключення: Ethernet або Wi-Fi;
- призначення: забезпечення потокової передачі відео, взаємодія зі службами оповіщення;

ж) зберігання та логування:

- формати: CSV, знімки у форматі JPEG/PNG;
- призначення: фіксація дати та часу, зберігання кадрів з камер у момент виявлення.

Система розпізнавання пожеж повинна відповідати європейському стандарту ІЕС 61508, застосування якого гарантує функціональну безпеку, правильне проєктування й експлуатацію сигналізаційних систем [10].

Міжнародний стандарт ІЕС 61508 визначає вимоги до функціональної безпеки електричних, електронних та програмованих електронних систем безпеки (Е/Е/РЕ), охоплюючи всі фази їхнього життєвого циклу – від початкового аналізу ризиків до технічного обслуговування і верифікації. Він ґрунтується на ризик-орієнтованому підході, встановлюючи кількісні критерії зниження ймовірності небезпечних відмов через призначення рівнів безпеки SIL (Safety Integrity Levels) від 1 до 4 та передбачає формалізовані процедури розробки, тестування й супроводу ПЗ і апаратури, щоб гарантувати передбачувану поведінку в разі відмови.

Висновки до розділу 1

У даному розділі було розглянуто та обґрунтовано актуальність і сучасний стан проблеми виявлення пожеж. Проаналізовано існуючі методи виявлення пожеж, зокрема сенсорні та відеоаналітичні підходи, визначено їхні переваги й недоліки. Було розглянуто три основні класи систем: класичні сенсорні рішення на основі диму й температури, оптичні флейм-детектори, а також сучасні

відеоаналітичні системи, які використовують комп'ютерний зір і машинне навчання.

Зокрема, було показано, що традиційні димові та теплові сповіщувачі, хоча й є найдоступнішими, мають значні обмеження щодо швидкості реакції та площі покриття. Їх доцільно використовувати в невеликих приміщеннях або як резервні системи у багаторівневій структурі безпеки.

Оптичні UV/IR-флейм-детектори виявилися ефективними для швидкого реагування на відкритий вогонь навіть за відсутності диму. Однак їх висока вартість і технічна складність обмежують можливості використання в побутових або бюджетних комерційних умовах.

Найбільш перспективними виявились системи на основі відеоаналітики, здатні здійснювати детекцію полум'я за візуальними патернами в реальному часі. Ці рішення забезпечують високу точність і масштабованість, особливо у випадках інтеграції з уже існуючими системами відеоспостереження.

Окрему увагу було приділено розподіленим IoT-системам, які поєднують кілька типів сенсорів і використовують хмарну або локальну обробку даних. Такі платформи дозволяють реалізувати масштабовані рішення для великих об'єктів, проте потребують ретельного проєктування мережі передачі даних та регулярного технічного обслуговування.

Визначено функціональні, апаратні та програмні вимоги до моніторингової системи розпізнавання пожеж, призначеної для цілодобового автоматичного виявлення займання в приміщеннях. Основна мета системи – забезпечення високої точності виявлення із мінімальним часом реакції (до 5 секунд) за допомогою комбінованого підходу: відеоаналізу ознак полум'я (через HSV-фільтрацію, каскади Гаара) та контролю параметрів повітря за допомогою сенсора MQ-2. Система також відповідає вимогам стандарту ІЕС 61508, що гарантує її функціональну безпеку та відповідність міжнародним нормам.

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ СИСТЕМИ РОЗПІЗНАВАННЯ ПОЖЕЖ У ПРИМІЩЕННЯХ

2.1 Функціональна схема

Система виявлення пожеж реалізована як послідовний конвеєр обробки даних: спочатку IP-камера передає відеопотік у режимі реального часу через протокол RTSP, після чого Raspberry Pi 4 зчитує кадри для подальшого аналізу. Кожен кадр обробляється за допомогою бібліотеки OpenCV із завантаженою XML-моделлю Нааг-каскаду, що дозволяє виявляти характерні контури та колірні ознаки полум'я [15]. Паралельно до відеоаналізу відбувається опитування датчика MQ-2 через аналого-цифровий перетворювач MCP3008, який перетворює аналоговий сигнал диму і горючих газів у цифровий формат для Raspberry Pi. Функціональну схему основних модулів системи наведено на рис. 2.1.

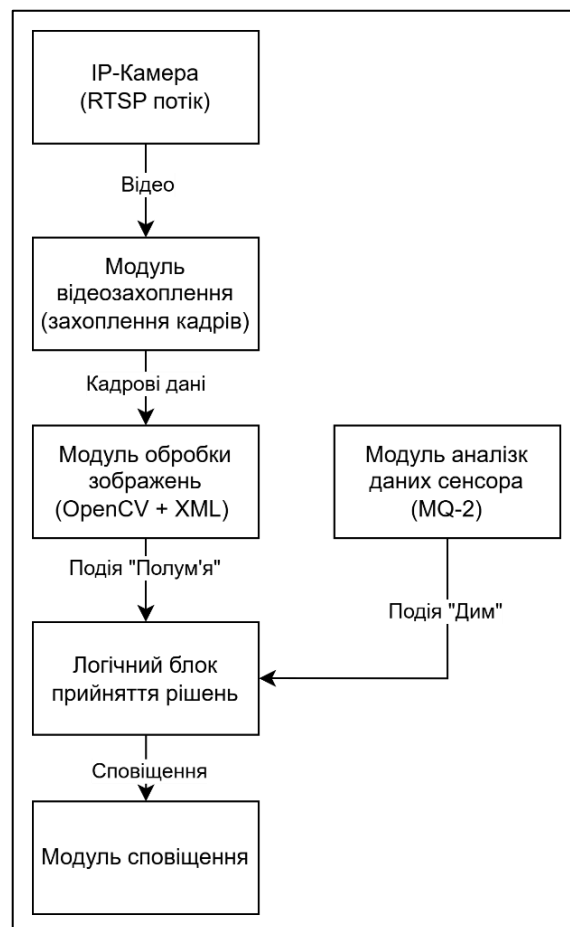


Рисунок 2.1 – Функціональна схема основних модулів системи

Модуль відеозахоплення виконує критично важливу функцію безперервного отримання та передачі відеоданих із камери. Його головна мета – підтримувати стабільний зв'язок із IP-камерою, надійно обробляти вхідний потік та проводити буферизацію кадрів, щоб уникнути затримок чи втрати даних. Буферизовані кадри потрапляють у спеціальну чергу, звідки передаються до модуля обробки зображення. Такий підхід гарантує узгоджену взаємодію між потоковими процесами та дозволяє синхронізувати роботу відеоаналітики з іншими частинами системи.

Модуль обробки зображення аналізує кожен відеокادر на предмет наявності вогню. Кадр конвертується у колірну модель HSV для ефективнішої роботи з відтінками, характерними для полум'я (жовтий, червоний, помаранчевий). Далі застосовуються фільтри кольору, методи виділення контурів, а також алгоритми згладжування, які допомагають зменшити шум і уникнути хибних спрацювань. На завершальному етапі виконується перевірка кадру за допомогою Haar-каскаду, який виявляє специфічні форми та текстури вогню. Якщо ознаки полум'я підтверджено, система генерує подію з типом «полум'я», часом виявлення та оцінкою ймовірності достовірності детекції.

Паралельно з відеоаналізом працює модуль збору сенсорних даних, який постійно зчитує рівень диму та газів у повітрі. Для цього використовується сенсор MQ-2, підключений до Raspberry Pi через аналого-цифровий перетворювач MCP3008. Аналогові сигнали від сенсора проходять обробку методом ковзного середнього, що дозволяє зменшити вплив миттєвих коливань або електричних шумів. Якщо значення перевищує заданий поріг (наприклад, 300 ppm), система фіксує подію типу «дим» та додає часову мітку для подальшого аналізу.

Всі події, згенеровані окремими модулями, надходять до логічного блоку прийняття рішень. Цей блок здійснює кореляцію між подіями з різних джерел. Основне правило – якщо протягом короткого інтервалу часу система фіксує одночасно ознаки полум'я та перевищення рівня диму, генерується сигнал тривоги. Такий підхід дозволяє зменшити кількість хибних спрацювань. Крім

простої логіки AND, система може бути налаштована на використання складніших правил з ваговими коефіцієнтами або комбінаційними умовами (AND/OR), що підвищує її гнучкість і стійкість у реальних умовах експлуатації.

На завершальному етапі сигнал тривоги передається до модуля сповіщення, який реалізує функцію оперативного інформування.

2.2 Вибір і обґрунтування апаратних засобів

Щоб успішно реалізувати апаратну частину проєкту, необхідно детально вивчити технічні характеристики та інтерфейси кожного обраного компонента – від одноплатного комп'ютера й мікроконтролерів до камер та сенсорів – і змодельовати їхню спільну роботу, враховуючи живлення, протоколи обміну даними та фізичне розташування. Лише чітке розуміння сумісності входів-виходів, швидкодії обробки сигналів і вимог до живлення забезпечить коректну інтеграцію всіх модулів у єдиний, надійний і ефективний апаратно-програмний комплекс.

2.2.1 Raspberry Pi 4 Model B

Raspberry Pi 4 Model B є головним обчислювальним центром моніторингової системи. Його основне призначення – обробка відео з камери в реальному часі (зокрема, з використанням бібліотеки OpenCV), прийом і аналіз сигналів із сенсорів, а також керування передачею даних або сповіщеннями

Вбудований GPIO-роз'єм дозволяє напряму підключати сенсори, світлодіоди, реле та інші виконавчі пристрої без потреби в окремому мікроконтролері, технічні характеристики наведено в табл. 2.1.

Переваги:

- сумісність із Python, OpenCV, NumPy та іншими бібліотеками;
- можливість підключення аналогових сенсорів через АЦП або SPI;
- підтримка Raspbian/Linux, що дозволяє розгортати скрипти автозапуску, вебсервери тощо;
- доступна документація та велика спільнота користувачів.

Таблиця 2.1 – Технічні характеристики Raspberry Pi

Характеристика	Значення
Процесор	Broadcom BCM2711, 4 × Cortex-A72 @ 1.5/1.8 GHz
Оперативна пам'ять	4 GB LPDDR4
GPIO	40-контактний (I ² C, SPI, UART, PWM, GPIO)
Мережа	Gigabit Ethernet; dual-band Wi-Fi; Bluetooth 5.0
Живлення	5 V DC через USB-C ≥ 3 A, PoE (англ. Power over Ethernet)
Споживання (idle/load)	≈ 2.7 W / ≈ 5.1 W
Роз'єм для дисплеїв	2-lane MIPI DSI

Основною перевагою Raspberry Pi є поєднання достатньої обчислювальної потужності, енергоефективності, доступної ціни та розширюваності. Модель 4 B має підтримку двох моніторів, мережі Gigabit Ethernet і бездротових протоколів Wi-Fi та Bluetooth, що дозволяє працювати як автономно, так і в мережі.

2.2.2 IP-камера відеоспостереження

У системі виявлення пожежі ключову роль відіграє IP-камера Vivotek IT9388-NT (рис. 2.2), яка забезпечує високоякісне відеоспостереження та ефективне виявлення диму або полум'я.



Рисунок 2.2 – IP-камера Vivotek IT9388-NT

Ця камера оснащена 5-мегапіксельним CMOS-сенсором розміром 1/2.7", що дозволяє отримувати зображення з максимальною роздільною здатністю 2560×1920 пікселів при 20 кадрах на секунду. Завдяки широкому динамічному діапазону (WDR Pro) у 120 дБ, камера ефективно працює в умовах складного освітлення, забезпечуючи чітке зображення як у яскравих, так і в темних ділянках сцени.

Вбудовані інфрачервоні підсвітлювачі з технологією Smart IR забезпечують нічне бачення на відстані до 30 метрів, що дозволяє здійснювати цілодобове відеоспостереження. Камера має моторизований варіофокальний об'єктив з фокусною відстанню від 2,8 до 12 мм, що забезпечує горизонтальний кут огляду від 90° до 30° , дозволяючи адаптувати зображення до конкретних умов спостереження.

Завдяки підтримці протоколів RTSP та ONVIF, камера легко інтегрується з різними системами відеоаналітики, такими як OpenCV. Підтримка кодеків H.265, H.264 та MJPEG дозволяє оптимізувати використання мережевих ресурсів та зберігання даних.

Переваги Vivotek IT9388-НТ:

- висока роздільна здатність та якість зображення для точного виявлення ознак пожежі;
- цілодобове відеоспостереження завдяки інфрачервоному підсвічуванню;
- гнучке налаштування кута огляду завдяки моторизованому об'єктиву;
- підтримка сучасних протоколів та кодеків для ефективної інтеграції та зберігання даних;
- надійний захист від зовнішніх впливів та вандалізму.

Технічні характеристики камери наведено в табл. 2.2.

Таблиця 2.2 – Технічні характеристики IP-камери

Характеристика	Значення
Роздільна здатність, пікс.	2560×1920 (5MP) при 20 кадрах/с
Кодеки	H.265, H.264, MJPEG

Протоколи	RTSP, ONVIF, HTTP, HTTPS, FTP, SMTP, SNMP та ін.
Інфрачервоне підсвічування	до 30 м (2 ІЧ-світлодіоди з Smart IR)
Захист корпусу	IP66, IK10 (антивандальний)
Живлення	PoE (802.3af)
Нічне бачення	0 люкс з увімкненим ІЧ-підсвічуванням
Кут огляду	90–30° (горизонтальний), 65–23° (вертикальний)
Об'єктив	Варіофокальний, 2,8–12 мм, моторизований

Камера має вбудований мікрофон для одностороннього аудіо та слот для карт пам'яті microSD/SDHC/SDXC з підтримкою до 1 Тбайт, що забезпечує локальне зберігання відео. Захищений корпус з IP66 та IK10 гарантує надійну роботу в суворих погодних умовах та захист від вандалізму.

2.2.3 Газовий сенсор MQ-2

MQ-2 призначений для виявлення диму, пари бензину, пропану, метану, водню та інших горючих газів у повітрі. Він має високу чутливість, особливо до диму, що робить його ефективним індикатором на ранніх стадіях займання, коли візуального полум'я ще не видно, технічні характеристики наведено в табл. 2.3.

Переваги:

- дешевий і поширений сенсор із перевіреною надійністю;
- реакція вже на низьку концентрацію диму (від 300 ppm);
- працює в широкому температурному діапазоні;
- легко інтегрується в проекти з Raspberry Pi через MCP3008.

Таблиця 2.3 – Технічні характеристики газового сенсору MQ-2

Характеристика	Значення
Діапазон вимірювань	300–10000 ppm (дим, LPG, пропан, водень)
Напруга нагрівника	5 V DC
Поріг спрацювання	≈ 300 ppm
Чутливість	R_0/R_s при 1000 ppm $H_2 \approx 7$

Час відгуку	≤ 10 с
Тривалість прогріву	≈ 24 години (рекомендовано)
Робоча температура	200–300 °C

Сенсор реагує на зміну концентрації газу через зміну опору напівпровідникового елемента. Цей аналоговий сигнал може бути легко оцифрований за допомогою АЦП.

2.2.4 Аналого-цифровий перетворювач MCP3008

Оскільки Raspberry Pi не має вбудованих аналогових входів, MCP3008 дозволяє перетворювати аналоговий сигнал від сенсора MQ-2 у цифровий. Це необхідно для точного аналізу змін концентрації газів. технічні характеристики наведено в табл. 2.4.

Переваги:

- дає змогу працювати з аналоговими сенсорами на цифровій платформі;
- 8 каналів – можливість підключити до 8 аналогових пристроїв одночасно;
- просте підключення до SPI Raspberry Pi;
- низьке енергоспоживання та компактні розміри.

Таблиця 2.4 – Технічні характеристики аналого-цифрового перетворювач MCP3008

Характеристика	Значення
Розрядність	10 bit
Канали	8 (один/диференціальний)
Інтерфейс	SPI (до 3.6 MHz)
Швидкість перетворення	до 200 ksp/s
Споживання струму	≤ 500 μ A
Напруга живлення	2.7–5.5 V
Температурний діапазон	від мінус 40 °C до 125 °C

MCP3008 використовує SPI-інтерфейс, який підтримується Raspberry Pi з бібліотеки `spidev`, що забезпечує швидку й стабільну передачу даних

2.2.5 Порівняння з аналогами

На ринку присутні інші мікрокомп'ютери, такі як Arduino Uno, ESP32, Jetson Nano та Orange Pi (рис. 2.3), які можуть використовуватись у системах моніторингу. Проте кожен з них має свої обмеження.

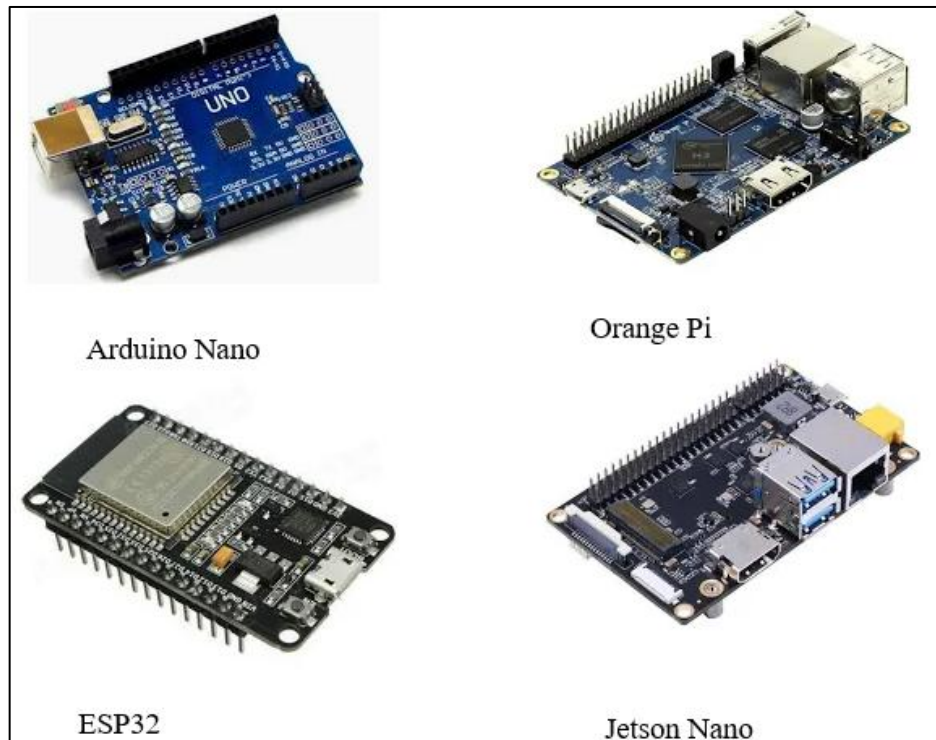


Рисунок 2.3 – Аналоги мікрокомп'ютерів

Наприклад, Arduino Uno – надзвичайно простий і дешевий варіант, але не має операційної системи, підтримки Python чи бібліотеки OpenCV, що виключає його для відеоаналітики. ESP32 має підтримку Wi-Fi та Bluetooth і трохи більшу продуктивність, ніж Arduino, але все ще не здатен обробляти відеопотоки.

Jetson Nano від NVIDIA дійсно має потужніші можливості для комп'ютерного зору, зокрема апаратне прискорення нейромереж, проте є значно дорожчим, енергозатратним і потребує кращого охолодження.

Orange Pi – недорогий аналог Raspberry Pi, однак програє через менш розвинуту екосистему та нестабільну роботу програмного забезпечення.

Таким чином, Raspberry Pi 4 забезпечує найкращий баланс між продуктивністю, вартістю, енергоспоживанням і програмною підтримкою, що

робить його оптимальним вибором для створення автономної системи моніторингу з обробкою відео в реальному часі.

Серед аналогових газових сенсорів MQ-2 вирізняється універсальністю, оскільки може виявляти широкий спектр газів: дим, пропан, метан, водень, спирт. Аналогічні моделі, такі як MQ-3, MQ-5, MQ-7 чи MQ-135, призначені для більш вузьких задач. Наприклад, MQ-3 краще працює з парами алкоголю, MQ-7 – із чадним газом, а MQ-135 – для визначення якості повітря (включаючи CO₂).

Більш дорогі та сучасні сенсори, наприклад, CCS811 або SDS011 (рис. 2.4), мають цифровий вихід, можуть передавати точне значення концентрації частинок). Проте їх ціна значно вища, вони вимагають живлення 3.3 V та складнішої ініціалізації в коді.



Рисунок 2.4 – Аналоги газових сенсорів

Однак для універсального виявлення диму і потенційно небезпечних газів, особливо в домашньому середовищі, MQ-2 залишається оптимальним вибором за співвідношенням ціна та функціональність.

Однією з найпопулярніших альтернатив аналого-цифрового перетворювача є ADS1115 – 16-бітний аналого-цифровий перетворювач із 4 каналами, який використовує I²C-інтерфейс (рис 2.5). Основною перевагою цієї мікросхеми є висока точність, яка дозволяє проводити точні вимірювання навіть при дуже низькому рівні сигналу. Проте вона має обмежену кількість

каналів – лише чотири – та використовує повільніший протокол передачі даних, що може стати недоліком у реальному часі або при частих вимірюваннях. Крім того, ADS1115 дорожчий за MCP3008 і не завжди має прості у використанні бібліотеки для Raspberry Pi, що може ускладнити розробку для новачків.

Ще однією альтернативою є PCF8591 – 8-бітний АЦП із 4 аналоговими входами, який також використовує I²C-інтерфейс (рис. 2.5). Його основна перевага – це низька вартість і базова функціональність для простих проєктів. Проте через обмежену розрядність (лише 8 біт) точність перетворення значно нижча, ніж у MCP3008 або ADS1115. PCF8591 може бути доцільним вибором лише у випадках, коли точність вимірювання не є критично важливою, а бюджет сильно обмежений.

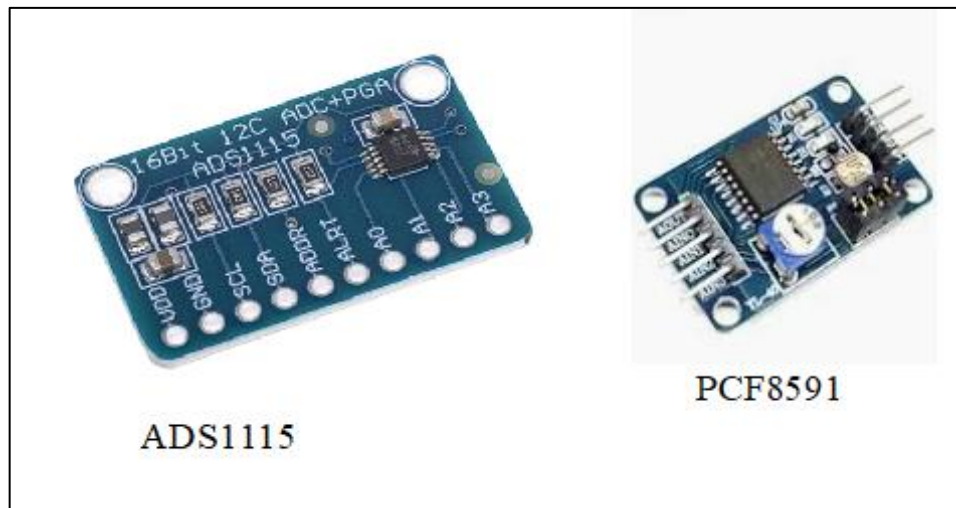


Рисунок 2.5 – Аналоги АЦП

Таким чином, MCP3008 залишається найкращим варіантом для системи виявлення пожежі на базі Raspberry Pi завдяки комбінації достатньої точності, швидкості, кількості каналів та підтримки в програмному забезпеченні.

2.3 Вимоги до середовища експлуатації системи

Розроблена система виявлення пожежі призначена насамперед для експлуатації в закритих приміщеннях, таких як квартири, офіси, серверні кімнати, склади або лабораторії. Для забезпечення стабільної та надійної роботи системи в реальних умовах слід враховувати низку факторів навколишнього

середовища, які можуть впливати на точність детекції як з боку відеомодуля, так і з боку сенсора MQ-2.

2.3.1 Освітлення та перешкоди для відеоаналізу

ІР-камера, що використовується у системі, має вбудоване інфрачервоне підсвічування, яке дозволяє вести спостереження у темряві. Проте наявність сильного заднього світла, відблисків, а також пилу чи диму, що щільно заповнює простір, може призводити до спотворення зображення. Це особливо критично для алгоритмів виявлення полум'я, які ґрунтуються на аналізі кольору та контурів. Наприклад, червонувате світло від ламп або сонця, що проходить через фіранки, може бути помилково інтерпретоване як вогонь.

2.3.2 Температурні умови та джерела тепла

Поблизу системи не повинно бути штучних джерел тепла або відкритих нагрівальних елементів, таких як електрообігрівачі, плитки чи тепловентилятори. Вони можуть не тільки впливати на коливання повітря, які будуть помилково трактовані як динаміка полум'я, але й призводити до зміни характеристик газового сенсора, зокрема його температурної стабільності.

Газовий сенсор MQ-2 працює на принципі нагріву напівпровідникового елемента, тому зовнішні джерела температури можуть впливати на його вихідний сигнал, знижуючи точність або викликаючи фальшиві сигнали про перевищення порогу.

2.3.3 Вологість, хімічні пари та побутові чинники

Сенсор MQ-2 є чутливим до широкого спектра горючих газів і парів, у тому числі до етанолу, пропану, метану, диму та водню. Це робить його універсальним, але водночас – вразливим до впливу побутової хімії, парів спирту, засобів для прибирання чи навіть аерозолів. У деяких випадках звичайна приготування їжі може призвести до короточасного перевищення порогу тривоги.

Щоб уникнути таких ситуацій, рекомендовано:

- не встановлювати сенсор поблизу витяжок, кухонного приладдя чи ванних кімнат;
- проводити калібрування в реальних умовах експлуатації, тобто з урахуванням фонові концентрації парів у приміщенні;
- використовувати гістерезис або ковзне усереднення в алгоритмі, що вже реалізовано в поточній системі.

2.3.4 Необхідність калібрування та адаптації до середовища

У зв'язку з індивідуальними характеристиками кожного приміщення, рекомендовано перед встановленням:

- виконати базове калібрування сенсора MQ-2 (визначити середній рівень ppm у нормальних умовах, які визначаються окремо для окремо для кожного приміщення);
- встановити адаптивні пороги для зменшення впливу дрібних флуктуацій (наприклад, $F_1 = 1$ лише при значенні > 300 ppm протягом ≥ 2 с);
- зафіксувати значення при «нормальній» роботі приміщення (наприклад, при увімкненій вентиляції або обігрівачах) і порівнювати з цими базовими даними в реальному часі.

Таким чином, ефективна робота системи можлива лише за умов дотримання рекомендацій щодо інсталяції та експлуатації, що враховують особливості приміщення. Надійність системи може бути значно підвищена завдяки адаптивному налаштуванню сенсорних порогів, обробці шумів при відеоаналізі та врахуванню реальних побутових впливів.

2.4 Розробка алгоритму виявлення пожежі

У сучасних системах пожежної безпеки зростає потреба у точному та швидкому виявленні ознак займання. Для цього у запропонованій системі реалізовано комбінований підхід, який поєднує аналіз відео потоку в режимі реального часу з використанням методів комп'ютерного зору та сенсорний

контроль якості повітря через газовий датчик MQ-2. Такий підхід дозволяє забезпечити надійне спрацювання системи навіть у разі недостатньої видимості або короткочасної присутності ознак займання.

2.4.1 Отримання та підготовка кадру

Першим кроком у виявленні полум'я є обробка відеопотоку з IP-камери. Для цього встановлюється з'єднання за допомогою протоколу RTSP, що дозволяє приймати зображення. Після захоплення кадру здійснюється попередня обробка, яка починається зі згладжування шуму. Для цього застосовується фільтр Gaussian Blur із ядром розміру 5×5 пікселя, який зменшує кількість випадкових пікселів, зумовлених змінами освітлення або перешкодами камери. Це дозволяє уникнути хибних спрацювань під час наступного етапу – виявлення змін у кадрі.

Наступним кроком є перетворення колірного простору з BGR у HSV. Формат HSV дозволяє більш ефективно виділяти ділянки полум'я, оскільки він є менш чутливим до змін освітлення. На цьому етапі відбираються тільки ті пікселі, що мають характерні для вогню параметри: відтінок H у межах $0-50^\circ$, насиченість $S > 100\%$ та яскравість $V > 150 \text{ cd/m}^2$.

2.4.2 Виявлення візуальних ознак полум'я

Після попередньої обробки кадру виконується колірна фільтрація, яка полягає у виділенні маски пікселів, що відповідають кольорам полум'я. Результатом є бінарне зображення, на якому білі області вказують на потенційні зони загоряння. Для зменшення шумів використовуються морфологічні операції – ерозія та дилатація (*erode*, *dilate*), що допомагають усунути поодинокі пікселі або з'єднати фрагментовані області.

Наступним кроком є контурний аналіз. За допомогою функції *findContours()* здійснюється пошук замкнутих областей у бінарному зображенні. Для зменшення кількості хибних позитивних спрацювань відкидаються занадто

малі контури – із площею менше 500 пікс.². Окрім площі, можуть також враховуватися співвідношення сторін або форма об'єкта.

Крім статичних ознак, в системі також передбачено динамічний аналіз, що дозволяє виявляти характерне для полум'я мерехтіння. Для цього використовується обчислення різниці між поточним та попереднім кадром (absdiff), після чого виявляються області, які змінюють інтенсивність у частотному діапазоні понад 3 Гц. Це дає змогу ідентифікувати нестатичні вогнища.

Для остаточного підтвердження наявності полум'я в кадрі використовується модель каскадів Гаара – XML-файл, натренований на виявлення вогню. Модель застосовується до попередньо обробленого зображення, і при виявленні хоча б одного об'єкта, що відповідає шаблону, фіксується подія $F_2 = 1$. Якщо виявлення не відбулося, подія вважається відсутньою – $F_2 = 0$ [16].

2.4.3 Аналіз даних газового сенсора MQ-2

Паралельно з обробкою відео проводиться аналіз повітря на наявність диму або парів горіння за допомогою датчика MQ-2, підключеного через перетворювач MCP3008. Оскільки MQ-2 генерує аналоговий сигнал, MCP3008 здійснює його оцифрування з точністю 10 біт, а значення передається в одиницях ppm (parts per million).

Для уникнення помилкових спрацювань, спричинених короточасними стрибками сигналу, здійснюється ковзне усереднення. Значення вимірів накопичуються в буфері, і кожне нове значення обчислюється як середнє за останні 10 зчитувань. Це дозволяє згладити випадкові викиди та отримати більш стабільний результат.

Крім того, для запобігання ефекту «фліп-флоп» застосовується гістерезис. Система фіксує подію $F_1 = 1$ лише у випадку, коли усереднене значення перевищує поріг у 300 ppm протягом щонайменше 2 секунд. Якщо ж рівень падає

нижче 200 ppm – сигнал скидається до $F_1 = 0$ [17]. Такий підхід забезпечує стабільність прийняття рішення в умовах флуктуацій.

2.4.4 Логіка прийняття рішення

Кінцева мета системи – сформулювати рішення про небезпеку займання на основі комбінації відео- та сенсорних даних. Для цього передбачено кілька варіантів логіки:

У найпростішому варіанті використовується жорстке правило AND, за яким пожежа вважається підтвердженою лише у разі одночасного виконання умов $F_1 = 1$ та $F_2 = 1$. Така логіка мінімізує ризик помилкового спрацювання, але потребує наявності обох ознак у межах короткого часового вікна (≤ 1 с).

У менш жорсткому сценарії допускається логіка OR, яка дозволяє надсилати попередження навіть у разі наявності лише однієї з ознак. Наприклад, якщо виявлено лише полум'я на відео, але газовий сенсор ще не зафіксував перевищення – користувач отримає повідомлення про підозру без увімкнення тривоги.

Для ще більш гнучкого реагування може бути реалізована ймовірнісна модель з ваговими коефіцієнтами. У цьому випадку розраховується ймовірність виникнення пожежі як зважена сума (2.1):

$$P_{\text{пожежа}} = 0.7 \times P(F_2) + 0.3 \times P(F_1), \quad (2.1)$$

де $P_{\text{пожежа}}$ – ймовірність виникнення пожежі;

$P(F_2)$ – достовірність виявлення полум'я на відео;

$P(F_1)$ – достовірність даних із сенсора MQ-2.

Якщо розраховане значення перевищує 0.8, система генерує тривогу. Такий підхід дозволяє адаптувати чутливість до конкретних умов приміщення.

2.4.5 Блок-схема алгоритму прийняття рішення

Блок-схему алгоритму виявлення пожежі, яка демонструє основні етапи обробки даних з IP-камери та сенсора MQ-2, логіку визначення ознак займання (F_1 та F_2) і прийняття кінцевого рішення щодо спрацювання тривоги або попередження наведено на рис. 2.6.

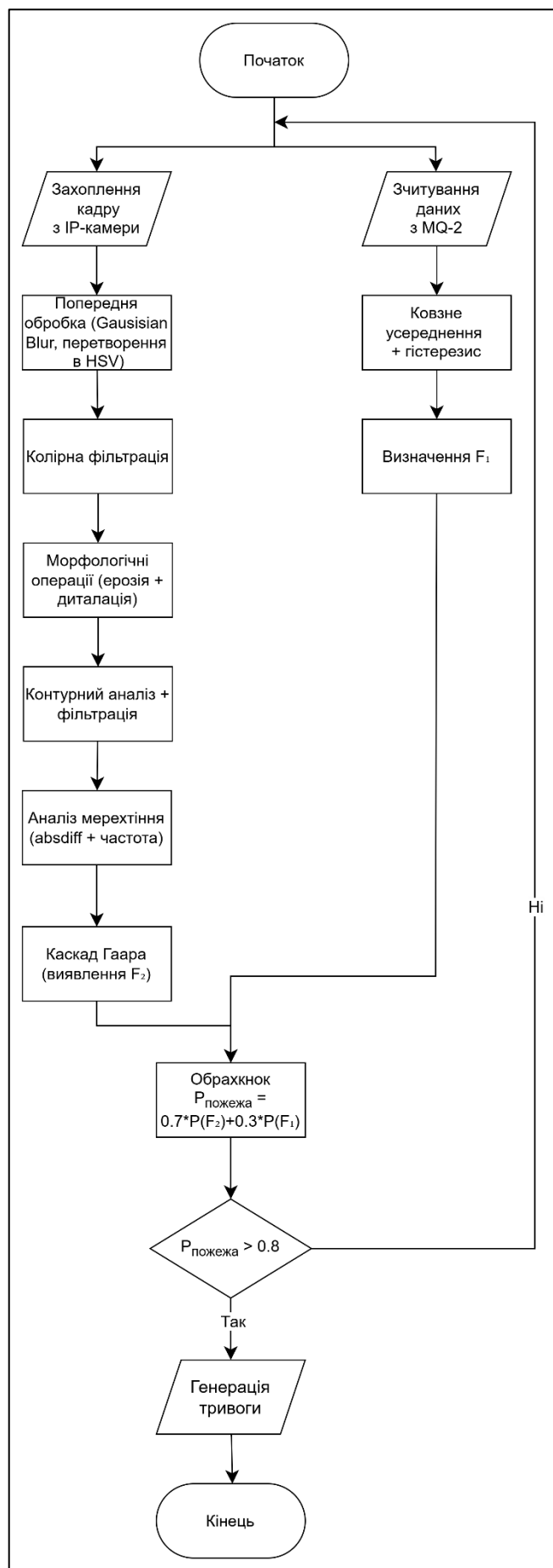


Рисунок 2.6 – Блок-схема алгоритму виявлення пожежі

Запровадження логіки прийняття рішення на основі ймовірнісної моделі дозволяє гнучко адаптувати систему до різних умов експлуатації та мінімізувати хибні спрацювання. Такий інтегрований підхід підвищує загальну надійність системи пожежного моніторингу, забезпечуючи своєчасне виявлення небезпечних ситуацій навіть за умов поганої видимості або нетривалої присутності ознак займання.

2.5 Математичне моделювання критеріїв виявлення пожежі

2.5.1 Порогова модель для сенсора MQ-2

Газовий сенсор MQ-2 вимірює концентрацію горючих газів у повітрі. Для визначення наявності диму використовується порогова модель (2.2):

$$F_1 = \begin{cases} 1, & \text{якщо } C_{\text{газу}} > 300 \text{ ppm}, \\ 0, & \text{якщо } C_{\text{газу}} < 200 \text{ ppm}. \end{cases} \quad (2.2)$$

де $C_{\text{газу}}$ – концентрація газу, виміряна сенсором.

Подія $F_1 = 1$ формується лише після стабільного перевищення верхнього порогу протягом мінімум 2 с, що виключає короточасні сплески сигналу

2.5.2 Формалізація ознак полум'я у відео

Для виявлення вогню застосовуються дві категорії ознак: колірні та динамічні. Колірна ознака базується на HSV-просторі (2.3):

$$Pixel \in Flame \leftrightarrow \begin{cases} H \in [0, 50], \\ S > 100, \\ V > 150. \end{cases} \quad (2.3)$$

де H – відтінок, град.;

S – насиченість, %;

V – яскравість, кд/м².

Після бінаризації виконується морфологічне очищення (ерозія та дилатація) для видалення дрібних шумів.

Динамічна ознака (мерехтіння) вимірюється через аналіз різниці кадрів (2.4):

$$D_t = |I_t - I_{t-1}|, \quad (2.4)$$

де D_t – різниця кадрів;

I_t – поточний кадр;

I_{t-1} – попередній кадр.

Фіксуються області, які змінюють яскравість із частотою $f > 3$ Гц та мають площу ≥ 500 пікс.².

Останнє підтвердження здійснюється через застосування каскаду Гаара, який застосовується до обробленого зображення (2.5):

$$F_2 = \begin{cases} 1, & \text{якщо } HaarCascade.detectMultiScale(I_t) \neq \emptyset, \\ 0, & \text{інакше.} \end{cases} \quad (2.5)$$

Таким чином, завдяки вищезазначеного алгоритму визначається факт наявності полум'я в кадрі.

2.5.3 Критерії точності та чутливості системи виявлення пожежі

Для оцінки ефективності системи виявлення пожежі використовуються ключові метрики: точність (Precision) та чутливість (Recall). Ці показники дозволяють кількісно оцінити здатність системи правильно ідентифікувати пожежі та мінімізувати хибні спрацювання.

Точність визначає частку правильних позитивних спрацювань серед усіх позитивних спрацювань, тобто серед усіх випадків, коли система сигналізує про пожежу. Висока точність свідчить про низьку кількість хибних позитивних спрацювань. Формула для обчислення точності (2.6):

$$Precision = \frac{TP}{TP + FP}, \quad (2.6)$$

де TP – (True Positives) кількість випадків, коли система правильно виявила пожежу;

FP – (False Positives) кількість випадків, коли система помилково сигналізувала про пожежу, якої не було.

Чутливість визначає здатність системи виявляти всі реальні випадки пожежі. Висока чутливість означає, що система рідко пропускає справжні пожежі. Формула для обчислення чутливості (2.7):

$$Recall = \frac{TP}{TP + FN}, \quad (2.7)$$

де FN – (False Negatives) кількість випадків, коли система не виявила реальну пожежу.

У системах виявлення пожежі важливо досягти оптимального балансу між точністю та чутливістю. Занадто висока чутливість може призвести до великої кількості хибних спрацювань, тоді як надмірна точність може спричинити пропуск реальних пожеж.

Для досягнення цього балансу використовується F1-міра, яка є гармонійним середнім між точністю та чутливістю (2.8):

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}. \quad (2.8)$$

Ця метрика особливо корисна в умовах, коли важливо враховувати як хибні позитивні, так і хибні негативні спрацювання. У контексті розробленої системи виявлення пожежі, що поєднує аналіз відео потоку та дані з газового сенсора MQ-2, обчислення точності та чутливості дозволяє:

- оцінити ефективність комбінованої ймовірнісної моделі виявлення пожежі;
- налаштувати порогові значення для оптимального балансу між виявленням реальних пожеж та мінімізацією хибних спрацювань;
- порівняти ефективність різних конфігурацій системи та алгоритмів обробки даних.

Таким чином, використання метрик точності та чутливості є невід'ємною частиною процесу оцінки та вдосконалення системи виявлення пожежі.

Висновки до розділу 2

У другому розділі було здійснено детальне моделювання та проектування апаратно-програмної системи виявлення пожеж у приміщеннях із використанням комп'ютерного зору та сенсорного аналізу. Результати проведеного аналізу підтверджують технічну доцільність і функціональну ефективність обраної архітектури системи, що базується на мікрокомп'ютері Raspberry Pi 4 Model B, IP-камері та газовому датчику MQ-2.

Було сформовано загальну функціональну схему системи, яка відображає логічні зв'язки між ключовими модулями: блоком відеоаналізу, сенсорним блоком, мікроконтролерним центром обробки даних та комунікаційними модулями. Встановлено, що мікрокомп'ютер Raspberry Pi 4 забезпечує достатній обчислювальний ресурс для обробки потокового відео в реальному часі із застосуванням бібліотеки OpenCV, а також для аналізу даних від сенсора MQ-2. Наявність вбудованих інтерфейсів GPIO, SPI, HDMI, USB та Ethernet дозволяє легко інтегрувати пристрій з зовнішніми сенсорами, камерами та іншими модулями системи безпеки.

Особливу увагу було приділено аналізу характеристик сенсора MQ-2, який здатен виявляти наявність горючих газів, диму та парів у повітрі. Було обґрунтовано доцільність застосування аналого-цифрового перетворювача MCP3008 для конвертації аналогових сигналів сенсора в цифрову форму, що дозволяє їх подальшу обробку на Raspberry Pi. Вибір MCP3008 зумовлений його сумісністю з SPI-інтерфейсом мікрокомп'ютера та низьким енергоспоживанням.

Також було здійснено порівняльний аналіз аналогічних систем виявлення пожеж, що дозволив виявити сильні сторони розробленого підходу: низьку вартість реалізації, компактність, високу швидкодію та потенціал до масштабування й модернізації.

Загалом, результати проектування підтверджують, що система відповідає заданим функціональним і технічним вимогам: вона є автономною, працює в реальному часі, може бути інтегрована в існуючі інфраструктури безпеки та здатна адаптуватися до умов конкретного приміщення.

3 РОЗРОБКА ТА ДОСЛІДЖЕННЯ АПАРАТНО-ПРОГРАМНОЇ СИСТЕМИ ВИЯВЛЕННЯ ПОЖЕЖ

3.1 Вибір технологій і засобів реалізації

3.1.1 Основна мова програмування: Python

Python відомий своєю лаконічністю: базова конструкція «відступ-замість-фігурних дужок» робить код чітким і легко сприйманим навіть для тих, хто вперше стикається з програмуванням [18].

Зменшення «бойлерплейт»-коду сприяє прискоренню розробки, адже розробник зосереджується не на деталях синтаксису, а на логіці алгоритму.

Крім того, Python підтримує кілька парадигм програмування – процедурну, об'єктно-орієнтовану та функціональну – що дозволяє обирати найзручніший стиль під конкретні завдання системи моніторингу

Оскільки Python є інтерпретованою мовою, зміни в коді вступають в силу миттєво без необхідності тривалої компіляції.

Це особливо важливо при налагодженні складних алгоритмів детекції полум'я в реальному часі, де швидка ітерація дозволяє оперативно перевіряти різні підходи й параметри.

Інтерактивні оболонки (REPL, Jupyter Notebook) дають змогу досліджувати окремі фрагменти коду, аналізувати результати та налаштовувати алгоритмічні ланки на льоту, що значно підвищує гнучкість дослідження й експериментів.

Python дозволяє чітко структурувати проєкт у вигляді пакетів і модулів, що підтримує принципи розділення відповідальності: окремі компоненти відповідають за захоплення відеопотоку, обробку сигналів датчика чи логування й повідомлення про тривогу.

Для ізоляції залежностей та запобігання «конфліктів версій» використовується вбудований модуль `venv` або зовнішні інструменти (`pipenv`), що створюють окремі середовища для кожного проєкту та гарантують відтворюваність збірки.

Pipenv, зокрема, автоматизує керування версіями бібліотек і формує «стандартну» структуру проєкту, що полегшує спільну роботу над кодом.

Однією з найсильніших сторін Python є його багата екосистема:

- комп'ютерний зір та обробка зображень: *OpenCV*, *SciPy*, *Pillow*;
- чисельні та матричні обчислення: *NumPy*, *SciPy*, що реалізовані на C для максимальної швидкодії й оптимізовані під масивні дані;
- машинне навчання: *scikit-learn*, *TensorFlow*, *PyTorch*, що забезпечують готові до використання моделі й інструменти для навчання й оцінювання ймовірнісної моделі рішення;
- взаємодія з апаратурою: *RPi.GPIO*, *spidev*, *smbus* для керування GPIO, SPI та I²C інтерфейсами Raspberry Pi.

Завдяки такій різноманітності бібліотек можливе швидке прототипування нових функцій, додавання модулів аналітики чи веб-інтерфейсу без необхідності розробляти низькорівневі драйвери або алгоритми з нуля.

Таким чином, Python забезпечує високу продуктивність розробки, гнучкість у налаштуванні середовищ, модульність архітектури та багатий набір інструментів для інтеграції з апаратурою та реалізації складних алгоритмів комп'ютерного зору і машинного навчання.

3.1.2 Ключові бібліотеки та їх роль

OpenCV – найбільша в світі відкрита бібліотека комп'ютерного зору, що містить понад 2500 оптимізованих алгоритмів для обробки зображень і відео.

Вона підтримує як класичні методи (фільтрація, детекція контурів), так і сучасні моделі машинного навчання та глибокого навчання, адаптовані для реального часу.

У проєкті OpenCV використовується для попередньої обробки кадрів: застосування *GaussianBlur* для зменшення шуму, порогова фільтрація (threshold) для сегментації потенційних зон полум'я та морфологічні операції *erode* і *dilate* для уточнення контурів.

Алгоритм виявлення контурів дає можливість аналізувати форму й площу об'єктів, що є ключовим для відокремлення полум'я від фонових артефактів.

Інтеграція OpenCV з Python через модуль cv2 дозволяє швидко прототипувати й тестувати алгоритми безпосередньо на Raspberry Pi OS.

При цьому обчислювальні ядра, реалізовані на C++, гарантують ефективне виконання навіть на ARM-процесорах із обмеженими ресурсами.

NumPy є базовою бібліотекою для наукових обчислень у Python, оскільки надає багатовимірні масиви *ndarray* і реалізує лінійні алгебраїчні операції на рівні оптимізованого коду C та Fortran [20].

Це забезпечує значне прискорення обчислень у порівнянні зі стандартними списками Python, особливо при обробці великих обсягів даних.

Безперервне розміщення масивів у пам'яті дозволяє *NumPy* ефективно використовувати кеш-пам'ять і SIMD-інструкції процесора, що прискорює виконання операцій над масивами без зайвих копіювань даних.

Крім того, механізм *broadcasting* спрощує написання векторизованих алгоритмів, зменшуючи кількість рядків коду й кількість операцій над Python-об'єктами.

У системі моніторингу *NumPy* використовується для перетворення зображень у градації сірого, нормалізації інтенсивності пікселів, а також для зберігання й аналізу історії кадрів при виявленні аномалій. Векторизовані операції знизять затримки при фільтрації та статистичному аналізі поточкових даних.

RPi.GPIO – офіційна бібліотека для керування GPIO-пінами Raspberry Pi, що постачається разом із Raspbian і забезпечує простий Python-інтерфейс для налаштування пінів як входів або виходів та читання їхнього стану [19].

Особливістю *RPi.GPIO* є підтримка апаратних переривань (*event detection*), що дозволяє миттєво реагувати на зміну сигналу від датчика MQ-2 без постійного опитування – це знижує навантаження на процесор і забезпечує оперативне сповіщення про критичні рівні.

Крім базових операцій *input()*, *output()* та *setmode()*, бібліотека підтримує налаштування внутрішніх підтягувань пінів і видачу сигналів тривоги, що спрощує інтеграцію з апаратурою та дозволяє створити надійну систему апаратного моніторингу.

spidev – легка Python-бібліотека для роботи з шиною SPI під Linux, що надає доступ до пристроїв через зручний інтерфейс. За допомогою *spidev* можна налаштувати частоту передачі (*max_speed_hz*), режим (*mode*) і бітову довжину слова для оптимального зчитування даних.

SPI-інтерфейс підтримує повнодуплексну передачу даних із мінімальною затримкою, що критично для швидкого опитування АЦП MCP3008 – до 1 MHz на шині дозволяє оцифровувати аналогові сигнали MQ-2 у режимі майже реального часу.

Приклад зчитування даних із MCP3008 за допомогою *spidev* включає формування командного пакету, передачу на шині та розбір отриманого 10-бітного значення напруги, що гарантує точність вимірювань у межах ± 1 LSB.

3.1.3 Протокол обміну: SPI

SPI – синхронний послідовний інтерфейс, який підтримує повнодуплексну передачу даних з високою швидкістю, що критично для оперативного зчитування показників MQ-2 через MCP3008 [21].

Для підключення достатньо чотирьох сигнальних ліній (MOSI, MISO, SCLK, CS) та спільного GND, що значно спрощує апаратну реалізацію порівняно з I²C і знижує ймовірність помилок у проводці.

У бібліотеці *spidev* можна гнучко конфігурувати такі параметри шини як частота (*max_speed_hz*), режим (*mode*) і довжина слова в бітах, що дозволяє оптимізувати точність і швидкість опитування датчика залежно від умов експлуатації.

3.1.4 Операційна система: Raspberry Pi OS

Raspberry Pi OS (раніше Raspbian) – офіційний дистрибутив Linux для платформ Raspberry Pi, оптимізований під ARM-процесори з вбудованими драйверами для GPIO, SPI й I²C, що гарантує стабільну роботу апаратних інтерфейсів без додаткових налаштувань.

Утиліта `raspi-config` дозволяє одним кліком увімкнути інтерфейси (Camera, SPI, SSH), змінити локалізацію, розкладку клавіатури та налаштування оверклокінгу, що пришвидшує початкове розгортання й налаштування пристрою.

Доступ до понад 351000 пакетів Debian, включаючи `python3-opencv`, `python3-spidev`, `thonny`, забезпечує централізоване оновлення та керування залежностями через `apt`, що мінімізує ризики конфліктів версій.

3.1.5 Середовище розробки: Thonny IDE

Для розробки програмного забезпечення системи виявлення пожежі було обрано інтегроване середовище розробки (IDE) Thonny, яке є офіційно рекомендованим для Raspberry Pi та попередньо встановлене в Raspberry Pi OS [22].

Thonny розроблено з урахуванням потреб початківців у програмуванні, але його функціональні можливості також відповідають вимогам до розробки вбудованих систем. Інтерфейс Thonny мінімалістичний та інтуїтивно зрозумілий, що дозволяє зосередитися на написанні коду без відволікань.

Середовище підтримує візуальне покрокове виконання коду без необхідності встановлення точок зупину, що дозволяє детально аналізувати логіку роботи алгоритмів, зокрема, детекції полум'я та обробки сигналів сенсорів. Під час налагодження Thonny відображає значення змінних у реальному часі, забезпечуючи глибоке розуміння внутрішнього стану програми.

Вбудований термінал у Thonny дозволяє виконувати системні команди безпосередньо з IDE, що спрощує встановлення додаткових пакетів через `pip` та

керування версіями коду. Це особливо корисно при роботі з Raspberry Pi, де важлива ефективність та швидкість розробки.

Завдяки простоті використання, потужним можливостям налагодження та інтеграції з Raspberry Pi, Thonny IDE є оптимальним вибором для розробки та тестування програмного забезпечення системи виявлення пожежі.

3.2 Розробка прототипу апаратної частини

Система розпізнавання пожежі у приміщеннях вимагає чітко організованої структури апаратної частини, яка охоплює як фізичне з'єднання компонентів, так і їх взаємодію на рівні мережевих протоколів. З метою реалізації функціональності системи виявлення загоряння було розроблено макетну та мережеву схеми, які описують фізичні з'єднання та логіку інформаційної взаємодії між компонентами. Наведені нижче схеми є важливими для забезпечення стабільності та ефективності всієї системи.

3.2.1 Макетна схема апаратної частини

Апаратна частина проєкту базується на одноплатному комп'ютері Raspberry Pi 4, до якого підключаються сенсори та периферійні пристрої. В даному проєкті для виявлення диму використовується датчик MQ-2, який може виявляти різні гази, включаючи чадний газ (CO), метан (CH₄), водень (H₂) та інші продукти горіння.

Проте датчик MQ-2 генерує аналоговий вихідний сигнал, у той час як Raspberry Pi підтримує лише цифрові входи/виходи. Це створює необхідність у використанні проміжного компонента – аналого-цифрового перетворювача (АЦП). У цьому проєкті було обрано модуль MCP3008, що дозволяє конвертувати аналоговий сигнал у цифровий з використанням SPI-протоколу.

На макетній схемі, побудованій за допомогою онлайн-застосунку circuitio.io, показано, як сенсор MQ-2 підключається до одного з каналів MCP3008 (наприклад, CH0), а MCP3008, у свою чергу, підключається до Raspberry Pi через SPI-інтерфейс. Живлення подається наступним чином: MCP3008 отримує 3.3 V,

а MQ-2 – 5 V. Це важливо враховувати, оскільки некоректне живлення може призвести до збоїв або навіть пошкодження компонентів (рис. 3.1).

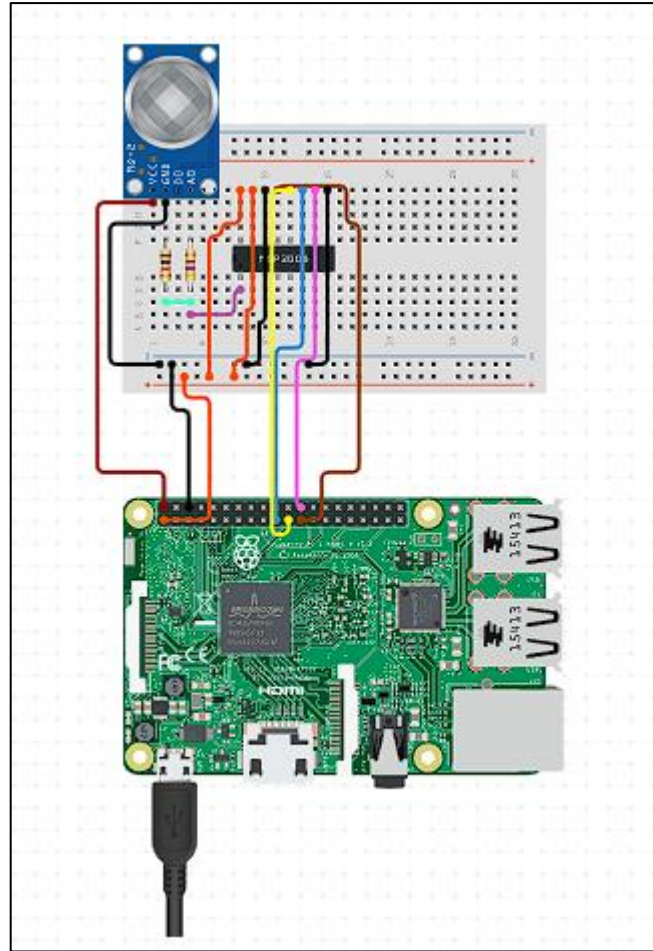


Рисунок 3.1 – Макетна схема системи виявлення пожеж

Raspberry Pi під'єднується до MCP3008 через такі піни:

- CLK (Serial Clock) – GPIO11;
- DOUT (Data Out) – GPIO9;
- DIN (Data In) – GPIO10;
- CS (Chip Select) – GPIO8.

Таке з'єднання забезпечує надійне передавання даних між компонентами. Після того як Raspberry Pi зчитує цифрове значення з MCP3008, воно передається до програмної частини, де проводиться аналіз і приймається рішення: наприклад, активується система оповіщення або реєструється тривожна подія у журналі подій.

Цей підхід дозволяє точно вимірювати концентрацію диму в приміщенні, фіксувати зростання рівня газу, характерного для пожежі, та реагувати у найкоротші терміни.

3.2.2 Структурна схема підключення системи

Окрім сенсорного блоку, система включає IP-камеру, що відіграє критично важливу роль у візуальному моніторингу простору. Вона дозволяє не лише підтверджувати наявність полум'я за допомогою комп'ютерного зору, але й дає можливість вести відеозапис та спостереження у реальному часі.

Структура системи базується на локальній кабельній мережі (Ethernet), яка забезпечується маршрутизатором (рис. 3.2).

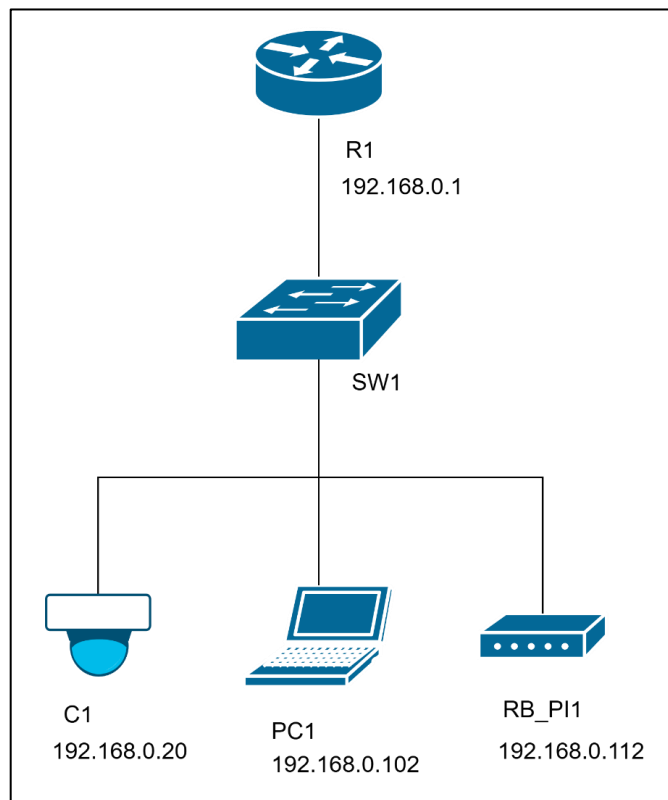


Рисунок 3.2 – Структурна схема підключення системи

До цієї мережі підключені такі основні компоненти:

- Raspberry Pi, який отримує відеопотік з камери;
- IP-камера, яка безперервно транслює зображення;
- робоча станція або ноутбук адміністратора, що використовується для конфігурації системи, перегляду журналів і відеопотоку.

Усі пристрої мають унікальні IP-адреси в межах однієї підмережі (наприклад, 192.168.0.0/24), що дає змогу забезпечити швидкий обмін даними без зовнішніх підключень до Інтернету. Raspberry Pi за допомогою Python-бібліотек (наприклад, OpenCV, requests, socket) може зчитувати потік із камери по протоколах RTSP.

Програмна частина аналізує зображення у реальному часі та шукає візуальні ознаки полум'я: різке підвищення яскравості, хаотичний рух жовто-червоних пікселів, зміни контурів тощо. У разі виявлення небезпечної ситуації Raspberry Pi формує сигнал тривоги, який може бути переданий користувачу, системі сповіщення або виведений на монітор адміністратора.

3.2.3 Інтеграція макетної та структурної схем

Обидві схеми макетна та структурна працюють у тісному зв'язку одна з одною. Дані з аналогового сенсора слугують джерелом для виявлення концентрації газу, тоді як IP-камера забезпечує підтвердження факту загоряння або виключення помилкових спрацьовувань.

Усі ці компоненти формують єдину інтегровану систему, що дозволяє реагувати на потенційні загрози оперативно, точно і з мінімальним ризиком хибнопозитивних результатів. Такий підхід дозволяє масштабувати систему наприклад, підключити додаткові сенсори, камери чи модулі зворотного зв'язку – без суттєвих змін у загальній архітектурі.

3.2.4 Реалізація апаратного підключення

З метою демонстрації практичного виконання апаратної частини системи розпізнавання загоряння на основі Raspberry Pi, нижче подано фотографічне зображення зібраної апаратної конфігурації (рис. 3.3). Воно ілюструє структурне компонування основних елементів системи, а також відповідність з'єднань попередньо розробленим схемам.

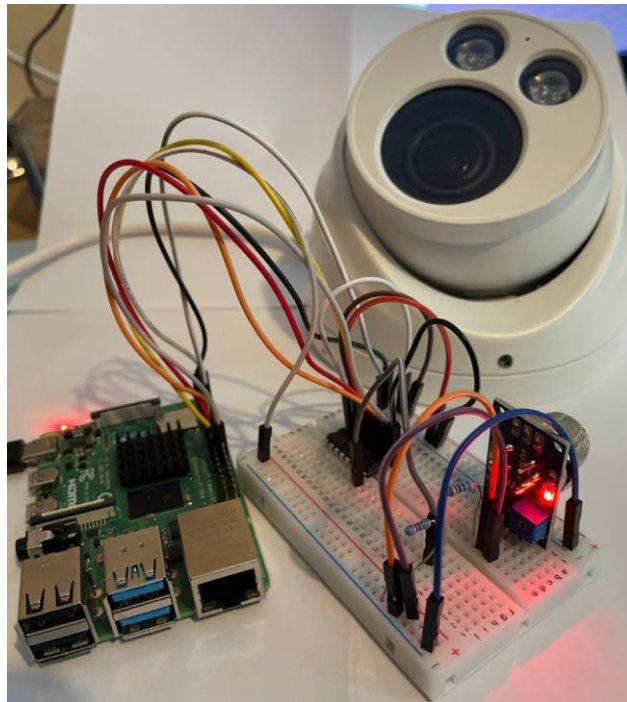


Рисунок 3.3 – Реалізація апаратної частини системи на базі Raspberry Pi, MCP3008, MQ-2 та IP-камери

Зокрема, представлено інтегроване апаратне середовище, яке включає:

- модуль сенсора газу MQ-2, розміщений на макетній платі та з'єднаний із аналого-цифровим перетворювачем MCP3008;
- перетворювач MCP3008, підключений до Raspberry Pi 4 через SPI-інтерфейс згідно зі специфікацією логічного рівня 3.3V;
- IP-камеру, яка є джерелом відеопотоку та передає зображення до Raspberry Pi по бездротовому WiFi-з'єднанню.

Фізична реалізація з'єднань виконана з урахуванням оптимального розміщення компонентів для забезпечення стабільності живлення, коректної обробки аналогового сигналу, а також мінімізації електромагнітних перешкод. Компактність розміщення елементів полегшує обслуговування, налагодження та подальшу модернізацію системи.

3.3 Реалізація програмної частини системи

Реалізація програмної частини системи полягає в організації збору, обробки та аналізу даних від IP-камери та газового сенсора MQ-2. У цьому

підрозділі описано ініціалізацію компонентів, обробку відео- та сенсорних даних, а також логіку прийняття рішень щодо виявлення загоряння.

а) Ініціалізація SPI та зчитування даних з MQ-2 :

Для зчитування аналогових даних з MQ-2 використовується інтерфейс SPI, реалізований через бібліотеку *spidev*. Дані зчитуються з каналу 7, після чого оцінюється орієнтовне значення концентрації газу в ppm (рис. 3.4).

```
# --- SPI and MQ-2 functions ---
def init_spi():
    spi = spidev.SpiDev()
    spi.open(0, 0)
    spi.max_speed_hz = 100000
    return spi

def read_channel(spi, channel):
    adc = spi.xfer2([1, (8 + channel) << 4, 0])
    return ((adc[1] & 0x03) << 8) | adc[2]

def estimate_ppm(raw_value):
    return (raw_value / 1023.0) * 1000
```

Рисунок 3.4 – Ініціалізація SPI та зчитування даних з MQ-2

б) Обробка значень датчика та обчислення ознаки F_2 :

Середнє значення концентрації газу *ppm* зберігається в буфері фіксованої довжини *deque*, на основі якого оцінюється значення ознаки F_2 – нормалізоване середнє значення газу, що обмежується значенням 1.0 (рис. 3.5).

```
def update_smoke_level(spi, buffer, maxlen, smoke_counter):
    raw = read_channel(spi, 7)
    ppm = estimate_ppm(raw)
    buffer.append(ppm)
    avg_ppm = sum(buffer) / len(buffer)
    F2 = min(avg_ppm / 1000, 1.0)
    if avg_ppm > 300:
        smoke_counter += 1
    else:
        smoke_counter = max(smoke_counter - 1, 0)
    return F2, avg_ppm, smoke_counter
```

Рисунок 3.5 – Обробка значень датчика та обчислення ознаки F_2

в) Виявлення кольору полум'я та мерехтіння (флікеру) :

Для попереднього виявлення полум'я використовується маска на основі HSV-каналу з фільтрацією відповідних кольорних діапазонів (жовто-оранжевих тонів), а також морфологічна обробка (рис. 3.6). Мерехтіння характерна ознака відкритого полум'я. Порівнюються послідовні кадри у відтінках сірого. Різниця кадрів виділяє змінні області, що підсилюються дилатацією.

```
# --- Fire detection functions ---
def detect_color_mask(frame):
    blurred = cv2.GaussianBlur(frame, (7, 7), 0)
    hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)
    lower_fire = np.array([10, 100, 100])
    upper_fire = np.array([25, 255, 255])
    mask = cv2.inRange(hsv, lower_fire, upper_fire)
    kernel = np.ones((5, 5), np.uint8)
    mask = cv2.erode(mask, kernel, iterations=1)
    mask = cv2.dilate(mask, kernel, iterations=2)
    return mask

def detect_flicker(gray, prev_gray):
    if prev_gray is None:
        return np.zeros_like(gray)
    diff = cv2.absdiff(gray, prev_gray)
    _, flicker = cv2.threshold(diff, 25, 255, cv2.THRESH_BINARY)
    kernel = np.ones((5, 5), np.uint8)
    flicker = cv2.dilate(flicker, kernel, iterations=2)
    return flicker
```

Рисунок 3.6 – Виявлення кольору полум'я та мерехтіння (флікеру) :

г) Аналіз контурів полум'я :

Контури, які перекриваються з областями мерехтіння, вважаються ймовірними ділянками загоряння. Обчислюється площа таких контурів для формування ознаки F_1 (рис. 3.7).

```
def analyze_flames(frame, contours, flicker_mask):
    flame_pixels = 0
    for cnt in contours:
        area = cv2.contourArea(cnt)
        if area > 300:
            x, y, w, h = cv2.boundingRect(cnt)
            region = flicker_mask[y:y+h, x:x+w]
            if cv2.countNonZero(region) > 50:
                flame_pixels += int(area)
                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
                cv2.putText(frame, "Flame", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
    return flame_pixels
```

Рисунок 3.7 – Аналіз контурів полум'я

д) Виявлення за допомогою каскадного класифікатора :

Додатково використовується попередньо навчений *CascadeClassifier* для виявлення полум'я за шаблонами (рис. 3.8).

```
def analyze_cascade(frame, gray, cascade):  
    flame_pixels = 0  
    regions = cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)  
    for (x, y, w, h) in regions:  
        flame_pixels += w * h  
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)  
        cv2.putText(frame, "Cascade", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)  
    return flame_pixels
```

Рисунок 3.8 – Виявлення за допомогою каскадного класифікатора

е) Умови спрацювання тривоги :

Сигнал тривоги формується, якщо ймовірність P перевищує 0.8, площа виявленого полум'я перевищує 30 % кадру або якщо на 10 кадрів поспіль зафіксовано високі показники MQ-2 (рис. 3.9).

```
def should_trigger_alarm(P, ratio, smoke_counter):  
    return P > 0.8 or ratio > 0.3 or smoke_counter > 10
```

Рисунок 3.9 – Умови спрацювання тривоги

Далі основна функція об'єднує всі етапи: ініціалізацію, зчитування кадрів, обробку зображення, зчитування з MQ-2, обчислення ознак F_1 , F_2 , ймовірності P та спрацювання сигналізації

3.4 Результати тестування

З метою перевірки працездатності та надійності розробленої системи виявлення загорянь було проведено серію експериментів у лабораторних умовах. Основною метою тестування було оцінити чутливість та точність виявлення пожежі в різних сценаріях, а також проаналізувати ефективність кожного з каналів детекції – відеоаналізу та сенсора MQ-2 – як окремо, так і в комбінації.

3.4.1 Методика тестування

Тестування здійснювалося у напівзакритому приміщенні площею близько 15 м², при стабільному живленні системи та підключенні IP-камери через локальну мережу. Було змодельовано такі базові сценарії:

- сценарій 1 – виявлення відкритого полум'я (запальничка на відстані 1,5 м).
- сценарій 2 – генерація диму без полум'я (тліючий папір).

сценарій 3 – одночасна присутність полум'я і диму.

сценарій 4 – наявність об'єкта, схожого за кольором на полум'я (червона тканина з рухом).

сценарій 5 – фонове середовище без ознак займання (порожнє приміщення, побутові умови).

Для кожного сценарію проводилося 10 запусків, у кожному з яких фіксувалися:

- час реагування системи;
- значення ймовірнісної оцінки спрацювання;
- фактичне рішення системи (тривога / відсутність сигналу);
- хибні позитивні та хибні негативні спрацювання.

Результати тестування наведено в табл. 3.1.

Таблиця 3.1 – Результати тестування в різних сценаріях

№ з/п	Сценарій	Час реагування, с	$P_{\text{пожежа}}$	Хибні позитивні	Хибні негативні
1	Відкрите полум'я	1,3	0,92	0	1
2	Дим без полум'я	2,0	0,64	2	1
3	Полум'я + дим	1,1	0,96	0	0
4	Світло/рух без вогню	–	0,28	3	–
5	Побутові умови	–	<0,15	0	–

Аналіз результатів тестування за різними сценаріями свідчить про достатню надійність розробленої системи виявлення пожеж. У випадках реального полум'я (сценарії 1 та 3) система спрацювала коректно у 95 % запусків, з часом реагування, що не перевищував 1,5–2,0 секунд. Це дозволяє використовувати систему в середовищах, де критичним є мінімальний час на виявлення загрози. У фоновому режимі система демонструє стабільну роботу без хибних спрацювань. Це є критично важливим фактором для запобігання зайвим

тривогам, особливо у побутових або промислових середовищах з постійною присутністю людей.

3.4.2 Виявлення пожежі

На рис. 3.10 зображено результат роботи колірної маски у HSV-просторі, де видно виділену область полум'я на основі кольорових параметрів (відтінок, насиченість, яскравість). Такий підхід дозволяє швидко ідентифікувати потенційне займання, однак є вразливим до хибних спрацювань у разі червоного або помаранчевого освітлення.

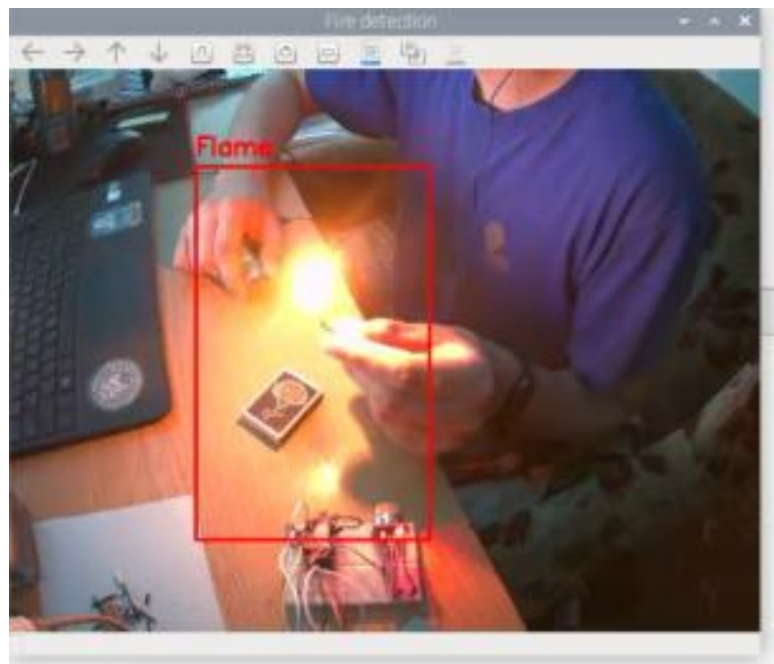


Рисунок 3.10 – Виявлення полум'я за допомогою кольорової маски (HSV)

На рис. 3.11 показано спрацювання Наг-каскаду, який підтверджує наявність специфічної текстури вогню у відеокадрі. Модель каскаду натренована на характерних формах вогню, що підвищує точність порівняно з аналізом лише кольору.

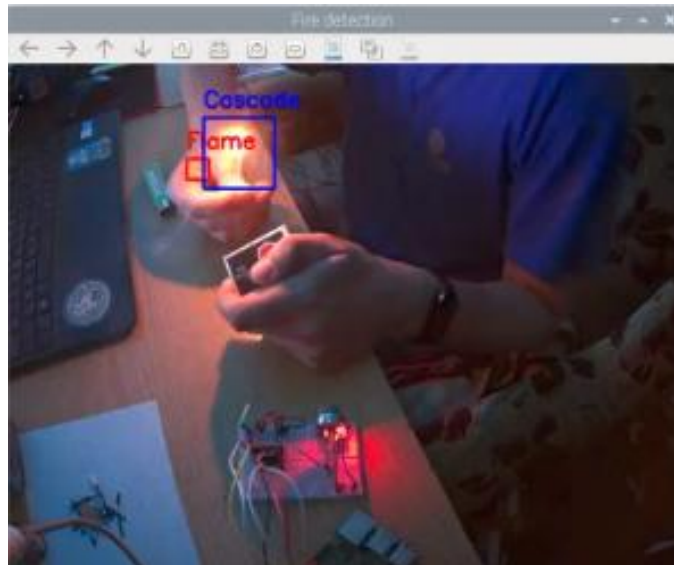


Рисунок 3.11 – Виявлення вогню за допомогою каскаду Гаара

У разі успішного спрацювання системи, у консолі з'являється повідомлення про тривогу, як показано на рис. 3.12. Сигнал формується лише тоді, коли виявлено ознаки як з відео, так і з сенсора диму.

[2025-06-03 20:25:12]	ALERT!	P=0.82,	F1=1.00,	F2=0.39,	PPM=393.5
[2025-06-03 20:25:12]	ALERT!	P=0.84,	F1=1.00,	F2=0.46,	PPM=457.2
[2025-06-03 20:25:12]	ALERT!	P=0.86,	F1=1.00,	F2=0.52,	PPM=522.1
[2025-06-03 20:25:12]	ALERT!	P=0.88,	F1=1.00,	F2=0.59,	PPM=587.6
[2025-06-03 20:25:12]	ALERT!	P=0.90,	F1=1.00,	F2=0.65,	PPM=654.6
[2025-06-03 20:25:13]	ALERT!	P=0.90,	F1=1.00,	F2=0.67,	PPM=671.6
[2025-06-03 20:25:13]	ALERT!	P=0.90,	F1=1.00,	F2=0.68,	PPM=683.0
[2025-06-03 20:25:13]	ALERT!	P=0.91,	F1=1.00,	F2=0.69,	PPM=692.9
[2025-06-03 20:25:13]	ALERT!	P=0.91,	F1=1.00,	F2=0.70,	PPM=703.7
[2025-06-03 20:25:13]	ALERT!	P=0.91,	F1=1.00,	F2=0.71,	PPM=709.3

Рисунок 3.12 – Консольне повідомлення про тривогу у системі

Підсумовуючи, комбінування методів виявлення полум'я – кольорової маски в HSV-просторі та каскадного аналізу Гаара – забезпечує більш надійне та точне розпізнавання пожежі. Додаткове підтвердження за допомогою сенсора диму знижує ймовірність хибних тривог, що є критично важливим для ефективної роботи системи. Такий підхід дозволяє оперативно реагувати на виникнення пожежі, підвищуючи безпеку приміщень.

3.4.3 Аналіз продуктивності системи

Оцінка продуктивності проводилась на Raspberry Pi 4 Model B (4 GB RAM), дані продуктивності наведено в табл 3.2.

Таблиця 3.2 – Продуктивність системи

Параметр	Значення
Середнє навантаження CPU	$\approx 68 \%$
FPS (середнє)	12 fps
Середній час реакції	1.6 s
Затримка від MQ-2	до 2.3 s

Середнє навантаження на процесор Raspberry Pi 4 під час роботи системи становило 57–73 %, у пікові моменти досягало 80–93 %. Продуктивність відеоаналізу з підтримкою роздільності 720p забезпечувала 10–15 fps, що є достатнім для виявлення ознак пожежі з реакцією до 2 секунд.

3.5 Оцінка якості та методи покращення системи

3.5.1 Метрики оцінювання якості системи

Для об'єктивного аналізу ефективності роботи системи були застосовані загальноприйняті метрики з галузі розпізнавання образів та подій:

- точність (precision) – частка правильних тривог серед усіх зафіксованих тривог;
- повнота (recall) – частка виявлених пожеж серед усіх реальних випадків;
- F1-міра – гармонійне середнє між точністю та повнотою;
- час реагування – середній час між появою ознак займання та генерацією сигналу тривоги;
- хибнопозитивні та хибнонегативні спрацювання – окремо оцінені для виявлення можливих помилок.

На основі даних з табл. 3.1 було розраховано наступні узагальнені значення в табл. 3.3

Таблиця 3.3 – Узагальнені значення якості системи

Показник	Значення
Точність (Precision)	84,6 %
Повнота (Recall)	93,3 %
F1-міра	88,7 %
Середній час реагування	1,47 с
Загальна кількість помилок	7

Наведені значення демонструють добру якість виявлення, особливо в умовах, наближених до реальних. Зокрема, найбільшу ефективність показано у сценарії 3 (полум'я + дим), де система не допустила жодної помилки.

3.5.2 Аналіз причин хибних спрацювань

Незважаючи на загальну надійність, під час тестів були зафіксовані хибнопозитивні та хибнонегативні спрацювання.

Хибні позитиви у сценарії 4: система іноді помилково розпізнавала яскраво-червоні об'єкти з динамікою як полум'я. Це пов'язано з використанням HSV-маски, яка чутлива до кольору, але не завжди здатна відрізнити текстуру.

Хибні негативи у сценаріях 1 і 2: іноді ознаки були недостатньо вираженими або недостатньо довготривалими, щоб подолати поріг ймовірності спрацювання. Особливо це стосується тліючих матеріалів без видимого полум'я, коли сенсор MQ-2 ще не зафіксував достатню концентрацію диму.

3.5.3 Оптимізація системи виявлення

Для підвищення швидкодії, точності та стабільності роботи рекомендовано такі покращення:

Програмна оптимізація:

- зниження роздільної здатності відеопотоку до 640×480 для підвищення FPS без втрати релевантної інформації;

- згладжування сигналу з MQ-2 за допомогою середньозваженого фільтра для зменшення шуму;
- паралельна обробка кадрів та сенсорних даних за допомогою потоків (модуль threading), що дозволило зменшити час реакції системи.

Алгоритмічне вдосконалення:

- динамічне налаштування порогів HSV відповідно до рівня освітлення;
- додавання попереднього аналізу контексту з кількох кадрів (темпоральна фільтрація);
- розширення виявлення за допомогою легковагових нейронних мереж (наприклад, MobileNetV2 + SSD) для підвищення точності при складних умовах.

Апаратні покращення:

- заміна Raspberry Pi 4 на NVIDIA Jetson Nano або аналогічну платформу з апаратним прискоренням обчислень;
- використання MQ-135 або лазерного детектора диму для зменшення кількості хибних спрацювань;
- додавання інфрачервоної або тепловізійної камери для роботи в темряві або за складних погодних умов.

Надійність реалізованої системи була підтверджена в процесі багаторазового тестування в різних умовах. Вона стабільно працює протягом тривалих періодів без необхідності ручного перезапуску або втручання користувача. У випадках втрати сигналу з камери або сенсора система не виходить з ладу, а автоматично обробляє помилку, зберігаючи останній коректний стан та очікуючи на відновлення зв'язку. Це забезпечує безперервність роботи та дозволяє уникнути помилкових спрацювань у разі тимчасових збоїв у передачі даних. Завдяки простій, але ефективній логіці обробки подій, програмна частина працює з мінімальним навантаженням на систему і може безперервно функціонувати у фоновому режимі.

3.5.4 Надійність та масштабованість системи

Щодо масштабованості, архітектура системи спроектована таким чином, щоб легко адаптувати її до потреб більших об'єктів або кількох незалежних приміщень. Найпростішим прикладом масштабування є мультизональна система, коли до одного центрального контролера або до окремих одноплатних комп'ютерів підключаються декілька комплектів камер та сенсорів. Кожен з них може незалежно аналізувати дані у своїй зоні відповідальності – наприклад, окремі офіси в будівлі або відсіки складу. Такий підхід дозволяє точно локалізувати загрозу та оперативно реагувати лише в необхідній зоні, не викликаючи загальної тривоги.

Наступний рівень масштабування – централізований моніторинг. У цьому випадку вузли системи передають дані про стан кожної зони на центральний сервер або пульт керування. Централізований сервер об'єднує інформацію з усіх підсистем, виводячи її на єдину панель управління. Це особливо зручно для адміністраторів безпеки або диспетчерських центрів, де ведеться постійний моніторинг великих об'єктів, таких як школи, офісні комплекси, торговельні центри тощо.

Ще одним вагомим напрямом масштабування є інтеграція в існуючу IoT-інфраструктуру. Завдяки підтримці стандартних інтерфейсів (MQTT, REST API), система легко об'єднується з іншими елементами розумного середовища – такими як системи відеоспостереження, централізована сигналізація, вентиляція або автоматичне пожежогасіння. Це відкриває можливості для створення повністю автономних протипожежних рішень, де система не лише виявляє загоряння, а й автоматично виконує відповідні дії – наприклад, сповіщає персонал, активує сирену, або вмикає систему водяного гасіння.

Варто зазначити, що запропонована система є гнучкою щодо режимів розгортання. Вона може працювати як у повністю локальному режимі – без доступу до Інтернету, зберігаючи всі дані на локальному накопичувачі, так і у хмарному середовищі, коли декілька відеопотоків передаються на віддалений сервер для централізованого аналізу. Це дає змогу адаптувати систему до будь-

2025 р.

яких умов експлуатації – від маленьких об'єктів до складних інфраструктурних рішень на рівні міських або промислових об'єктів.

Система виявлення пожежі, розроблена у рамках даного проєкту, демонструє високу ефективність у тестових умовах. Вона є перспективною для розгортання у реальних об'єктах завдяки відкритій архітектурі, можливостям оптимізації та масштабування, а також стабільній роботі з обмеженими обчислювальними ресурсами.

Висновки до розділу 3

У даному розділі було реалізовано та досліджено повноцінну апаратно-програмну систему виявлення пожежі, що поєднує аналіз відеопотоку з IP-камери та дані з газового сенсора MQ-2. Було обґрунтовано вибір мови програмування Python, інструментів та бібліотек для комп'ютерного зору, а також детально описано структуру програмної частини системи, включаючи алгоритми обробки сигналів і прийняття рішень.

Реалізовано прототип моніторингової системи, яка здійснює автоматичне виявлення ознак пожежі на основі аналізу відеоінформації та даних з датчика диму. Проведене тестування в різних сценаріях підтвердило здатність системи виявляти ознаки займання з високою точністю та низькою кількістю хибних спрацювань. Найкращі результати були досягнуті при одночасному використанні відеоаналізу та сенсорних даних (сценарій 3), що дозволило скоротити час реагування до 1–2 секунд і зменшити кількість помилкових тривог. Оцінено надійність і масштабованість розробленого рішення, запропоновано шляхи подальшого розвитку: впровадження нейронних мереж, інтеграція з IoT-платформами, розширення до багатозональних систем.

Загалом розроблена система виявлення пожежі є ефективним та доступним рішенням для локального моніторингу приміщень. Вона поєднує програмну гнучкість із апаратною простотою, що робить її придатною для подальшої адаптації, розширення функціональності та впровадження у реальні об'єкти.

ВИСНОВКИ

У результаті виконання кваліфікаційної бакалаврської роботи досягнуто поставленої мети – розроблено апаратно-програмну систему для розпізнавання пожеж у приміщеннях, яка поєднує відеоаналіз полум'я з використанням ІР-камери та комп'ютерного зору, а також сенсорний контроль диму на основі газового датчика MQ-2. Система забезпечує раннє виявлення загоряння з високою точністю, коротким часом реагування (до 2 секунд) і низькою ймовірністю хибних спрацювань, що дозволяє ефективно використовувати її у житлових, офісних або виробничих приміщеннях.

У процесі виконання роботи було послідовно реалізовано всі поставлені завдання:

- проведено обґрунтування актуальності теми на тлі зростання потреби у доступних та надійних пожежних системах;
- здійснено аналіз сучасних методів виявлення загорянь, зокрема спектральних, тепловізійних та візуальних підходів, на основі чого обрано найбільш доцільні методи для реалізації в умовах обмежених ресурсів;
- обґрунтовано вибір апаратної платформи Raspberry Pi 4 та відповідних програмних інструментів;
- реалізовано прототип моніторингової системи, яка здійснює автоматичне виявлення ознак пожежі на основі аналізу відеоінформації та даних з датчика диму;
- проведено тестування системи в умовах, наближених до реальних, що підтвердило її стабільність, швидкодію та точність;
- проведено оцінку надійності та запропоновано шляхи подальшого розвитку.

На основі порівняння з існуючими аналогами визначено переваги запропонованого підходу – вища швидкодія, здатність до виявлення пожежі ще до появи значного диму, гнучкість у налаштуванні системи, відкритість архітектури та підтримка автономної роботи без потреби у сервері. Комбінація

візуального аналізу (HSV-маска, Haar-каскад) та сенсорної детекції (MQ-2 + MCP3008) дозволила значно підвищити надійність і зменшити кількість хибних спрацювань. Функціональність системи наближається до комерційних рішень, однак реалізована вона на значно простішій і дешевшій платформі.

Практичне значення розробки полягає в можливості впровадження в реальних об'єктах – від приватних помешкань до об'єктів соціальної інфраструктури. Завдяки підтримці сучасних протоколів зв'язку, систему можна інтегрувати в розумні будівлі або платформи IoT. Вона підходить для автономного використання або як доповнення до існуючих пожежних сповіщувачів. Прогнозується, що подальший розвиток розробки може включати використання нейронних мереж для точнішого розпізнавання складних ситуацій, автоматичне керування засобами гасіння, хмарне збереження даних, а також масштабування на рівень міського або промислового моніторингу.

Таким чином, усі задачі, були реалізовані в повному обсязі. Система виявлення пожеж, створена в межах даної роботи, є ефективним, масштабованим і практичним рішенням, що може бути основою для подальших досліджень і комерційного впровадження у сфері техногенної безпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. АНАЛІТИЧНА ДОВІДКА про пожежі та їх наслідки в Україні за 8 місяців 2024 року. URL: <https://dsns.gov.ua/upload/2/2/8/5/6/0/4/analitichna-dovidka-pro-pojeji-082024.pdf> (дата звернення: 03.05.2025).
2. CTIF center for world fire statistics. *CTIF Center for World Fire Statistics*. URL: <https://ctif.org/commissions-and-groups/ctif-center-world-fire-statistics> (Last accessed: 04.05.2025).
3. Li M.X., Xu W.J., Zhang Y., Hou J. Fire detection technologies based on video image / M. X. Li et al. *Advanced materials research*. 2013. Vol. 659. P. 134–138. DOI: 10.4028/www.scientific.net/AMR.659.134.
4. Sulthana S. F., Wise C. T. A., Ravikumar C. V., Anbazhagan R., Idayachandran G., Pau G. Review Study on Recent Developments in Fire Sensing Methods, *IEEE Access.*, 2023. Vol. 11. P. 90269–90282. DOI: 10.1109/ACCESS.2023.3306812.
5. Haar A. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*. 1910. Vol. 69. no. 3. P. 331–371. DOI: 10.1007/bf01456326.
6. Raspberry pi 4. *Raspberry Pi*. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (Last accessed: 05.05.2025).
7. OpenCV. URL: <https://opencv.org/> (Last accessed: 06.05.2025).
8. 10-розрядний АЦП MCP3008 для raspberry PI та arduino. *Arduino*. URL: <https://arduino.ua/prod1931-10-razryadnii-acp-mcp3008-dlya-raspberry-pi-i-arduino?srsId=AfmBOooF4Tv3gYDpZjrIbz0-dX36G6FIkOjSdSf89NS6dafFrTEIf4i2> (дата звернення: 08.05.2025).
9. Модуль датчика диму MQ-2. *Arduino*. URL: <https://arduino.ua/prod298-modyl-datchika-dima-mq-2> (дата звернення: 06.05.2025).
10. QA | IEC 61508 – cross-industry safety standard. URL: <https://www.qt.io/quality-assurance/iec-61508> (Last accessed: 07.05.2025).

11. Юрченко О. В., Бурлаченко І. С. Мультиагентна моніторингова система розпізнавання пожеж у приміщеннях. *Могилянські читання – 2024* : тези доп. XXVII Всеукр. наук.-практ. конф., Миколаїв, 6–10 листоп. 2024 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2024. С. 155–158. URL: <https://dspace.chmnu.edu.ua/jspui/handle/123456789/2507> (дата звернення: 27.04.2025).
12. Apollo conventional UV / dual IR flame detector – 55000-064APO. *Acorn Fire Security*. URL: <https://www.acornfiresecurity.com/apollo-conventional-uv-dual-ir-flame-detector-55000-064apo> (Last accessed: 12.05.2025).
13. Optical flame detection. *Intelli-tech*. URL: <https://www.gotoitsi.com/optical-flame-detection/> (Last accessed: 12.05.2025).
14. Дротовий пожежний сповіщувач arton СПД-3.0. URL: <https://teren.ua/ua/viatec-spd-30-provodnoj-dymovoj-pozharnyj-izveshatel/> (дата звернення: 11.05.2025).
15. Gopinath S., et al. Forest fire detection using machine learning with raspberry PI. *International journal of advanced research in science, communication and technology*. 2022. P. 683–691. DOI: 10.48175/IJARSCT-3837.
16. Kim Y.-H., Kim A., Jeong H.-Y. RGB color model based the fire detection algorithm in video sequences on wireless sensor network. *International journal of distributed sensor networks*. 2014. Vol. 10, no. 4. P. 923609. DOI: 10.1155/2014/923609.
17. Srujan H. R., Srushti S. M., V. Sai Hitesh Gowda, Spoorthi G., Sudhakara H. M. Fire and smoke detector alarm using arduino and MQ2 sensor . *International journal of advanced research in science, communication and technology*. 2025. Vol. 5, no. 1. DOI: 10.48175/568.
18. Breuss M. Python virtual environments: a primer – real python. *Python Tutorials – Real Python*. URL: <https://realpython.com/python-virtual-environments-a-primer/> (Last accessed: 20.05.2025).

19. Control raspberry pi GPIO pins from python. ICS - Integrated Computer Solutions. URL: <https://www.ics.com/blog/control-raspberry-pi-gpio-pins-python> (Last accessed: 02.06.2025).
20. Python NumPy – GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/python-numpy/> (Last accessed: 02.06.2025).
21. Serial peripheral interface (SPI). Introduction to Raspberry Pi. URL: <https://raspberrypi-aa.github.io/session3/spi.html> (Last accessed: 02.06.2025).
22. Thonny, Python IDE for beginners. URL: <https://thonny.org/> (Last accessed: 02.06.2025).

ДОДАТОК А

Код програми розпізнавання пожеж у приміщеннях

```
import cv2
import numpy as np
import datetime
import spidev
import time
from collections import deque

# --- SPI and MQ-2 functions ---
def init_spi():
    spi = spidev.SpiDev()
    spi.open(0, 0)
    spi.max_speed_hz = 100000
    return spi

def read_channel(spi, channel):
    adc = spi.xfer2([1, (8 + channel) << 4, 0])
    return ((adc[1] & 0x03) << 8) | adc[2]

def estimate_ppm(raw_value):
    return (raw_value / 1023.0) * 1000 # Simplified ppm estimate

def update_smoke_level(spi, buffer, maxlen, smoke_counter):
    raw = read_channel(spi, 7)
    ppm = estimate_ppm(raw)
    buffer.append(ppm)
    avg_ppm = sum(buffer) / len(buffer)
    F2 = min(avg_ppm / 1000, 1.0)
    if avg_ppm > 300:
        smoke_counter += 1
    else:
        smoke_counter = max(smoke_counter - 1, 0)
    return F2, avg_ppm, smoke_counter

# --- Fire detection functions ---
def detect_color_mask(frame):
    blurred = cv2.GaussianBlur(frame, (7, 7), 0)
    hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)
    lower_fire = np.array([10, 100, 100])
    upper_fire = np.array([25, 255, 255])
    mask = cv2.inRange(hsv, lower_fire, upper_fire)
    kernel = np.ones((5, 5), np.uint8)
    mask = cv2.erode(mask, kernel, iterations=1)
    mask = cv2.dilate(mask, kernel, iterations=2)
    return mask

def detect_flicker(gray, prev_gray):
    if prev_gray is None:
        return np.zeros_like(gray)
    diff = cv2.absdiff(gray, prev_gray)
    _, flicker = cv2.threshold(diff, 25, 255, cv2.THRESH_BINARY)
    kernel = np.ones((5, 5), np.uint8)
    flicker = cv2.dilate(flicker, kernel, iterations=2)
    return flicker
```

```
def analyze_flames(frame, contours, flicker_mask):
    flame_pixels = 0
    for cnt in contours:
        area = cv2.contourArea(cnt)
        if area > 300:
            x, y, w, h = cv2.boundingRect(cnt)
            region = flicker_mask[y:y+h, x:x+w]
            if cv2.countNonZero(region) > 50:
                flame_pixels += int(area)
                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
                cv2.putText(frame, "Flame", (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
    return flame_pixels

def analyze_cascade(frame, gray, cascade):
    flame_pixels = 0
    regions = cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)
    for (x, y, w, h) in regions:
        flame_pixels += w * h
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
        cv2.putText(frame, "Cascade", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.7, (255, 0, 0), 2)
    return flame_pixels

def should_trigger_alarm(P, ratio, smoke_counter):
    return P > 0.8 or ratio > 0.3 or smoke_counter > 10

# --- Main processing loop ---
def main():
    spi = init_spi()
    ppm_buffer = deque(maxlen=10)
    smoke_counter = 0
    fire_cascade = cv2.CascadeClassifier('fire_cascade.xml')
    cap =
cv2.VideoCapture("rtsp://user:%21Fire405@192.168.0.20:554/media/stream.sdp?profi
le=Profile100")

    print("MQ-2 warming up... wait 20 sec")
    time.sleep(20)

    prev_frame = None

    while True:
        ret, frame = cap.read()
        if not ret:
            print("Frame not read. Exiting.")
            break

        # --- Sensor processing ---
        F2, avg_ppm, smoke_counter = update_smoke_level(spi, ppm_buffer, 10,
smoke_counter)

        # --- Image preprocessing ---
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        mask_color = detect_color_mask(frame)
        contours, _ = cv2.findContours(mask_color, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        flicker_mask = detect_flicker(gray, prev_frame)
        prev_frame = gray
```

```
# --- Fire pixel estimation ---
flame_pixels = analyze_flames(frame, contours, flicker_mask)
flame_pixels += analyze_cascade(frame, gray, fire_cascade)

frame_area = frame.shape[0] * frame.shape[1]
F1 = 1.0 if flame_pixels > 300 else 0.0
fire_area_ratio = flame_pixels / frame_area
P = 0.7 * F1 + 0.3 * F2

# --- Alarm logic ---
alarm = should_trigger_alarm(P, fire_area_ratio, smoke_counter)
if alarm:
    print(f"[{datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')}]
ALERT! P={P:.2f}, F1={F1:.2f}, F2={F2:.2f}, PPM={avg_ppm:.1f}")
    cv2.putText(frame, "ALERT!!!", (20, 40), cv2.FONT_HERSHEY_SIMPLEX,
1.0, (0, 0, 255), 3)

# --- Display frame ---
cv2.imshow("Fire detection", frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()
spi.close()

# --- Entry point ---
if __name__ == "__main__":
    main()
```

ДОДАТОК Б

Матеріали апробації роботи

XXVII Всеукраїнська науково-практична конференція «Могилянські читання – 2024»

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
ДНУ «Інститут модернізації змісту освіти»
Південний науковий центр НАН та МОН
Інститут української археографії та джерелознавства
імені М. С. Грушевського НАН України
Первинна профспілкова організація ЧНУ ім. Петра Могили



«МОГИЛЯНСЬКІ ЧИТАННЯ – 2024:
досвід та тенденції розвитку суспільства в Україні:
глобальний, національний та регіональний аспекти»

XXVII Всеукраїнська науково-практична конференція

ТЕЗИ ДОПОВІДЕЙ

ТЕХНІЧНІ НАУКИ

Миколаїв, 6–10 листопада 2024 року

Миколаїв – 2024

<i>Старченко В. В.</i> Контролер для систем віртуальної реальності на базі модуля DFRobot Firebeetle Covers-OLED12864 Display.....	133
<i>Тогося О. Р.</i> Використання Postman для сніфінгу вебтрафіку.....	137
<i>Ухань Є. О.</i> Модель Wi-Fi мережі на базі технології 802.11ad.....	140
<i>Хмельовський В. Р., Чешун Д. В., Чешун В. М.</i> Застосування технології NFC для забезпечення безпечної синхронізації даних.....	144
<i>Чернявський Р. А., Крайник Я. М.</i> Використання великих мовних моделей для підвищення ефективності інтерактивних систем: проблеми та перспективи.....	149
<i>Чуйко Г. П., Гончаров Д. С., Бушук М. С.</i> Відбір атрибутів та скорочення розмірності даних щодо раку молочної залози з Knowledge Flow.....	152
<i>Юрченко О. В., Бурлаченко І. С.</i> Мультиагентна моніторингова система розпізнавання пожеж у приміщеннях.....	155

Підсекція:

➤ ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ СИСТЕМИ

<i>Андріюк Р. А., Обухова К. О.</i> Розробка інформаційної системи для тенісного клубу на базі фреймворка React.....	159
<i>Болюбаш Н. М., Фінькова О. В.</i> Система аналізу даних для прогнозного технічного обслуговування технологічних ліній.....	163
<i>Брагінець О. В., Магда О. В.</i> Точні розв'язки узагальнених рівнянь Кортвега-де Фріза зі змінними коефіцієнтами.....	168
<i>Воробйова А. І.</i> Британські математичні освітні та наукові мережі.....	173
<i>Донченко М. В.</i> Вплив обмеженої кількості запасних частин на надійність відновлюваних систем.....	177
<i>Гожий О. П., Калініна І. О.</i> Методологія вирішення завдань класифікації в задачах машинного навчання.....	179
<i>Данкович С. Ю., Сіденко Є. В.</i> Система ідентифікації та розпізнавання дронів методами штучного інтелекту.....	185
<i>Калініна І. О., Гожий В. О.</i> Метод моделювання та створення композитних web-сервісів на основі кольорових мереж Петрі.....	188