

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ**

На правах рукопису
УДК 004.852

До захисту допущено
В.о. завідувача кафедри ШІ
_____ О.І. Чумаченко
« ____ » _____ 2022 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 122 «Комп'ютерні науки»

**на тему: «Розпізнавання пожежі за допомогою згорткових нейронних
мереж»**

Виконав:
студент II курсу, групи КІ-з11мп
Маркін Іван Дмитрович _____

Керівник:
Старший викладач кафедри ШІ,
к.т.н., Шаповал Н. В. _____

Рецензент:
доцент кафедри системного проектування
КПІ ім. Ігоря Сікорського, к.ф.-м.н., Шубенкова І. А. _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ
2022

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ О.І. Чумаченко

«___» _____ 2022 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Маркіну Івану Дмитровичу

1. Тема дисертації: «Розпізнавання пожежі за допомогою згорткових нейронних мереж», науковий керівник роботи Шаповал Наталія Віталіївна, старший викладач кафедри ММСА, к.т.н., затверджено наказом по університету від «02» листопада 2022 р. № 4040-с.
2. Термін подання студентом дисертації 12.12.2022р.
3. Об'єкт дослідження: Задача розпізнавання пожежі.
4. Предмет дослідження: Згорткові нейронні мережі для розпізнавання пожежі.
5. Перелік завдань, які потрібно зробити:
 - 1) здійснити огляд технічної літератури за темою роботи;
 - 2) дослідити актуальність обраної теми;
 - 3) ознайомитись із існуючими методами розпізнавання пожеж;

- 4) розробити та реалізувати систему, що використовує апарат нейронних мереж, та вирішує задачу розпізнавання пожежі;
 - 5) провести експеримент, що засвідчує працездатність запропонованої моделі, виконати аналіз результатів;
 - 6) провести аналіз ринкових можливостей запуску стартап проекту;
 - 7) розробити концептуальні висновки;
 - 8) підготувати ілюстративний матеріал;
 - 9) оформити пояснювальну записку.
6. Перелік ілюстративного матеріалу.
7. Дата видачі завдання: 1 вересня 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за темою роботи.	02.09.2022 – 14.09.2022	Виконано
2.	Підготовка першого розділу.	15.09.2022 – 29.09.2022	Виконано
3.	Підготовка другого розділу.	30.09.2022 – 18.10.2022	Виконано
4.	Розробка програмного продукту.	20.10.2022 – 05.11.2022	Виконано
5.	Підготовка третього розділу	06.11.2022 – 18.11.2022	Виконано
6.	Підготовка частини стартап-проєкту	19.11.2022 – 22.11.2022	Виконано
9.	Концептуальні висновки. Перспективи розвитку отриманих рішень	23.11.2022 – 25.11.2022	Виконано
10.	Оформлення пояснювальної записки	26.11.2022 – 03.12.2022	Виконано

Студент

Іван МАРКІН

Керівник

Наталія

ШАПОВАЛ

РЕФЕРАТ

Магістерська дисертація: 114 с., 25 табл., 10 рис., 19 джерел, 1 додаток.

ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ЛОКАЛІЗАЦІЯ GRAD-CAM

Об'єктом дослідження є задача розпізнавання об'єктів на наборі зображень.

Предмет дослідження – методи обробки зображень та згорткові нейронні мережі для класифікації зображень.

Мета дослідження полягає у аналізі алгоритмів класифікації зображень, що базуються на моделях згорткових нейронних мереж, а також алгоритмів, локалізації, розробка власної нейронної мережі.

Розроблена згорткова нейронна мережа для вирішення задачі розпізнавання пожежі. Для локалізації пожежі на зображенні використано алгоритм Grad-CAM.

Проведено порівняння запропонованої нейронної мережі на вибірці зображень з попередньою локалізацією та без.

ABSTRACT

Master's thesis: 114 p., 25 tab., 10 fig., 19 references, 1 appendix.

CONVOLUTIONAL NEURAL NETWORK, IMAGE CLASSIFICATION, GRAD-CAM LOCALIZATION

The object of research is the task of object recognition on a set of images.

The subject of research is image processing methods and convolutional neural networks for image classification.

The purpose of the research is to analyze image classification algorithms based on convolutional neural network models, as well as algorithms, localization, and development of one's own neural network.

A convolutional neural network was developed to solve the problem of fire recognition. The Grad-CAM algorithm was used to localize the fire in the image.

The proposed neural network was compared on a sample of images with and without prior localization.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ЗАДАЧІ РОЗПІЗНАВАННЯ ПОЖЕЖІ	9
1.1. Аналіз актуальності задачі розпізнавання пожежі	9
1.2. Аналіз існуючих методів розпізнавання пожеж	11
1.2.1. Wireless sensor network	11
1.2.2. «FireWatch»	14
1.3. Висновки до розділу	19
РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ АЛГОРИТМУ РОЗПІЗНАВАННЯ ПОЖЕЖІ	20
2.1 Згорткові нейронні мережі	20
2.2 Grad-CAM алгоритм	26
2.3 Висновки до розділу	30
РОЗДІЛ 3 ПОБУДОВА ВЛАСНОЇ CNN ТА АНАЛІЗ РЕЗУЛЬТАТІВ	31
3.1 Вибір та підготовка даних	31
3.2 Побудова CNN	32
3.3 Алгоритм Grad-CAM	32
3.4 Аналіз та порівняння результатів	33
3.5 Висновки до розділу	38
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ	39
4.1 Опис ідеї проекту	39
4.2 Технологічний аудит ідеї проекту	41
4.3 Аналіз ринкової стратегії проекту	50
4.4 Розроблення маркетингової програми стартап-проекту	53

4.5	Висновки	57
	ВИСНОВКИ	58
	ПЕРЕЛІК ПОСИЛАНЬ	59
	ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО КОДУ	60

ВСТУП

Виявлення пожежі на ранній стадії до її переростання в катастрофічну подію має вирішальне значення для нейтралізації пожеж та порятунку життя та майна людей. Через швидке поширення пожежі важливо виявити його на стадії перших проявів вогню.

Незважаючи на важливість цієї проблеми, вона залишається невирішеною на даний момент. Датчики, основним принципом яких є виявлення диму, добре працюють, коли пожежа триває протягом певного часу і утворилася достатня кількість диму для спрацьовування сигналізації. Ці пристрої не можуть бути використані у великих масштабах на відкритому повітрі, наприклад, у лісі. Супутникові системи, стаціонарні віддалені камери та пілотовані літаки - це звичайні технології дистанційного моніторингу пожеж, проте ці технології мають низку обмежень з точки зору вартості, тимчасового та просторового дозволу.

Також для вирішення поставленої задачі пропонується використання згорткових нейронних мереж. На даний момент, технології нейронних мереж досягли високого рівня розвитку і вже дозволяють знаходити рішення специфічним завданням, працюючи за принципом, схожим на процес прийняття рішення людиною. Використання таких методів на даному етапі розвитку технологій не може зробити прийняття рішень повністю автоматизованим, проте воно суттєво знижує вплив людського фактора та забезпечує високу результативність за менших витрат часу та зусиль.

Метою даної роботи є розробка програмного забезпечення для виявлення лісової пожежі за даними аерофотозйомки за допомогою нейронних мереж.

РОЗДІЛ 1 АНАЛІЗ ЗАДАЧІ РОЗПІЗНАВАННЯ ПОЖЕЖІ

1.1. Аналіз актуальності задачі розпізнавання пожежі

Пожежа – це неконтрольоване горіння, що знищує або ушкоджує матеріальні цінності, що створює небезпеку для життя людей . У будь-якій ситуації навіть невелике займання може перерости у стихійне лихо. В даний час ймовірність спалаху та масштабного поширення вогню через природні фактори не перевищує 20%. Більшість лісових пожеж спровоковано діяльністю громадян [1]. Найбільш поширені причини виникнення пожежі у лісі:

1. безвідповідальна поведінка людей;
2. дитяча витівка з вогнем;
3. спалювання сміття поблизу житлових будинків та на території,
4. прилеглої до лісових масивів;
5. удар блискавки;
6. підпал.

Інфрачервоне випромінювання – електромагнітне випромінювання, яке займає область на спектрі між червоним світлом та короткохвильовим радіовипромінювання. Вперше воно було виявлено у 1800 році астрономом Вільямом Гершелем. Його іноді називають «тепловим випромінюванням», оскільки інфрачервоне випромінювання від нагрітих тіл сприймається шкірою людини як відчуття тепла. Таке випромінювання випускають усі тіла, температура яких відрізняється від нуля. Інфрачервоне випромінювання широко використовується в промисловості, наукових дослідженнях та медицині. У цій галузі, існує величезне кількість розробок, заснованих на електромагнітному випромінюванні, наприклад: прилади для спостереження в темряві, прилади для дистанційного вимірювання температури тіл і прилади для виявлення цілей щодо їхнього власного теплового

інфрачервоного випромінювання (термографія). Ці прилади перетворюють зовнішній фотон світла на електрони, які підсилюють хімічні та електричні процеси і перетворюються назад на видиме джерело світло.

Використання даних у тепловому інфрачервоному фільтрі дозволяє уникнути роботи із задимленням та збільшити точність роботи алгоритму розпізнавання, знизивши відсоток похибки.

Виявлення пожежі на ранній стадії до її переростання в катастрофічна подія має вирішальне значення для нейтралізації пожеж та порятунку життя та майна людей. Через швидке поширення пожежі, важливо виявити його на стадії перших проявів вогню. Датчикам, які виявляють присутність частинок, що утворюються в результаті диму та вогню, шляхом іонізації, потрібна безпосередня близькість до вогню. Ці пристрої не можуть бути застосовані у великих масштабах на відкритому повітрі, наприклад, у лісі. Більше того, вони не можуть надати інформацію про початкове місцезнаходження пожежі, напрямі поширення диму, розмірі, швидкості зростання пожежі і т.д. Супутникові системи, стаціонарні віддалені камери та пілотовані літаки - це звичайні технології дистанційного моніторингу пожеж для надання інформаційних даних про стихійне лихо. Однак ці технології мають низку обмежень з точки зору вартості, тимчасового та просторового дозволу. Для подолання цих обмежень та розширення можливостей традиційних систем, є перспективним використання можливостей безпілотних літальних апаратів (далі БПЛА). В останні роки, з постійним розвитком технологій, БПЛА стали застосовуватися в багатьох областях, таких як метеорологічні дослідження, моніторинг стихійних лих, інспекція ліній електропередач та рятувальних операцій. Зокрема, легкі та малогабаритні БПЛА відрізняються низькою вартістю, простотою в експлуатації та гнучкістю маневрування. У порівнянні з тепловізійними системами, інфрачервоними та багатоспектральними датчиками, безпілотні літальні апарати, оснащені звичайними камерами для

дослідження пожеж, мають ряд переваг – низька ціна, простота експлуатації тощо [1].

1.2. Аналіз існуючих методів розпізнавання пожеж

1.2.1. Wireless sensor network

Виявлення лісових пожеж було в центрі уваги багатьох дослідників протягом останнього десятиліття через збільшення кількості повідомлень про випадки лісових пожеж з усього світу, через серйозну шкоду, завдану суспільству та навколишньому середовищу. Було запропоновано багато методів виявлення лісових пожеж, таких як системи на основі камер, системи на основі wireless sensor network (далі – WSN) і додатків на основі машинного навчання, як з позитивними так і з негативними перспективами та показниками ефективності. Завдяки вищій ймовірності точного та раннього виявлення, завдяки використанню кількох джерел датчиків і розгортанню вузлів датчиків у зонах, невидимих для супутників, бездротові сенсорні мережі мають більші перспективи, і вони стали технологією для широкого використання.

Багато дослідників зосереджувалися на параметрах навколишнього середовища, таких як температура повітря, відносна вологість, тиск, звук, інтенсивність світла, вологість ґрунту, швидкість і напрям вітру, а також гази, такі як CO, CO₂, метан, H₂ та вуглеводні, які виділяються при задимленні, щоб з'ясувати умови для виникнення лісової пожежі. Враховуючи можливі коливання цих параметрів під час пожежі були обрані датчики відповідно до діапазону, чутливості, енергоспоживання та вартості.

Оскільки живлення сенсорного вузла є складним завданням у лісистих районах, використовувати лише акумулятори важко, оскільки вони не

працюють довго, а розподіл живлення за допомогою дроту вимагатиме вищих витрат для розгортання у великому лісі. Тому багато дослідників пропонували сонячні системи як вторинні джерела живлення разом з акумуляторними батареями як основним джерелом енергії, тоді як деякі дослідники запропонували сонячні батареї, оскільки вони служать довше. Щоб зменшити енергоспоживання сенсорних вузлів, були запропоновані такі методи, як утримання вибраних компонентів активними, а інші дезактивованими.

Більшість систем виявлення на основі WSN зосереджені навколо базової станції через обмеження пам'яті та обробки вузлів. Важливі та частково оброблені дані передаються на базову станцію через бездротове середовище для обробки та забезпечення відповідних дій, тоді як базова станція також діє як шлюз між вузлами з датчиками та користувачем системи.

Під час побудови WSN головною метою є обмін даними між відповідними об'єктами, а зіркоподібна топологія та мережі на основі сітчастої топології були запропоновані в багатьох роботах завдяки різним атрибутам в їхніх системах. Сітчаста топологія була обрана замість зіркоподібної через її здатність до самоорганізації, самоналаштування та автоматичного визначення себе між вузлами в мережі. Оскільки менша кількість вузлів, залучених до передачі, призводить до меншого споживання енергії, були використані концепції, засновані на центрах кластерів. Для мінімізації втрат енергії та пакетів даних під час передачі, була запропонована структура мережевої топології дерева кластерів. Враховуючи зону покриття вузла, відмовостійкість і енергоспоживання, запропоновано застосувати техніку k -покриття, яка забезпечує виявлення подій за допомогою статичних вузлів зі змінними діапазонами покриття. Цей метод використовує максимальну ефективність виявлення з мінімальним енергоспоживанням. В дослідженнях щодо виявлення рідкісних подій

згадується багато стратегій їх виявлення, які забезпечують максимальні можливості виявлення, мінімізовану затримку виявлення та низьке енергоспоживання, наприклад робочий цикл, дезактивація компонентів, перенаселення / надлишковість вузлів, співпраця та збір енергії.

Щоб зменшити витрати на розгортання та енергоспоживання пропонується нова схема локалізації, яка розділяє всю лісову територію на різні сітки та розподіляє їх на відповідні зони з ще вісьмома сусідніми сітками. Один вузол центра з цих сіток, який називається вузлом-ініціатором, визначає чи є зона з високою активністю (HA), середньою активністю (MA) і низькою активністю (LA). Зони HA безперервно надсилають дані до базового вузла через внутрішній вузол, зони MA надсилають дані періодично, а зони LA не передають дані та перебувають у стані ефективного споживання енергії. Інші дослідження пропонують отримувати дані від датчиків кожні 2 хвилини, якщо є потенційна лісова пожежа, або отримувати дані кожні 15 хвилин в іншому випадку, щоб зменшити втрати енергії.

Щоб розмістити вузли датчиків у найбільш ефективній конфігурації для виявлення умов пожежі, було запропоновано вузли датчиків на трьох різних висотах для виконання різних вимірювань, також запропоновано накрити вузли датчиків, щоб уникнути впливу прямого сонячного світла та мінімізувати частоту помилкових тривог. Оскільки підключення до мережі постачальників послуг у лісових районах не є надійним, методи зв'язку, які використовують виділені мережеві шляхи, такі як LoRa, ZigBee та XBee, використовувалися як комунікаційна інфраструктура. Розглядаючи такі атрибути, як дальність передачі, високий рівень безпеки, низьке енергоспоживання (протокол LPWAN) та інші відповідні конфігурації, у більшості документів пропонується використовувати модуль LoRa для передачі даних.

Більшість досліджень пропонує виявлення пожежі на основі порогових значень на вузлі датчика, і якщо перевищений поріг залишається незмінним,

то вузол визначає місце розташування та надсилає сигнал тривоги у пожежну службу. Оскільки параметри навколишнього середовища змінюються залежно від місця та часу, порогові значення налаштовуються користувачем з урахуванням географічної ситуації, кліматичних змін, сезонних змін тощо після того, як датчики отримують дані з навколишнього середовища.

Було запропоновано процес злиття інформації, де інформація з кількох джерел враховується для прийняття остаточного рішення, що краще, ніж використання цих джерел окремо, і використовувалися два алгоритми, засновані на методі порогового співвідношення та теорії Демпстера-Шейфера. Щоб підвищити точність виявлення, у багатьох статтях були запропоновані програми машинного навчання на основі різних підходів до машинного навчання, таких як класифікація опорних векторних машин (SVM) і методи регресії, такі як логістична регресія. Однак застосування методів машинного навчання до систем виявлення пожежі має багато обмежень, таких як обмежена кількість енергії, енергія, необхідна для обробки даних, короткий діапазон зв'язку та обмежені обчислення, складність алгоритмів ML під час виконання на сенсорних вузлах та складність розподілу на кожному сенсорному вузлі [2].

1.2.2«FireWatch»

Автоматизована система виявлення лісових пожеж "FireWatch". Система працює шляхом розпізнавання хмар диму. Кожні 6 хвилин сенсор, який зображений на рисунку 2.1, встановлений на висоті 10 метрів над лісовим покривом, обертається на 360 градусів і фіксує наявність диму. У разі виявлення ознак задимлення, сигнал тривоги відправляється через

бездротове з'єднання до центру управління, де оператор приймає рішення про подальші дії. Принцип роботи системи FireWatch зображено на рисунку 2.2.



Рис. 2.1 - Сенсора системи FireWatch

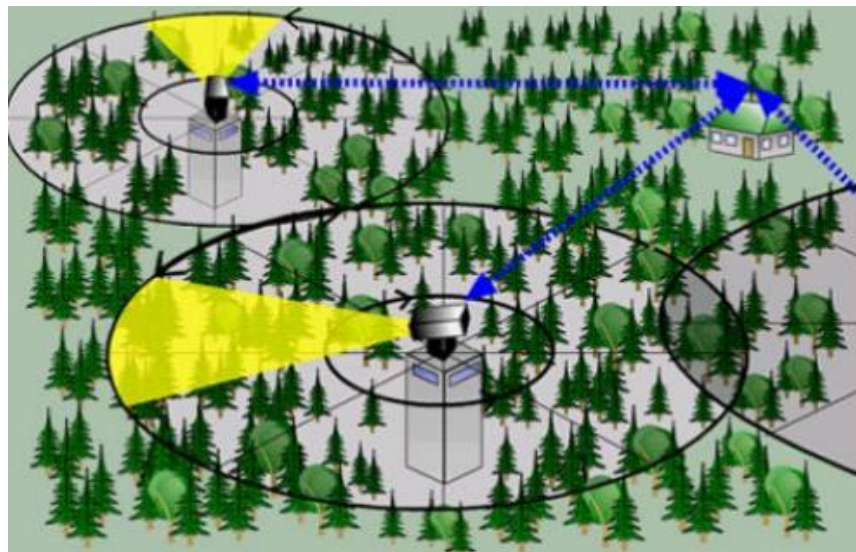


Рис. 2.2 - Принцип роботи системи FireWatch

FireWatch виявляє дим та схожі на дим явища за допомогою мультиспектрального датчика, який включає три-чотири оптичних датчика:

- Монохромний сенсор для денного спостереження.
- Датчик ближнього інфрачервоного діапазону для нічного виявлення.
- Датчик RGB для додаткової спектральної інформації.
- Тепловий інфрачервоний датчик для виявлення гарячих точок.

Використовуючи кілька датчиків, система поєднує в собі переваги оптичного виявлення диму, яке зарекомендувала себе за багато років, з перевагами теплового виявлення та уникає їхніх недоліків залежно від застосування. Таким чином, система може бути оптимально адаптована до всіх вимог.

Основними недоліками цієї системи є: висока вартість установки, обмеженість огляду (наявність сліпих зон) та сильна залежність результату роботи від погодних умов. На деяких територіях нерідким явищем є туман, це означає, що система працюватиме некоректно.

1.2.3. Forest Fire Detection Using Artificial Neural Network Algorithm Implemented in Wireless Sensor Networks [3].

Юншен Лю, Янсун Ян, Чанг Лю та Юй Гу з Дослідницького інституту мережевих технологій у Китаї запропонували систему виявлення лісових пожеж, засновану на штучній нейронній мережі та реалізовану за допомогою бездротової сенсорної мережі (Wireless Sensor Networks або WSN). Вчені вважають, що використання системи, яка залежить лише від одного критерію лісової пожежі, збільшує ймовірність помилкових спрацьовувань, тому їхній алгоритм враховує кілька атрибутів. Хоча траплялися й хибні тривоги. Головною причиною їх виникнення була чутливість фотоелектричного блоку виявлення диму до білих частинок від тліючого вогню та пилу.

Модуль виявлення пожежі відповідає за прийняття рішення та надсилає сигнал тривоги. Він містить п'ять сенсорних вузлів, чотири з яких збирають дані про можливе виникнення пожежі. Нейронна мережа реалізована на кожному окремому сенсорному вузлі. П'ятий вузол датчика

працює, як базова станція і збирає дані зондування та пожежну тривогу від інших чотирьох вузлів за допомогою бездротового зв'язку.

Для того, щоб безперервно використовувати систему датчиків було ухвалено рішення про встановлення сонячних батарей тому що це самий дешевий та екологічний варіант.

Оскільки в лісі доступне лише переривчасте сонячне світло, вченими було розроблено спеціальну модель сонячної батареї. Енергія сонячної панелі накопичується суперконденсатором.

Модуль інтерфейсу користувача відповідає за відображення даних, зібраних з датчиків. Дані зондування та пожежна сигналізація передаються від базової станції до користувача через Інтернет, тому користувач може знаходитися далеко від системи пожежогасіння.

1.2.4. Image-Based Fire Detection Using Neural Networks

Wen-Bing Horng та Jian-Wen Peng з Тайванського університету (факультет комп'ютерних наук та інженерії) розробили алгоритм виявлення вогню на основі зображень з використанням нейронних мереж зворотного розповсюдження [4]. Цей метод складається з трьох етапів: сегментація моделі кольорового полум'я, видалення несправжніх областей, які схожі на пожежу та оцінка пожежонебезпеки. На першому етапі три компонента HSI моделі (тон, насиченість та яскравість) кожного пікселя подаються на навчання нейронній мережі. Якщо піксель не відноситься до виявленої області полум'я, він зафарбовується чорним в інакшому разі колір не змінюється. Приклади цього етапу наведено на рисунку 2.3. Помилковими полями може бути відображення полум'я та об'єкти із схожими кольорами. У першій ситуації область можна відфільтрувати шляхом видалення пікселів із

найменшою інтенсивністю. У другому випадку враховується колір фону об'єкта. Якщо він аналогічний чи дуже схожий, то метод різниці видалить об'єкт із зображення.



Рис. 2.3 - Зображення в оригіналі та після обробки нейронною мережею

1.3.Висновки до розділу

У першому розділі було описано актуальність задачі розпізнавання пожежі. Також в першому розділі описувались вже існуючі методи розпізнавання пожежі та вже існуючі системи для вирішення даної задачі. У другому розділі будуть описані математичні основи для запропонованого рішення задачі розпізнавання пожежі.

РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ АЛГОРИТМУ РОЗПІЗНАВАННЯ ПОЖЕЖІ

2.1 Згорткові нейронні мережі

Згорткова нейронна мережа (Convolutional Neural Network — ConvNet/CNN) — це Deep Learning-алгоритм, який може приймати вхідне зображення, надавати важливість (засвоювані ваги та зміщення) різним областям/об'єктам у зображенні і може відрізнити одне від одного. Попередньої обробки ConvNet потрібно значно менше в порівнянні з іншими алгоритмами класифікації. У той час, як у примітивних методах фільтри сконструйовані вручну, ConvNets при достатньому навчанні здатні вивчати ці фільтри/характеристики [5].

Архітектура ConvNet схожа з архітектурою зв'язності нейронів у людському мозку і була натхненна організацією зорової кори. Окремі нейрони реагують на подразники лише у обмеженій зоні поля зору, відомої як рецептивне поле. Сукупність таких полів накладається щоб покрити всю зону поля зору.

Згорткові нейронні мережі працюють на основі фільтрів [6], які займаються розпізнаванням певних характеристик зображення (наприклад, прямих ліній). Фільтр – це колекція ядер, іноді у фільтрі використовується одне ядро. Ядро - це звичайна матриця чисел, званих вагами, які "навчаються" з метою пошуку на зображеннях певних характеристик. Фільтр переміщається вздовж зображення і визначає, чи є певна потрібна характеристика в конкретній його частині. Для отримання відповіді такого роду відбувається операція згортки, яка є сумою творів елементів фільтра та матриці вхідних сигналів.

ω_{11}	ω_{12}	ω_{13}
ω_{21}	ω_{22}	ω_{23}
ω_{31}	ω_{32}	ω_{33}

+1 • ω_0

bias

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	

Рисунок 2.4 – ядро фільтру

Ці числа помножуються на значення відповідних пікселів зображення та підсумовуються:

$$v_{0,0} = \sum_{i=1}^3 \sum_{j=1}^3 x_{i,j} \cdot \omega_{ij} + \omega_0$$

Загалом для всього зображення суму можна визначити так:

$$v_{k,m} = \sum_{i=1}^3 \sum_{j=1}^3 x_{i+k,j+m} \cdot \omega_{ij} + \omega_0, \quad k, m = 0, 1, 2, \dots$$

У цифровій обробці сигналів ця сума називається згортою, а вікно з ваговими коефіцієнтами - імпульсним відгуком фільтру (або ядром фільтру).

І на виході формується набір карток ознак, які називаються каналами. Значні величини в кожній карті показують наявність ознаки в певному місці зображення. Якщо таких ознак буде кілька (на різних ділянках зображення), то на виході активуватимуться кілька нейронів, пов'язаних із цими областями. Завдяки цьому наступні шари згорток можуть узагальнювати знайдені особливості до більш складних, наприклад, еліпсів, прямокутників, різних перетинів ліній і т.п.

Значення карт ознак – це виходи функцій активації нейронів, тобто, тут: сума (згортка) проходить через функцію активації і формуються вихідні значення.

Якщо обробляється повнокольорове зображення, представлене, наприклад, трьома кольоровими компонентами RGB, кожна колірна компонента спочатку перетворюється своїм окремим, незалежним ядром, потім, обчислені карти ознак, складаються, до них додається зміщення і формується єдина підсумкова матриця ознак, яка проходить через функцію активації нейронів і виходять вихідні значення відповідному каналі [6]. Принцип роботи згорткової нейронної мережі на багатоканальному RGB зображенні зображений на рисунку 2.5.

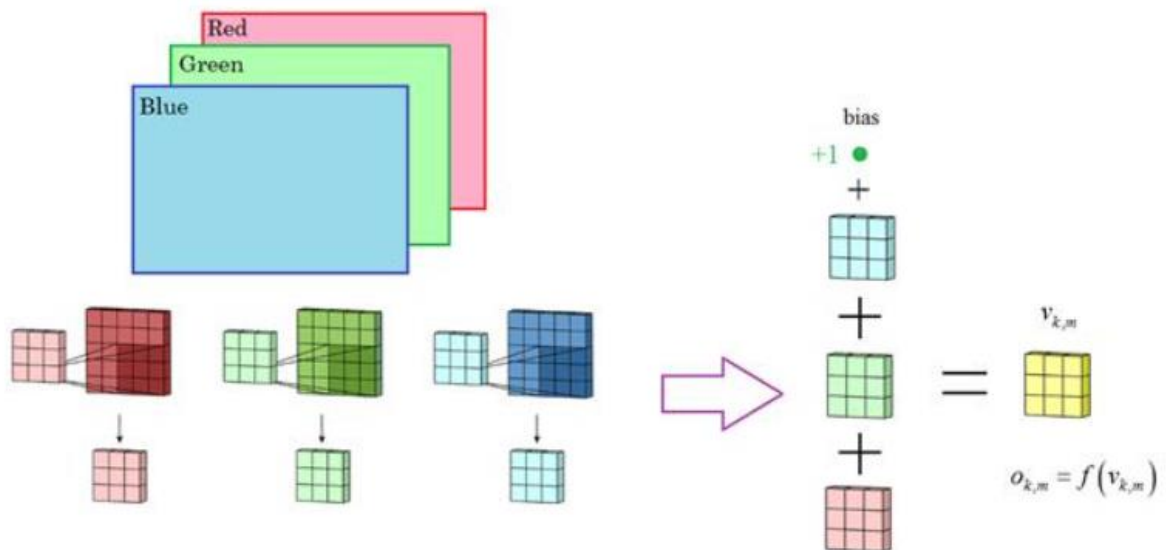


Рисунок 2.5 – Принцип роботи згорткової нейронної мережі на багатоканальному RGB зображенні

Ось так виглядає обробка повнокольорових зображень на першому прихованому шарі ЗНМ. Наступні шари працюють за таким же принципом, тільки замість колірних каналів RGB, виконується обробка каналів з карт ознак, сформованих на попередньому шарі.

Мета операції згортки — витягти із вхідного зображення високорівневі ознаки, наприклад лінії, краї. Згортковій нейронній мережі зовсім не обов'язково повинні бути обмежені лише одним шаром згортки. Традиційно перший шар відповідає за захоплення низькорівневих ознак, таких як краї, колір, градієнтна орієнтація і т.д. Завдяки доданим шарам архітектура

адаптується також до високорівневих ознак, видаючи нам мережу, яка має так само здорове розуміння зображень у наборі даних, що й ми.

Ця операція може давати два типи результатів: один, в якому згорнутий ознака має меншу розмірність порівняно з вхідним, в іншому розмірність або збільшується, або залишається незмінною.

У структурі згорткових мереж також зустрічаються пулінгові шари (Pooling Layer). Подібно до згортального шару, пулінговий шар необхідний для зменшення розміру згорнутого елемента в просторі. Це допомагає зменшити обчислювальну потужність, необхідну обробки даних, рахунок зменшення розмірності. Крім того, це важливо і для отримання домінуючих ознак, інваріантних обертання та позиціонування, таким чином підтримуючи процес ефективного навчання моделі [6].

Існує два типи пулінгу: максимальний (Max Pooling) та середній (Average Pooling). Максимальний пулінг повертає максимальне значення частини зображення, що покривається ядром. А Середній пулінг повертає середнє всіх значень частини зображення, що покривається ядром.

Максимальний пулінг також виступає в ролі шумоподавлювача (Noise Suppressant). Він повністю виключає шумові сигнали, також поєднує придушення шуму зі зменшенням розмірності. Середній пулінг просто використовує зменшення розмірності як спосіб придушення шуму. Тобто можна сказати, що Max Pooling працює набагато краще, ніж Average Pooling.

Найпопулярніші види шарів пулінгу:

- MaxPooling - відбір найбільших значень;
- MinPooling - відбір найменших значень;
- AveragePooling – вибір середніх значень.

Схематичне зображення пулінгово шару зображено на рисунку 2.5.

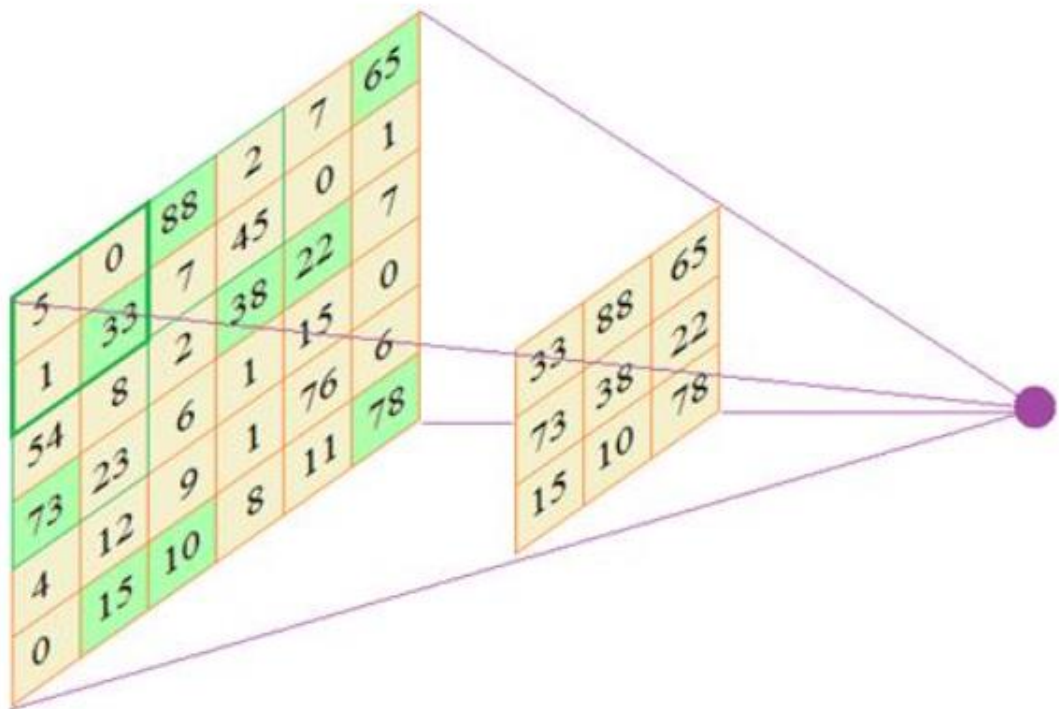


Рисунок 2.5 – pooling layer

В результаті отримуємо аналіз даних на більшому масштабі і завдяки цьому нейрони наступного шару здатні виділяти більш загальні ознаки на зображенні.

Як приклад наведемо операцію MaxPooling і це не випадково. Справа в тому, що, як правило, великі значення відповідають наявності певної ознаки, а малі – її відсутності. Тому, відбираючи максимальні числа, ми тим самим відбираємо знайдені ознаки і зберігаємо їх для подальшого аналізу на більшому масштабі. Саме операція MaxPooling, в основному, використовується в ЗНМ при аналізі зображень.

Давайте підсумуємо викладений матеріал і подивимося на загальну архітектуру ЗНМ. Для простоти припустимо, що на вхід подається зображення розміром 32 x 32 пікселя і воно проходить через 10 каналів першого прихованого шару. Приклад роботи згорткової нейронної мережі зображено на рисунку 2.6.

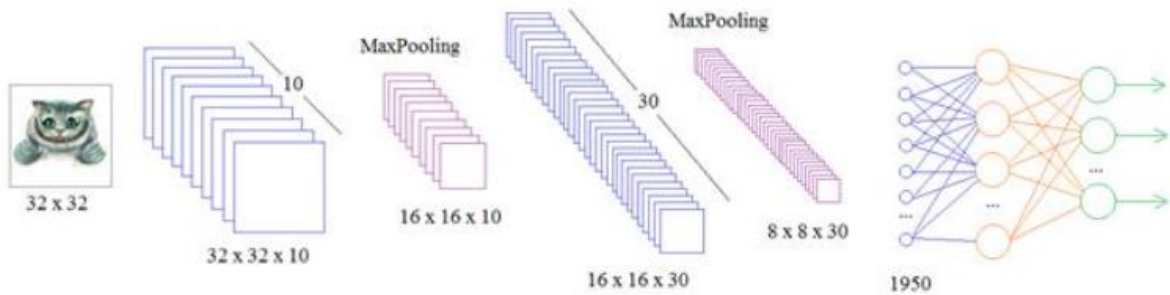


Рисунок 2.6 – приклад згорткової нейронної мережі

На виході отримуємо набір значень у вигляді тензора розмірністю $32 \times 32 \times 10$ елементів. Далі застосовується операція MaxPooling для переходу на більший масштаб. Наступний шар має вже 30 різних фільтрів, кожен, що складається з 10 ядер для обробки 10 каналів попереднього шару. На виході отримуємо тензор розмірністю $16 \times 16 \times 30$. Знову виконуємо операцію MaxPooling та скорочуємо розмірність до $8 \times 8 \times 30$. Такі перетворення можна продовжувати і далі, поки кожен вихідний сегмент стане розміром в 1 піксель. Або можна зупинитися на будь-якому етапі перетворення і подати обчислені карти ознак на вхід звичайної повнозв'язкової мережі. Кінцевий етап ЗНМ у завданнях класифікації, як правило, завершується повнозв'язковою мережею, на виході якої отримуємо ймовірність приналежності до того чи іншого класу.

Вихідний шар відповідає за формування ймовірностей приналежності вхідного образу тому чи іншому класу (деякому числу). Щоб досягти цього, вихідний шар повинен містити кількість нейронів, які відповідають кількості класів. Зважені та підсумовані сигнали далі модифікуються за допомогою функції активації ми використовуватимемо ReLU. Графіки найрозповсюдженіших функцій активацій зображено на рисунку 2.7.

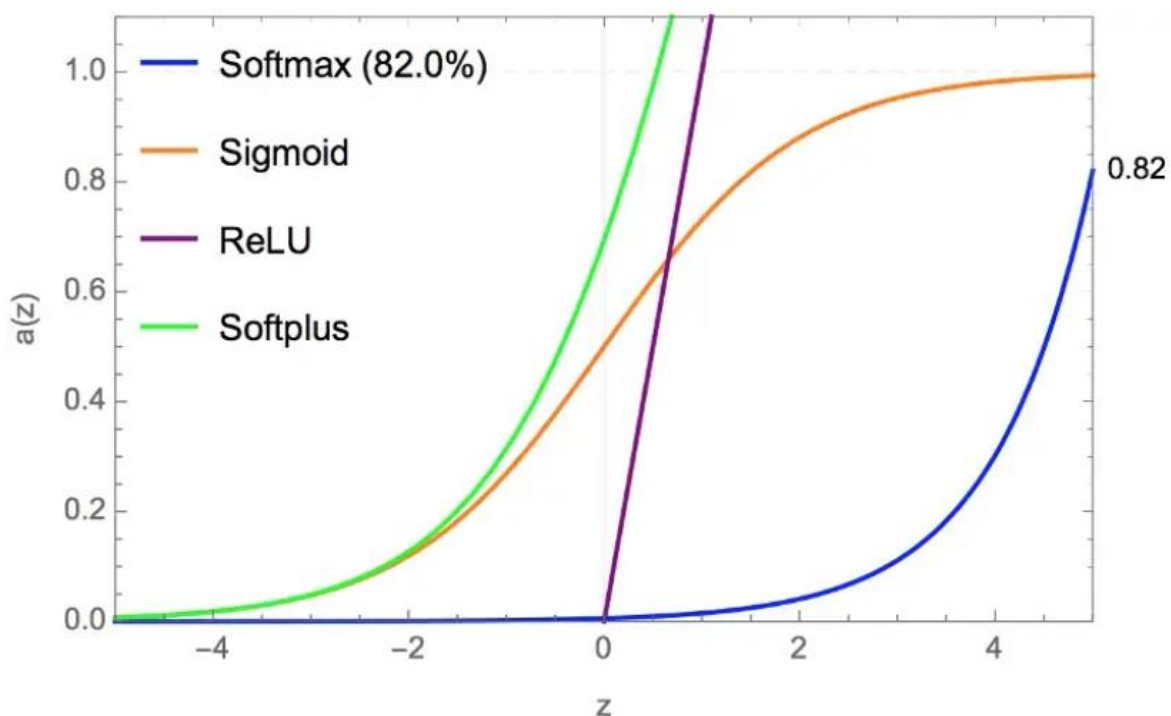


Рисунок 2.7 – Графіки функцій активації.

2.2 Grad-CAM алгоритм

Gradient-weighted Class Activation Mapping (Grad-CAM) [7], використовує градієнти будь-якої цільової концепції (скажімо «собака» в класифікаційній мережі або послідовності слів у мережі субтитрів), що перетікає в кінцевий згортковий шар для створення карти грубої локалізації, що висвітлює важливі регіони зображення для прогнозування концепції. На відміну від попередніх підходів, Grad-CAM застосовний до широкої різноманітності сімейств моделей CNN: CNN з повністю зв'язаними рівнями, CNN, що використовуються для структурованих вихідних даних (наприклад, субтитри), CNN, що використовуються в завданнях із мультимодальним

входом (наприклад, візуальні відповіді на запитання) або навчання з підкріпленням, усе без змін архітектури чи повторного навчання.

Grad-CAM створений з існуючими детальними візуалізаціями для класифікації зображень, підписів до зображень і візуальних відповідей на запитання (VQA), включаючи архітектури на основі ResNet.

У контексті моделей класифікації зображень візуалізації дають зрозуміти режими відмов цих моделей (показуючи, що, здавалося б, необґрунтовані прогнози мають розумні пояснення), є стійкі до збурень, є більш точними до базової моделі і допомагають досягти узагальнення моделі шляхом визначення зміщення набору даних.

Існує спосіб ідентифікації важливих нейронів за допомогою GradCAM і поєднання його з назвами нейронів, щоб надати текстові пояснення для модельних рішень.

Ряд попередніх робіт стверджували, що більш глибокі репрезентації в CNN охоплюють візуальні конструкції вищого рівня. Крім того, згорткові шари природним чином зберігають простір інформації, який втрачається у повністю пов'язаних шарах, тому можна очікувати, що останні згорткові шари матимуть найкращий компроміс між семантикою високого рівня та просторовою інформацією. Нейрони в цих шарах шукають інформацію про клас на зображенні (скажімо, частини об'єкта). Grad-CAM використовує інформацію про градієнт, що надходить у останній згортковий шар CNN для призначення важливості значення для кожного нейрона для конкретного рішення, яке цікавить.

Хоча методика є досить загальною, тому її можна використовувати щоб пояснити активації на будь-якому рівні глибокої мережі, тому зосередимось лише на поясненні рішень вихідного рівня.

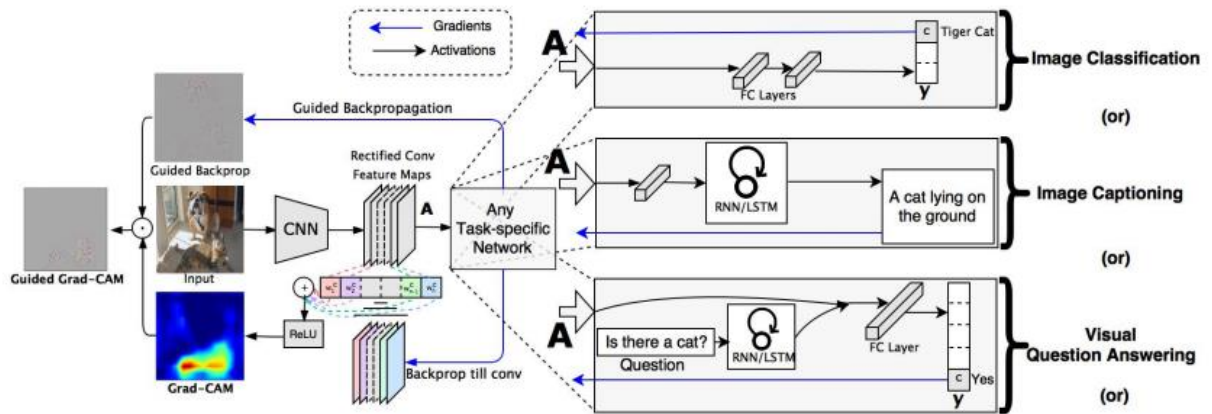


Рисунок 2.8 – схематичне зображення згорткової нейронної мережі з локалізацією Grad-CAM.

Як показано на рисунку 2.8, щоб отримати клас-дискримінаційних карт локалізації Grad-CAM $L_{Grad-CAM}^c \in R^{u \times v}$ ширини u і висота v для будь-якого класу c , ми спочатку обчислюємо градієнт оцінки за c, y^c (перед софтмаксом), відносно активації карти ознак A^k згорткового шару, тобто $\frac{\partial y^c}{\partial A^k}$.

Ці градієнти, що повертаються, є глобальним середнім над розмірами ширини та висоти (з індексами i та j відповідно), щоб отримати вагові коефіцієнти важливості нейронів α_k^c .

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \text{global average pooling} \frac{\partial y^c}{\partial A_{ij}^k} \text{gradient via backprop}$$

Під час обчислення $\alpha_{i,j}^k$ під час зворотного поширення градієнтів щодо активацій, точні суми обчислень до послідовних матричних добутків вагових матриць і градієнта щодо функцій активації до останнього шару згортки, до якого поширюються градієнти. Отже, цей ваговий коефіцієнт α с k представляє часткову лінеаризацію глибокої мережі нижче за течією від A та відображає «важливість» карти ознак k для цільового класу c . Виконуємо зважену комбінацію карт прямої активації та дотримуємося ReLU.

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \text{linear combination}$$

Це призводить до грубої теплової карти такого самого розміру як карти згортки (14×14 у випадку останні згорткові шари VGG [9] і AlexNet [10] мережі). Застосовується ReLU до лінійної комбінації карт, оскільки нас цікавлять лише функції, які мають позитивний вплив на цікавий клас, тобто пікселі інтенсивність якого слід збільшити, щоб збільшити y^c . Негативні пікселі, ймовірно, належать до інших категорій на зображенні. Без цього ReLU, карти локалізації іноді виділяють не тільки бажаний клас і гірше справляються з локалізацією.

Загалом, y^c не обов'язково має бути оцінкою класу, створеною згортковою мережею для класифікації зображень. Це може бути будь-яка диференційована активація, включаючи слова або відповіді на запитання.

Використовуючи Grad-CAM, ми можемо отримати пояснення, які підкреслюють підтримку регіонів, які б змусили мережу змінити свій прогноз. Як наслідок, видалення концепцій, що зустрічаються в цих регіонах, дасть змогу підкреслити модель більш впевнену щодо свого прогнозу. Ми посилаємося на цю модальність пояснення як контрфактичні пояснення. Зокрема, ми заперечуємо градієнт y^c по відношенню до карт ознак A згорткового шару. Таким чином ваги α_k^c тепер стають.

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \quad \rightsquigarrow \text{global average pooling} \quad - \frac{\partial y^c}{\partial A_{ij}^k} \quad \rightsquigarrow \text{negative gradients}$$

Беремо зважену суму прямої активації карти, A , з вагами α_k^c , і дотримуйтесь ReLU, щоб отримати контрфактичними поясненнями, як показано на рисунку 3.



Рис.2.9 – приклад роботи Grad-CAM

2.3 Висновки до розділу

В другому розділі були описані математичні моделі згорткових нейронних мереж та алгоритму Grad-CAM які будуть використовуватися в подальших експериментах. Була розглянута структура згорткових нейронних мереж.

РОЗДІЛ 3 ПОБУДОВА ВЛАСНОЇ CNN ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Вибір та підготовка даних

Для побудови, тренування та перевірки згорткової нейронної мережі та алгоритму Grad-CAM було обрано набір даних NASA Space Apps Challenge з фотографіями пожежі та без, приклади яких зображені на рисунках 3.1 та 3.2.

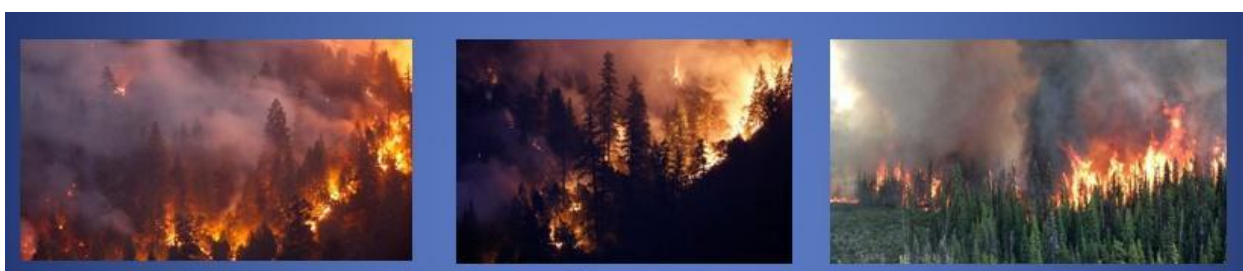


Рис 3.1 - Приклад зображень з пожежею



Рис 3.2 - Приклад зображень без пожежі

Загалом у вибірці присутні:

- 755 зображень пожеж, деякі з них містять сильний дим
- 244 зображення природи без пожеж

Дані будуть розділені на три різні категорії: навчання, перевірка та тестування. Навчальні дані використовуватимуться для навчання моделі CNN глибокого навчання, а її параметри будуть точно налаштовані за допомогою даних перевірки. Нарешті, продуктивність даних буде оцінена за допомогою тестових даних (даних, які модель раніше не бачила).

Усі зображення були оброблені алгоритмом локалізації Grad-CAM.

3.2 Побудова CNN

В даній роботі була спроектована та розроблена згорткова нейронна мережа та використано алгоритм Grad-Cam. Згорткова мережа має архітектуру зображену на рисунку 3.3.

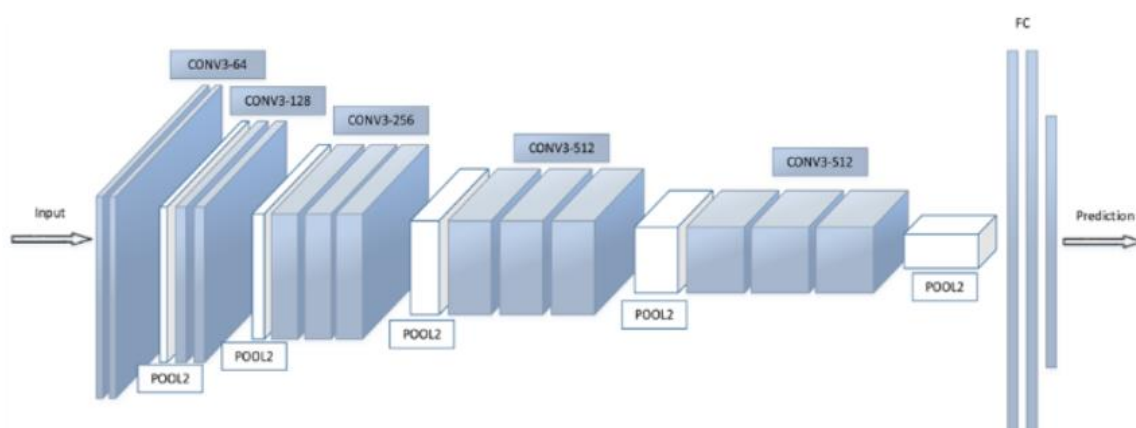


Рисунок 3.3 – Архітектура побудованої згорткової мережі

Дана мережа має наступні характеристики:

1. Кількість фільтрів – 32.
2. Розміри ядра - (3, 3).
3. Кольоровий простір - RGB (250, 250, 3)
4. Функція активації - 'ReLU'
5. MaxPooling із загальною кількістю пікселів 4 (2, 2)

3.3 Алгоритм Grad-CAM

Усі зображення були оброблені алгоритмом Grad-CAM. Нижче на рисунках 3.4 та 3.5 наведено приклади роботи алгоритму. Після обробки зображення надходили на вхід згорткової мережі.



Рисунок 3.4 - Приклад зображення



Рис3.5 - Приклад зображення з тепловою картою ознак

3.4 Аналіз та порівняння результатів

Для перевірки ефективності Grad-CAM алгоритму у розв'язуванні задачі розпізнавання пожежі було проведено два експерименти. В першому – початкові зображення надходили на вхід нейронної мережі без попередньої обробки. У другому – зображення спочатку проходили через Grad-CAM алгоритм який на виході видавав теплову карту ознак і вже ці зображення надходили на вхід згорткової нейронної мережі.

На рисунку 3.6 зображено графік точності мережі без локалізації. На рисунку 3.7 зображено графік функції втрат мережі без локалізації. На рисунку 3.8 зображено графік точності мережі з локалізацією. На рисунку 3.9 зображено графік функції втрат мережі з локалізацією. На рисунку 3.10 зображено матрицю невідповідностей мережі без локалізації. На рисунку 3.11 зображено матрицю невідповідностей мережі з локалізацією.

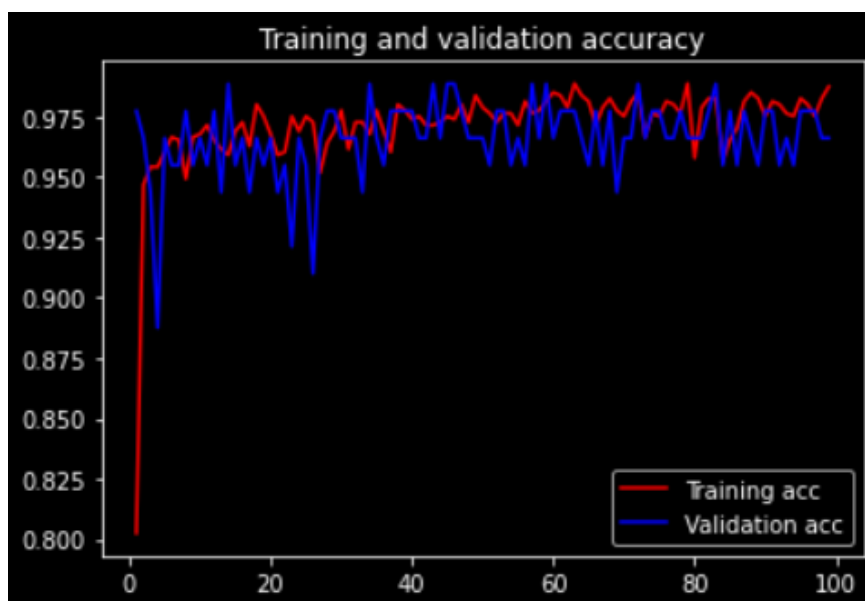


Рис. 3.6 - Точність мережі без локалізації

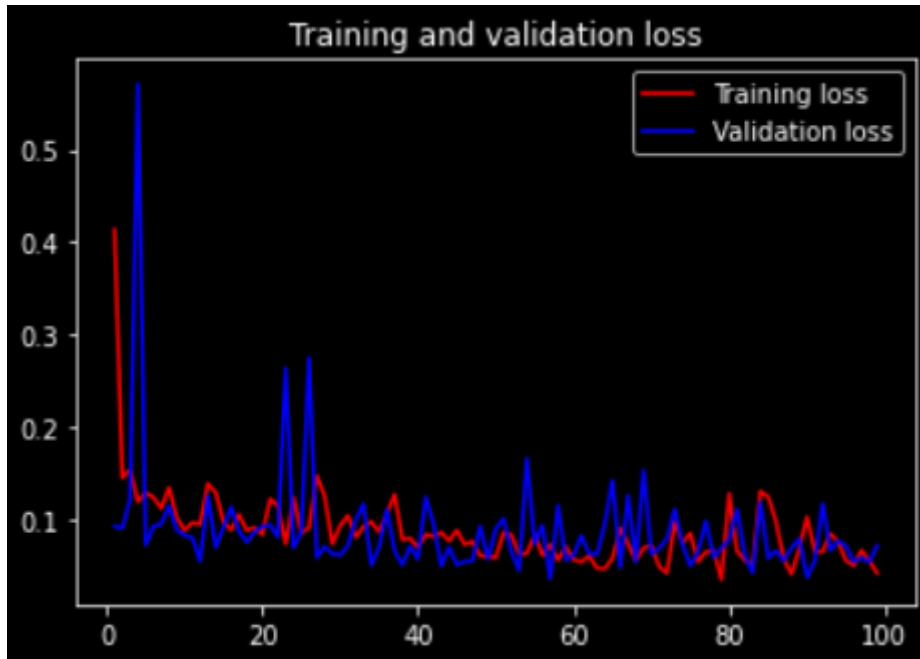


Рис. 3.7 - Функція втрат мережі без локалізації

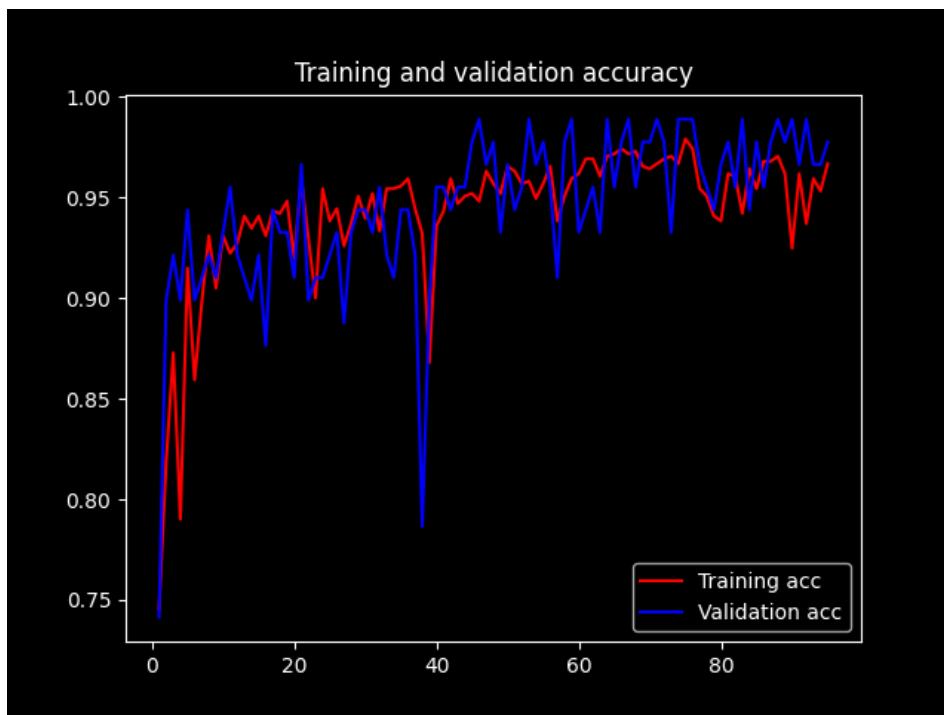


Рис. 3.8 - Точність мережі з локалізацією

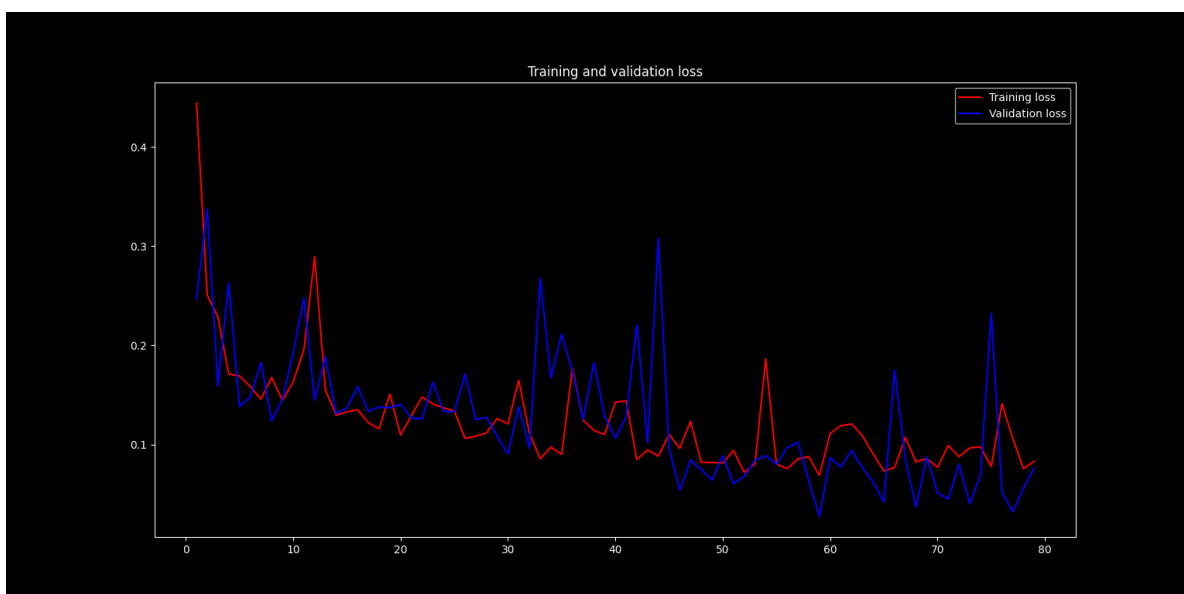


Рис. 3.9 - Функція втрат мережі з локалізацією

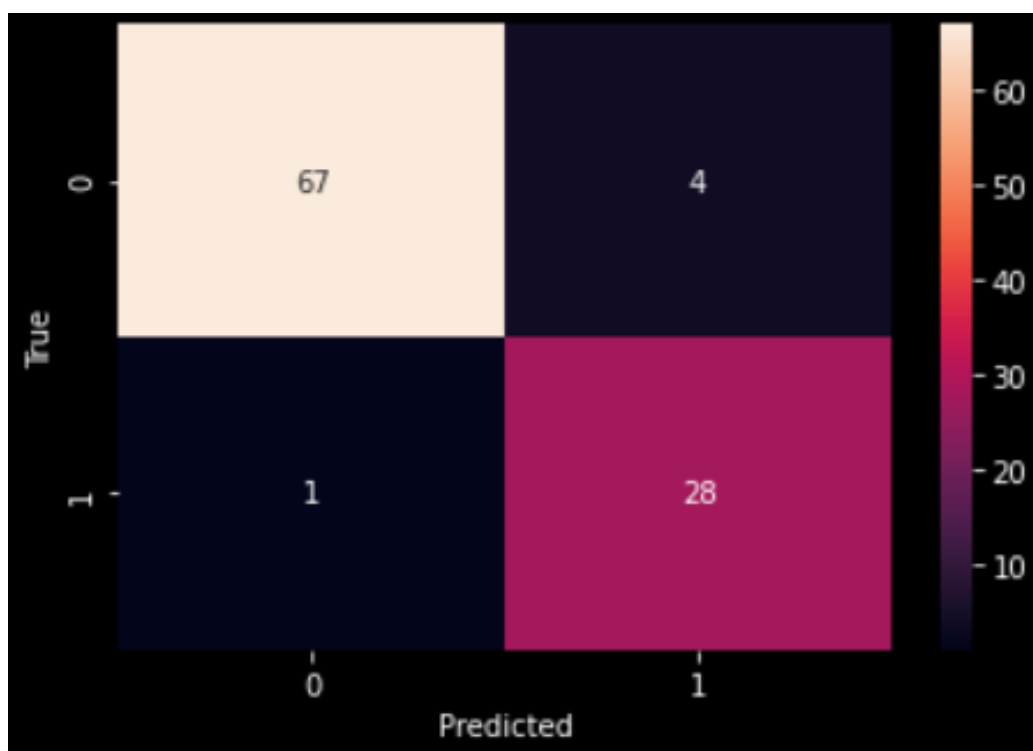


Рис. 3.10 - Матриця невідповідностей мережі без локалізації

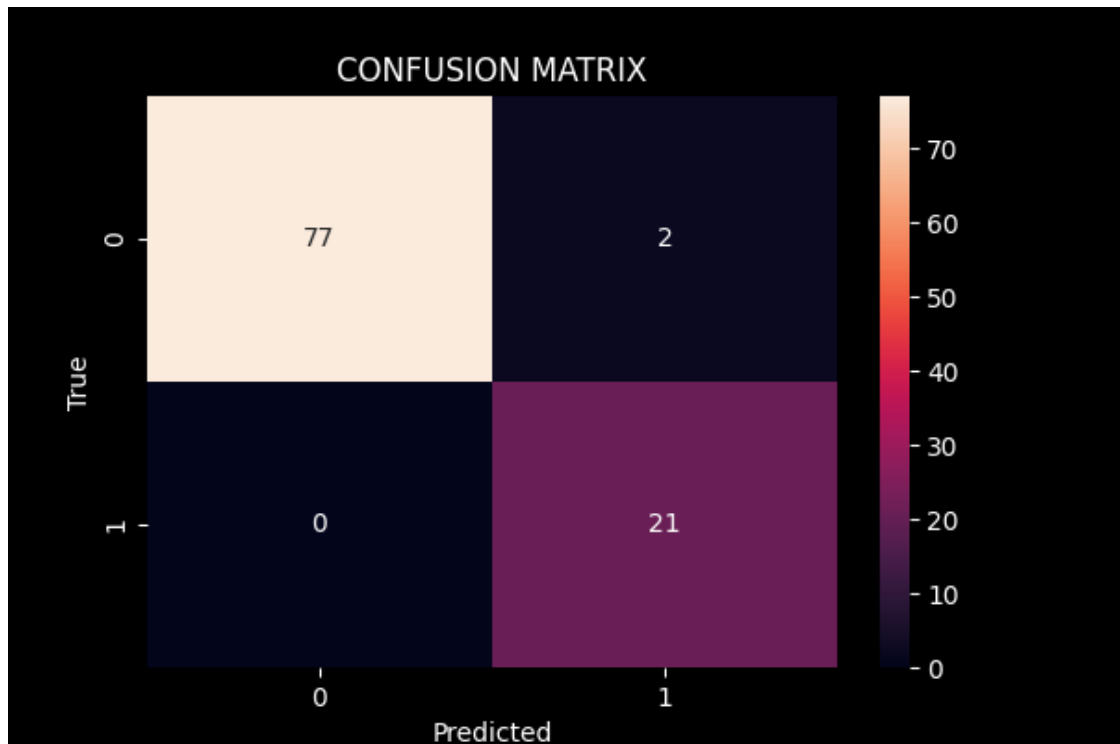


Рис. 3.11 - Матриця невідповідностей мережі з локалізацією

Можемо спостерігати, як точність покращується з часом, зрештою стабілізуючись. Відповідно, втрати з часом зменшуються. Подібні графіки можуть допомогти діагностувати перенавчання. Якби ми побачили висхідну криву функції втрати (U-подібна форма на графіку), ми б запідозрили, що модель починає запам'ятовувати тестовий набір і погано узагальнювала б нові дані.

Як видно з вище наведених рисунків мережа з попередньо локалізованими зображеннями дає кращий результат порівняно з мережею без локалізації. Результати експериментів занесені у таблицю 3.1.

Таблиця 3.1 – Результати експериментів.

Точність мережі без локалізації	0.949999988079071
Функція втрат без локалізації	0.11980465799570084
Точність мережі з локалізацією	0.9700000286102295
Функція втрат мережі з локалізацією	0.05301668122410774

3.5 Висновки до розділу

У даному розділі були побудовані згорткова нейронна мережа для розпізнавання пожеж, а також використано алгоритм локалізації Grad-CAM. Була обрана вибірка зображень, та проведено ряд експериментів під час яких порівнювалась ефективність згорткової нейронної мережі з попередньою локалізацією зображень за допомогою Grad-CAM та без попередньої локалізації. В загальному було проведено два експерименти. В першому – початкові зображення надходили на вхід нейронної мережі без попередньої обробки. У другому – зображення спочатку проходили через Grad-CAM алгоритм який на виході видавав теплову карту ознак і вже ці зображення надходили на вхід згорткової нейронної мережі.

За результатами експериментів з'ясувалось, що попередня обробка зображень алгоритмом Grad-CAM покращує точність нейронної мережі. Точність мережі без Grad-CAM – 95%, точність з алгоритмом Grad-CAM – 97%.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Стартап — це бізнес-структура, що розвивається та працює на основі інновацій, створена для вирішення проблеми шляхом надання нової пропозиції в умовах надзвичайної невизначеності.

Власне, стартап – це бізнес, який:

- швидко росте;
- порушує ринок або галузь (щось нове, що змушує конкурентів удосконалюватись);
- вирішує проблему;
- працює в умовах надзвичайної невизначеності.

Багато підприємців і відомих бізнес-магнатів визначають стартап як культуру та менталітет побудови бізнесу на основі інноваційної ідеї для вирішення критичних проблем.

Одне, що відрізняє стартапи від інших компаній — це зв'язок між їхнім продуктом та його попитом. Стартапи мають продукти, орієнтовані на невикористаний ринок. Підприємці-початківці знають ідеальну стратегію, щоб створити продукт, який хоче ринок, а також охопити й обслуговувати їх усіх. Це викликає швидке зростання.

4.1 Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

- зміст ідеї;
- можливі напрямки застосування;
- можливі напрямки застосування;

- основні вигоди, що може отримати користувач товару;
- чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці (таблиця 4.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Дана комплексна система дозволяє розв'язати проблему виявлення пожеж	Виявлення пожежі у приміщеннях	Запобігання людських жертв та втрати майна в наслідок пожежі
	Виявлення пожежі на відкритій місцевості	Запобігання втрати майна та природних об'єктів

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;
- визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають:
 - гірші значення (W, слабкі);
 - аналогічні (N, нейтральні) значення;

– кращі значення (S, сильні) (табл. 4.2).

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Технікоеконімічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слаб ка сторона)	N (нейт ра льна сторона)	S (силь на сторона)
		Мій проект	Fire Watch	WSN			
1.	Форма виконання	Надава ння послуг	Надава ння послуг	Надава ння послуг			+
2.	Собівартість	Середн я	Висока	Висока	+		
3.	Функціонал	Широк ий	Широк ий	Широк ий		+	

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 4.3):

1. За якою технологією буде виготовлено товар згідно ідеї проекту?
2. Чи існують такі технології, чи їх потрібно розробити/добробити?
3. Чи доступні такі технології авторам проекту?

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Створення системи розпізнавання пожежі за допомогою камер	Використання мови програмування Python	Наявна	Доступна
		Використання мови програмування C#	Наявна	Недоступні
		Використання мови C++	Наявна	Недоступні
Обрана технологія реалізації ідеї проекту: мова програмування Python.				

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як Python через доступність та безкоштовність.

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	15
2	Загальний обсяг продаж, грн/ум.од	10000
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі, %	10%

За результатами аналізу таблиці 4.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та сформовано орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Точна та швидке виявлення пожежі	Власник бізнесу	Швидка передача інформації до власника	Простота використання, висока точність
Надання інформації про рівень та місце пожежі	Власник бізнесу	Цікавить простота у використанні, низька ціна підтримки системи	Швидкість створення, низька ціна

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6, 4.7).

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
-------	--------	---------------	--------------------------

1	Конкуренція	Вихід на ринок продуктів з кращими характеристиками	Передбачити додаткові переваги власного програмного продукту (ПП) для того, щоб повідомити про них саме після виходу на ринок конкурентів. Вдосконалення технічних моментів власного продукту. Обрати нову цільову аудиторію і зосередитися на ній: зниження цін.
2	Зміна потреб користувачів	Користувачам необхідний сервіс з більшим/новим функціоналом.	Розроблення гнучкої архітектури програмного забезпечення для легшого впровадження нового функціоналу

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Гнучкі ціни	Зменшення ціни товару задля збільшення попиту	Введення власних гнучких цін
2	Поява нових методів обробки зображення	З'являться нові методи, що будуть швидше, та більш точно обробляти зображення	Покращити ПП додаванням нового функціоналу, розширення можливостей

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Тип конкуренції - чиста	Існує величезна кількість конкурентів на ринку.	Якісно провести рекламу.
2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу ПП таким чином, щоб можна було легко переробити даний ПП для використання у галузях інших країн.
3. За галузевою ознакою - міжгалузева	Продукт може використовуватись для різних галузей	Постійне вдосконалення продукту, що не має прив'язки до сфери
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між видами ПП, їх особливостями.	Створити ПП, враховуючи ПП конкурентів
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПП, щоб собівартість була нижчою	Удосконалення моделі. Використання більш дешевих технологій для розробки, ніж використовують конкуренти, але тільки якщо ці технології відповідають необхідним вимогам якості.
6. За інтенсивністю - не марочна	Бренд присутній, але його роль незначна	Реклама, участь у конференціях, семінарах.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
	Fire Watch, WSN	Наявність вже існуючих рішень	-	Контроль якості продукту	Наявність більш широкого функціоналу, зручнішого інтерфейсу та авторитет
Висновки:	Досить інтенсивна боротьба з іншими	Є можливість виходу на ринок, але є і конкуренти.	-	Клієнти диктують умови роботи на ринку: ефектив	Необхідно випускати ПЗ не гірше, ніж у конкурентів та розширяти

	гравцям и			ність	функціонал.
--	--------------	--	--	-------	-------------

За результатами аналізу було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію.

Цей висновок був врахований при формулюванні переліку факторів конкурентоспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в таблиці, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. 4.6, 4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності.

Аналіз оформлено у (табл. 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Ціна	Один із факторів для вибору продукту клієнтом.
2	Якість	Один із факторів для вибору продукту клієнтом.
3	Зручність роботи з програмою	Дозволяє користувачу легко працювати з програмою

За визначеними факторами конкурентоспроможності (табл. 4.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/ п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні							
			-3	-2	-1	0	+1	2	3	
1	Ціна	15		*						
2	Якість	10					*			
3	Зручність роботи з програмою	10				*				

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 4.11).

Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: Якість Низька ціна Висока швидкодія	Слабкі сторони: Дуже насичений ринок, мала кількість функціоналу, відсутня кросплатформеність.
--	---

<p>Можливості:</p> <p>насичення ринку новим підходом до прогнозування; різноманітна клієнтура, вдосконалення системи</p>	<p>Загрози:</p> <p>Конкуренція</p>
--	------------------------------------

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 4.13).

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	PR, просування бренду	50%	6 місяців
2	Перехід на безкоштовне розповсюдження	75%	3 місяців
3	Партнерство для об'єднання продукції	65%	2 місяці

Після аналізу було обрано альтернативу №2.

4.3 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 4.14).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Підприємці	Висока	Високий	Сильна	Середньо
2	Великі компанії	Висока	Високий	Сильна	Середньо
3	Маленькі компанії.	Низька	Низький	Слабка	Складно
Які цільові групи обрано: 1,2,3					

За результатами аналізу потенційних груп споживачів було обрано цільові групи, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Постійне оновлення і покращення продукту	Ринкове позиціонування на індивідуальних користувачів	Швидкодія, якість продукту	Концентрований маркетинг

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні.	Компанія буде шукати нових споживачів та забирати існуючих у	Буде копіювати, удосконалювати та створювати свої унікальні пропозиції	Зайняття конкурентної ніші

		конкурентів		
--	--	-------------	--	--

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (таблиця 4.16) розроблено стратегію позиціонування (таблиця 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартаппроекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Легкість розуміння, зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для генерації рекомендацій.	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування; Практичність та точність результату

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

4.4 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього підсумовано результати попереднього аналізу конкурентоспроможності товару (таблиця 4.18). Концепція товару – письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Швидкість отримання результату	Швидке надання інформації про пожежу	Необхідно покращити швидкість навчання моделі для виявлення пожежі
2	Зручність застосування	Нативна підтримка мови програмування Python	Розробка зручного прикладного програмного інтерфейсу
3	Практичність та точність результату	Користувач отримує точні результати виявлених пожеж	Користувач на виході роботи ПП отримує модель та прогноз, котрі відповідають необхідним

			показникам варіативності та точності.
--	--	--	---------------------------------------

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 4.19).

1-й рівень. При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень. Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень Товар з підкріпленням (супроводом) – додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Встановлення системи нагляду для виявлення пожежі		
II. Товар реальному виконанні	Властивості/характеристик и	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Індивідуальний підхід.	1.Нм	1.Технологічна
	2. Низька ціна.	2.Нм	2.Економічна
	3. Простота у використанні.	3.Нм	3.Технологічна
Якість: тестування фірмами аудиторами			

Пакування: відсутнє
Марка: SFD

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 4.20). Аналіз проведено експертним методом.

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	20000\$	37000\$	У всіх трьох груп високий рівень доходів	9000\$--

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 4.21).

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Канал нульового рівня	Продаж	0(напрямую)	Власна

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Інтеграція API у клієнтській системі	Інтернет	Низька ціна, простота використання, ефективність	Показати переваги рішення над конкурентами, виділити ключові особливості	Створення сайту продукту, розповсюдження інформації про продукт на спеціалізованих ресурсах.

Було визначено, що придбання продукту буде проводитись через мережу Інтернет або при безпосередньому спілкуванні із представниками компанії. Розповсюдження інформації про продукт буде проводитись виключно через Інтернет, адже аудиторія даного продукту активно користується всесвітньою мережею.

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз

можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

4.5 Висновки

В даному розділі проведено підготовчий аналіз для впровадження розробленої системи в якості стартап проекту. Досліджено аналогічні конкурентні системи, встановлено сильні та слабкі сторони системи в порівнянні з ними. Також було досліджено можливі шляхи розповсюдження продукту та його ймовірну аудиторію, рівень доходів та ймовірну ціну продукту, що розробляється.

Було проведено аналіз потенційних ризиків і можливостей, а також розраховані основні фінансово-економічні показники проекту. Отримані результати кажуть про те, що реалізація проекту є доцільною. Було визначено сильні сторони проекту: зручність у використанні, ціна, якість. Серед слабких варто виділити повільне навчання. Варто відмітити можливість реклами продукту на спеціалізованих ресурсах із зазначенням сильних сторін проекту.

ВИСНОВКИ

У даній дисертації було запропоновано власну архітектуру згорткової нейронної мережі для розпізнавання пожежі, а також запропоновано

використовувати алгоритм локалізації Grad-CAM для покращення результатів роботи мережі.

Задача розпізнавання пожежі належить до задач класифікації для вирішення яких було розроблено

Загалом створена мережа на високому рівні справляється із задачею локалізації та класифікації зображень. Основними критеріями успіху є великий процент точного розпізнавання. Запропонована нейронна мережа з локалізацією показала вищу точність у виявленні пожежі на зображенні, ніж мережа без локалізації. Точність досягнута мережею з локалізацією: 97%, точність мережі без локалізації: 95%.

ПЕРЕЛІК ПОСИЛАНЬ

1. Зібцев С.В. Аналіз особливостей лісопожежної обстановки та стану протипожежної охорони лісу в зонах радіаційного забруднення. Наукові доповіді НАУ. – 2006. – Вип. 4(5). – 17 с.
2. Udaya D., Lumini B., RidmaW., Kishanga K., Bathiya J. Forest fire detection system using wireless sensor networks and machine learning, p2.
3. Yansong Y. G., Liu Y. C., Liu Y., Forest Fire Detection Using Artificial Neural Network Algorithm Implemented in Wireless Sensor Networks. ZTE Communications, 2015
4. Horng W. -B., Peng J.-W. Image based fire detection using neural networks.// ResearchGate, 2006.
5. Помазан В. В. Розроблення програмного застосунку для ідентифікації емоцій на зображеннях. М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2022. – с. 34.
6. Senior A.W. Privacy protection and face recognition. / Senior A.W, Pankanti S.; in: Li S.Z., Jain A.K..editors. // Handbook of Face Recognition. 2nd ed. –Springer. 2011.
7. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra
8. K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In ICLR, 2015.
9. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012

ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО КОДУ

1. Файл CNN.py

```

import pandas as pd
import numpy as np
import datetime as dt
import os
import os.path
from pathlib import Path
import glob
import cv2
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dropout,
Flatten, Dense, BatchNormalization
from tensorflow.keras.layers import SpatialDropout2D
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import confusion_matrix, accuracy_score,
classification_report, roc_auc_score, roc_curve
from tensorflow.keras.utils import plot_model
from tensorflow.keras.preprocessing import image
from PIL import Image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,
img_to_array, load_img

dir_ = Path(r'D:\input\loc-fire-dataset\fire_dataset')
png_filepaths = list(dir_.glob(r'**\*.png'))
print(png_filepaths)

labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1],
png_filepaths))

filepaths = pd.Series(png_filepaths, name = 'File').astype(str)
labels = pd.Series(labels, name = 'Label')

df = pd.concat([filepaths, labels], axis=1)

df['Label'].replace({"non_fire_images":"nofire","fire_images":"fire"
}, inplace=True)

# Візуалізація даних набору даних у стовпчиковій діаграмі Змішування
# фрейму даних і скидання індексу
df = df.sample(frac = 1, random_state = 83).reset_index(drop = True)
df.head()

png_error = '../input/fire-
dataset/fire_dataset/non_fire_images/non_fire.189.png'
df = df.loc[~(df.loc[:, 'Label'] == png_error), :]
print(df)

train_df, test_df = train_test_split(df, train_size = 0.9,
random_state = 86)

```



```

width_shift_range = 0.1,
height_shift_range = 0.1,
horizontal_flip = True,
vertical_flip = True,
validation_split = 0.1)

test_datagen = ImageDataGenerator(rescale = 1./255)

print("Підготовка навчального набору даних...")
training_set = train_datagen.flow_from_dataframe(
    dataframe = train_df,
    x_col = "File",
    y_col = "Label",
    target_size = (250, 250),
    color_mode = "rgb",
    class_mode = "binary",
    batch_size = 32,
    shuffle = True,
    seed = 1,
    subset = "training")

print("Підготовка набору даних для перевірки...")
validation_set = train_datagen.flow_from_dataframe(
    dataframe = train_df,
    x_col = "File",
    y_col = "Label",
    target_size = (250, 250),
    color_mode = "rgb",
    class_mode = "binary",
    batch_size = 32,
    shuffle = True,
    seed = 1,
    subset = "validation")

print("Підготовка тестового набору даних...")
test_set = test_datagen.flow_from_dataframe(
    dataframe = test_df,
    x_col = "File",
    y_col = "Label",
    target_size = (250, 250),
    color_mode = "rgb",
    class_mode = "binary",
    shuffle = False,
    batch_size = 32)

print('Генератори даних готові!')

CNN = Sequential()

CNN.add(Conv2D(32, (3, 3), input_shape = (250, 250, 3), activation =
'relu'))

CNN.add(MaxPooling2D(pool_size = (2, 2)))

CNN.add(Conv2D(32, (3, 3), activation = 'relu'))
CNN.add(MaxPooling2D(pool_size = (2, 2)))

CNN.add(Conv2D(64, (3, 3), activation = 'relu'))
CNN.add(SpatialDropout2D(0.2))
CNN.add(MaxPooling2D(pool_size = (2, 2)))

```

```

CNN.add(Conv2D(128, (3, 3), activation = 'relu'))
CNN.add(SpatialDropout2D(0.4))
CNN.add(MaxPooling2D(pool_size = (2, 2)))

CNN.add(Flatten())

CNN.add(Dense(units = 256, activation = 'relu'))
CNN.add(Dropout(0.4))
CNN.add(Dense(units = 1, activation = 'sigmoid'))
# Callbacks
callbacks = [EarlyStopping(monitor = 'loss', mode = 'min', patience
= 20, restore_best_weights = True)]

print(CNN.summary())
#plot_model(CNN, to_file='CNN_model.png', show_layer_names = True ,
show_shapes = True)

CNN.compile(optimizer='adam', loss = 'binary_crossentropy', metrics
= ['accuracy'])

start = dt.datetime.now()

CNN_model = CNN.fit(training_set, epochs = 100, validation_data =
validation_set, callbacks = callbacks)

end = dt.datetime.now()
time_CNN = end - start
print ('\nЧас навчання та перевірки', time_CNN)

acc = CNN_model.history['accuracy']
val_acc = CNN_model.history['val_accuracy']
loss = CNN_model.history['loss']
val_loss = CNN_model.history['val_loss']
epochs = range(1, len(acc) + 1)

plt.title('Training and validation accuracy')
plt.plot(epochs, acc, 'red', label='Training acc')
plt.plot(epochs, val_acc, 'blue', label='Validation acc')
plt.legend()

"""
plt.show()

score = CNN.evaluate(test_set)
print("Test Loss:", score[0])
print("Test Accuracy:", score[1])

y_pred = CNN.predict(test_set)
y_pred = np.round(y_pred)

print(classification_report(y_test, y_pred))

plt.figure(figsize = (6, 4))

sns.heatmap(confusion_matrix(y_test, y_pred), annot = True)
plt.title("CONFUSION MATRIX")
plt.xlabel("Predicted")
plt.ylabel("True")

```



```
plt.show()
```

2. Файл gradcam.py

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from pathlib import Path
import os
import pandas as pd

# Display
from IPython.display import Image, display
import matplotlib.pyplot as plt
import matplotlib.cm as cm

def get_img_array(img_path, size):
    # `img` is a PIL image of size 299x299
    img = keras.preprocessing.image.load_img(img_path,
target_size=size)
    # `array` is a float32 Numpy array of shape (299, 299, 3)
    array = keras.preprocessing.image.img_to_array(img)
    # We add a dimension to transform our array into a "batch"
    # of size (1, 299, 299, 3)
    array = np.expand_dims(array, axis=0)
    return array

def make_gradcam_heatmap(img_array, model, last_conv_layer_name,
pred_index=None):
    # First, we create a model that maps the input image to the
    activations
    # of the last conv layer as well as the output predictions
    grad_model = tf.keras.models.Model(
        [model.inputs],
        [model.get_layer(last_conv_layer_name).output, model.output]
    )

    # Then, we compute the gradient of the top predicted class for
    our input image
    # with respect to the activations of the last conv layer
    with tf.GradientTape() as tape:
        last_conv_layer_output, preds = grad_model(img_array)
        if pred_index is None:
            pred_index = tf.argmax(preds[0])
            class_channel = preds[:, pred_index]

    # This is the gradient of the output neuron (top predicted or
    chosen)
    # with regard to the output feature map of the last conv layer
```

```

grads = tape.gradient(class_channel, last_conv_layer_output)

# This is a vector where each entry is the mean intensity of the
gradient
# over a specific feature map channel
pooled_grads = tf.reduce_mean(grads, axis=(0, 1, 2))

# We multiply each channel in the feature map array
# by "how important this channel is" with regard to the top
predicted class
# then sum all the channels to obtain the heatmap class
activation
last_conv_layer_output = last_conv_layer_output[0]
heatmap = last_conv_layer_output @ pooled_grads[..., tf.newaxis]
heatmap = tf.squeeze(heatmap)

# For visualization purpose, we will also normalize the heatmap
between 0 & 1
heatmap = tf.maximum(heatmap, 0) / tf.math.reduce_max(heatmap)
return heatmap.numpy()

def save_and_display_gradcam(img_path, heatmap, cam_path="cam.jpg",
alpha=0.4):
    # Load the original image
    img = keras.preprocessing.image.load_img(img_path)
    img = keras.preprocessing.image.img_to_array(img)

    # Rescale heatmap to a range 0-255
    heatmap = np.uint8(255 * heatmap)

    # Use jet colormap to colorize heatmap
    jet = cm.get_cmap("jet")

    # Use RGB values of the colormap
    jet_colors = jet(np.arange(256))[:, :3]
    jet_heatmap = jet_colors[heatmap]

    # Create an image with RGB colorized heatmap
    jet_heatmap =
keras.preprocessing.image.array_to_img(jet_heatmap)
    jet_heatmap = jet_heatmap.resize((img.shape[1], img.shape[0]))
    jet_heatmap =
keras.preprocessing.image.img_to_array(jet_heatmap)

    # Superimpose the heatmap on original image
    superimposed_img = jet_heatmap * alpha + img
    superimposed_img =
keras.preprocessing.image.array_to_img(superimposed_img)

    # Save the superimposed image
    superimposed_img.save(save_path)

    # Display Grad CAM
    display(Image(cam_path))

model_builder = keras.applications.xception.Xception
img_size = (299, 299)
preprocess_input = keras.applications.xception.preprocess_input
decode_predictions = keras.applications.xception.decode_predictions

```

```

last_conv_layer_name = "block14_sepconv2_act"

# The local path to our target image


```

```
heatmap = make_gradcam_heatmap(img_array, model,
last_conv_layer_name)

# Display heatmap
#plt.matshow(heatmap)
#plt.show()

save_path = img_path.replace('fire-dataset', 'loc-fire-dataset')

save_and_display_gradcam(img_path=img_path, heatmap=heatmap)
```