

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Радіотехнічний факультет**

**Кафедра прикладної радіоелектроніки**

«На правах рукопису»

УДК \_\_\_\_\_

До захисту допущено:

В.о. зав. кафедри

\_\_\_\_\_ Андрій МОВЧАНЮК

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою**

**«Інтелектуальні технології радіоелектронної техніки»**

**за спеціальністю 172 «Телекомунікації та радіотехніка»**

**на тему: « Підвищення ефективності нейронних мереж за рахунок  
комбінування цифрових фільтрів»**

Виконав:

студент 2 курсу, групи РЕ-21мп  
Аврамишин Андрій Ігорович

\_\_\_\_\_

Керівник:



Доцент к.т.н. Кирпатенко І.М.

\_\_\_\_\_

Рецензент:

Доцент к.т.н. каф. РТС Неуймін О. С.

\_\_\_\_\_

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

Студент \_\_\_\_\_

Київ – 2024 року  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Радіотехнічний факультет  
Кафедра прикладної радіоелектроніки

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інтелектуальні технології радіоелектронної техніки»

ЗАТВЕРДЖУЮ

В.о.зав. кафедри

\_\_\_\_\_ Андрій МОВЧАНЮК

« » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студента**  
**Аврамишину Андрію Ігоровичу**

1. Тема дисертації «Підвищення ефективності нейронних мереж за рахунок комбінування цифрових фільтрів»  
науковий керівник дисертації доцент к.т.н. Кирпатенко І.М.  
затверджені наказом по університету від «09» листопада 2023 р. № 5206-с
2. Термін подання студентом дисертації 11 січня 2024 року.
3. **Об'єкт дослідження:** розпізнавання та класифікація об'єктів нейронною мережею.
4. **Вихідні дані:** навчена нейронна мережа для розпізнавання та класифікації об'єктів.
5. **Перелік завдань, які потрібно розробити:**
  1. Вивчити поширені методи підвищення ефективності нейромережі.
  2. Розглянути використання фільтрів ЦОС, які змінюють зображення.

3. Дослідити найкращу комбінацію фільтрів для попередньої обробки зображень.

**6. Орієнтовний перелік графічного (ілюстративного) матеріалу:** слайди, які відображають результати дослідження в рамках написання магістерської дисертації.

**7. Орієнтовний перелік публікацій:** 1 теза доповіді на науково-практичній конференції.

**9. Дата видачі завдання:** 01 вересня 2023 року.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Отримання теми магістерської дисертації	05.09.2023р.	Виконано
2	Розробка плану магістерської дисертації	10.09. 2023р.	Виконано
3	Початок збору інформації для дослідження	24.09.2023р.	Виконано
4	Аналіз наявних методів покращення ефективності нейромережі	12.10.2023р.	Виконано
5	Розгляд цифрових фільтрів для обробки зображення	27.10.2023р.	Виконано
6	Підготовка до проведення дослідження	09.11.2023р.	Виконано
7	Проведення порівняння комбінацій фільтрів	23.11.2023р.	Виконано
8	Аналіз результатів	08.12.2023р.	Виконано
9	Оформлення магістерської дисертації	09.01.2024р.	Виконано

Студент

Андрій Аврамишин

Науковий керівник

Ілля Кирпатенко

## РЕФЕРАТ

Застосування машинного навчання стає все більш важливим, оскільки кількість доступних даних продовжує зростати в геометричній прогресії. За останні десятиріччя вони стали ключовим інструментом в області розпізнавання об'єктів на зображеннях, проявивши високу ефективність та точність. Нейронні мережі знаходять застосування в різних галузях, включаючи медицину, виробництво, безпеку, роздрібну торгівлю та інші. Їхні можливості роблять їх універсальним інструментом для різноманітних викликів.

Об'єктом дослідження виступає цифрові фільтри та нейрона мережа. Метою дослідження є покращена ефективність розпізнавання об'єктів на зображенні нейромережею. Предметом дослідження є комбінація цифрових фільтрів для підвищення ефективності нейронної мережі. В даний момент існують різні методи покращення ефективності розпізнавання об'єктів нейромережею. В роботі розглянуті такі методи: налаштування гіперпараметрів, доданавання нових шарів у архітектуру нейромережі, відсіюваний шар, розширення даних, рання зупинка та перенос навчання. Розпізнавання об'єктів наразі найзатребуваніша галузь нейронної мережі і тому покращення ефективності навіть на декілька відсотків має глибокий вплив на її використання у різних сферах, полегшуючи інтеграцію та підвищуючи загальний внесок у вирішення реальних завдань.

В роботі пропонується використовувати комбінацію цифрових фільтрів. Це допоможе попередньо змінити зображення щоб нейромережа легше зчитала контури об'єкта на зображенні і дала якомога правильнішу класифікацію. Фільтри в першу чергу підібрані щоб прибирати шум та робити контури об'єкта більш чіткими.

Ключові слова: нейронна мережа, цифрові фільтри, розпізнавання, класифікація, обробка зображення, методи покращення.

## SUMMARY

Machine learning applications are becoming increasingly important as the amount of available data continues to grow exponentially. Over the past decades, they have become a key tool in the field of object recognition in images, showing high efficiency and accuracy. Neural networks are used in various industries, including medicine, manufacturing, security, retail, and others. Their capabilities make them a versatile tool for a variety of challenges.

The object of study is image processing using digital filters. The purpose of the study is to consider the combination of digital filters to optimize the operation of the neural network and increase its performance. The subject of the study is the combination of digital filters to improve the efficiency of a neural network. Currently, there are various methods for improving the efficiency of object recognition by a neural network. The following methods are considered in this paper: tuning hyperparameters, adding new layers to the neural network architecture, dropout layer, data expansion, early stopping, and transfer of training. Object recognition is currently the most popular area of neural networking, and therefore, improving the efficiency by even a few percent has a profound impact on its use in various fields, facilitating integration and increasing the overall contribution to solving real-world problems.

In this paper, we propose to use a combination of digital filters. This will help to pre-alter the image so that the neural network can more easily read the contours of the object in the image and give the most correct classification. The filters are primarily selected to remove noise and make the contours of the object clearer.

**Keywords:** neural network, digital filters, recognition, classification, image processing, improvement methods.

## ЗМІСТ

Перелік умовних позначень .....	8
Вступ.....	9
1.1 Огляд основних понять та технологій в області нейронних мереж .....	11
1.2 Методи покращення.....	14
1.3 Перспективи використання запропонованих методів у майбутньому .....	23
2.1 Опис видів цифрових фільтрів .....	26
2.2 Фільтр Гауса .....	26
2.3 Медіанний фільтр.....	30
2.4 Фільтр середнього значення .....	34
2.5 Фільтр Собеля.....	38
2.6 Висновок .....	41
3.1 Метод дослідження .....	42
3.2 Процес дослідження підвищення ефективності нейронної мережі за допомогою поєднання цифрових фільтрів .....	43
3.3 Аналіз проведеного дослідження .....	47
Стартап-проект .....	50
Джерела .....	52
Додаток А.....	55
Додаток Б .....	56
Додаток В.....	57
Додаток Г .....	58

Додаток Д.....	59
Додаток Ж.....	60

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

НМ — нейронна мережа

ЦОС — цифрова обробка сигналу

ШІ — штучний інтелект

ГН — глибинне навчання

ФНЧ — фільтр низьких частот

ФВЧ — фільтр високих частот



## ВСТУП

В епоху стрімкого технологічного розвитку і високих технологій наші життя стають зручнішими та безпечнішими завдяки автоматичному виявленню та класифікації об'єктів на зображеннях. Ця інноваційна технологія дозволяє розпізнавати та ідентифікувати об'єкти, що відкриває нові можливості у безпеці, медицині, сільському господарстві та ще багатьох галузях.

Щоб забезпечити високу точність та ефективність виявлення та класифікації зображення, ключовим є етап підготовки зображень шляхом фільтрації. Це необхідний крок для оптимізації зображень перед їх подальшим аналізом штучним інтелектом. Цей процес відіграє критичну роль у спрощенні завдання для алгоритмів виявлення об'єктів. Зображення можуть містити різний рід шуму, що походить від різних джерел, таких як перешкоди або недосконалості обладнання. Фільтрація допомагає зменшити або видалити цей шум, що робить зображення чіткішим та покращує точність виявлення об'єктів.

Об'єкт дослідження: розпізнавання та класифікація об'єктів нейронною мережею.

Предмет дослідження: комбінація цифрових фільтрів для підвищення ефективності нейронної мережі.

Мета дослідження: покращена ефективність розпізнавання об'єктів на зображенні нейромережею.

Під час дослідження були поставлені такі задачі:

- Вивчити поширені методи підвищення ефективності нейромережі.
- Розглянути використання фільтрів ЦОС, які змінюють зображення.
- Дослідити найкращу комбінацію фільтрів для попередньої обробки зображень.

Дослідження спрямоване на знаходження найбільш ефективної комбінації цифрових фільтрів для використання у нейронних мережах. У якості засобу

досягнення цієї мети розглядається поєднання фільтра Гауса, фільтра середнього значення, медіанного фільтра та фільтра Собеля. Цей розділ присвячено обґрунтуванню актуальності підвищення ефективності нейронних мереж у завданнях розпізнавання та класифікації об'єктів на зображеннях, визначенню об'єкта та предмету дослідження, а також висвітленню основних завдань та методів, які будуть розглянуті в наступних розділах.

# 1. ВИКОРИСТАННЯ НЕЙРОННОЇ МЕРЕЖІ В ОБРОБЦІ ЗОБРАЖЕННЯ

## 1.1 Огляд основних понять та технологій в області нейронних мереж

Нейронні мережі є одними з найпотужніших інструментів машинного навчання, які використовуються в широкому спектрі застосувань, включаючи обробку зображень та розпізнавання мовлення. Вони моделюють структуру та функціонування людського мозку та здатні навчатися на великих обсягах даних. Нейронні мережі мають ряд переваг.[1] Можуть обробляти великі обсяги складних даних, що робить їх ефективними в таких завданнях, як розпізнавання зображень, обробка природної мови та аналіз настроїв. Архітектура може бути налаштована та оптимізована для різних типів даних і проблемних областей, що дозволяє компаніям пристосовувати нейронні мережі до своїх конкретних потреб. Добре продумані архітектури можуть підвищити точність прогнозів або класифікацій, фіксуючи складні закономірності та взаємозв'язки в даних. Архітектури нейронних мереж можуть автоматично навчатися і витягувати відповідні ознаки з необроблених даних, усуваючи необхідність ручної розробки ознак. Однак також мають ряд обмежень, таких як необхідність великих обсягів даних для навчання та схильність до перенавчання.[2]

Нейронні мережі ідеально підходять для того, щоб допомогти людям вирішувати складні проблеми в реальних життєвих ситуаціях. Вони можуть вивчати і моделювати нелінійні та складні взаємозв'язки між входами і виходами, робити узагальнення і висновки, виявляти приховані взаємозв'язки, закономірності і прогнози, а також моделювати дуже мінливі дані (наприклад, фінансові часові ряди) і варіації, необхідні для прогнозування рідкісних подій (наприклад, виявлення шахрайства). Нейронні мережі можуть вивчати і моделювати зв'язки між вхідними і вихідними даними, які є нелінійними і складними. Завдяки цьому комп'ютери можуть приймати правильні рішення з мінімальною допомогою людини.[3]

Штучна нейронна мережа складається зі штучних нейронів, які працюють разом для вирішення проблеми. Штучні нейрони - це програмні модулі, які називаються вузлами, а штучні нейронні мережі - це програми або алгоритми, які в своїй основі використовують обчислювальні системи для вирішення математичних розрахунків. Базова нейронна мережа має з'єднання штучних нейронів у три шари:

### 1. Вхідний шар

Інформація із зовнішнього світу надходить до штучної нейронної мережі з вхідного шару. Вхідні вузли обробляють дані, аналізують або класифікують їх і передають на наступний рівень.

### 2. Прихований шар

Приховані шари отримують вхідні дані від вхідного шару або інших прихованих шарів. Штучні нейронні мережі можуть мати велику кількість прихованих шарів. Кожен прихований шар аналізує вихідні дані попереднього шару, обробляє їх далі і передає наступному шару.

### 3. Вихідний шар

Вихідний шар дає кінцевий результат всієї обробки даних штучною нейронною мережею. Він може мати один або декілька вузлів. Наприклад, якщо ми маємо проблему бінарної класифікації (так/ні), вихідний шар матиме один вихідний вузол, який дасть результат у вигляді 1 або 0. Однак, якщо ми маємо проблему багатокласової класифікації, вихідний шар може складатися з більш ніж одного вихідного вузла.

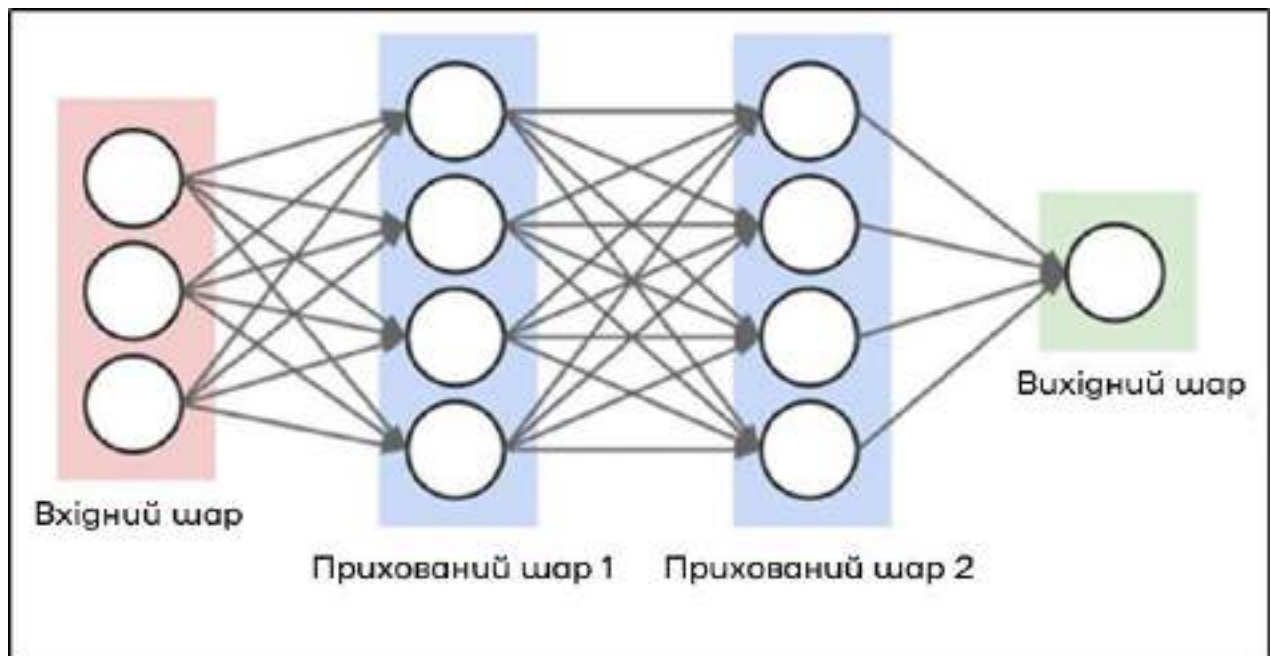


Рисунок 1.1 — Архітектура нейронної мережі

Глибокі нейронні мережі мають декілька прихованих шарів з мільйонами штучних нейронів, пов'язаних між собою. Число, яке називається вагою, відображає зв'язок між двома вузлами. Вага є додатнім числом, якщо один вузол збуджує інший, або від'ємним, якщо один вузол пригнічує інший. Вузли з більшою вагою мають більший вплив на інші вузли.[4]

Існують різні типи глибоких нейронних мереж - і кожна з них має свої переваги та недоліки, залежно від сфери застосування. Приклади включають:

1. Згорткові нейронні мережі (CNN) містять п'ять типів шарів: вхідний, згортковий, об'єднувальний, повністю з'єднаний і вихідний. Кожен шар має певне призначення, наприклад, підсумовувати, з'єднувати або активувати. Згорткові нейронні мережі популяризували класифікацію зображень і виявлення об'єктів. Однак CNN також застосовуються в інших сферах, таких як обробка природної мови та прогнозування.
2. Рекурентні нейронні мережі (RNN) використовують послідовну інформацію, таку як дані з позначкою часу з сенсорного пристрою або вимовлене

речення, що складається з послідовності термінів. На відміну від традиційних нейронних мереж, всі входи рекурентної нейронної мережі не є незалежними один від одного, а вихід кожного елемента залежить від обчислень попередніх елементів. ШНМ використовуються для прогнозування та обробки часових рядів, аналізу настроїв та інших текстових додатків.

3. Нейронні мережі прямого поширення, в яких кожен персептрон в одному шарі з'єднаний з кожним персептроном з наступного шару. Інформація передається від одного шару до іншого тільки в прямому напрямку. Зворотні зв'язки відсутні.
4. Автокодерні нейронні мережі використовуються для створення абстракцій, які називаються кодувальниками, що створюються на основі заданого набору входів. Хоча автокодери схожі на більш традиційні нейронні мережі, вони намагаються моделювати самі входи, і тому цей метод вважається некерованим. Передумовою автокодерів є десенсибілізація нерелевантних даних і сенсибілізація релевантних. По мірі додавання шарів, подальші абстракції формулюються на вищих шарах (шарах, найближчих до точки, в якій вводиться шар декодера). Ці абстракції потім можуть бути використані лінійними або нелінійними класифікаторами.

## 1.2 Методи покращення

### Налаштування гіперпараметрів

Оптимізація параметрів у моделі включає налаштування різних характеристик для покращення продуктивності. Ці параметри варіюються від кількості епох навчання до оптимізаторів та швидкості навчання. Гіперпараметри можуть мати різні значення, і кількість комбінацій може швидко досягти сотень. Налаштування гіперпараметрів — це мистецтво підібрати їх так, щоб ефективно створити

оптимальну модель.[12] Хоч ми можемо спробувати лише обмежену кількість параметрів, сервіс налаштування може визначити, які гіперпараметри слід залишити чи вилучити. Деякі алгоритми, такі як сітковий пошук у просторі гіперпараметрів, можуть автоматизувати цей процес. При виявленні гіперпараметра, що погіршує продуктивність, його можна виключити, а іноді вилучають і подібні гіперпараметри. Налаштування гіперпараметрів — завдання, що вимагає врахування різних вимог кожної конкретної задачі. Балансування десятків параметрів — це виклик, який допомагає підвищити точність класифікації зображень.[13]

Для пошуку оптимальних значень цих гіперпараметрів можна скористатися сітковим пошуком. Для цього спочатку потрібно визначити діапазон значень для кожного гіперпараметра, який треба налаштувати. Наприклад, можна взяти такі діапазони:

- Швидкість навчання: від 0,001 до 0,1.
- Кількість шарів: від 1 до 5
- Розмір вибірки: від 32 до 128

Далі створити сітку всіх можливих комбінацій цих значень:

- Швидкість навчання: 0.001, 0.01, 0.1
- Кількість шарів: 1, 3, 5
- Розмір вибірки: 32, 64, 128

Тоді сітка матиме 27(3x3x3) комбінацій. Потім потрібно навчити нейронну мережу, оцінити модель для кожної комбінації та записати результати. Наприклад, можна записати точність кожної моделі на валідаційному наборі. Далі обирати комбінацію, яка дає найкращі результати. Поєднання швидкості навчання 0,01, 3 шарів і розміру партії 64 дало найвищу точність. Ця комбінація буде оптимальною для гіперпараметрів.

### **Додавання нових шарів**

Додавання нових шарів до вашої моделі збільшує її здатність глибше вивчати особливості набору даних. Це означає, що вона зможе розпізнавати тонкі відмінності,

які людина могла б не помітити. Збільшення кількості шарів у нейронній мережі може підвищити її точність. Додавання нових шарів може допомогти нейронній мережі вивчити більш складні закономірності в даних, що може призвести до кращих прогнозів.[7] Традиційно нейронні мережі масштабуються шляхом збільшення кількості шарів або кількості нейронів у кожному шарі. Однак такий підхід має ряд обмежень. Зі збільшенням кількості шарів або нейронів модель стає складнішою, що може призвести до перенавчання та збільшення часу навчання. Крім того, збільшення кількості шарів або нейронів може також збільшити кількість параметрів, що може призвести до збільшення вимог до пам'яті і збільшення часу виведення.[7]

### Відсіюваний шар

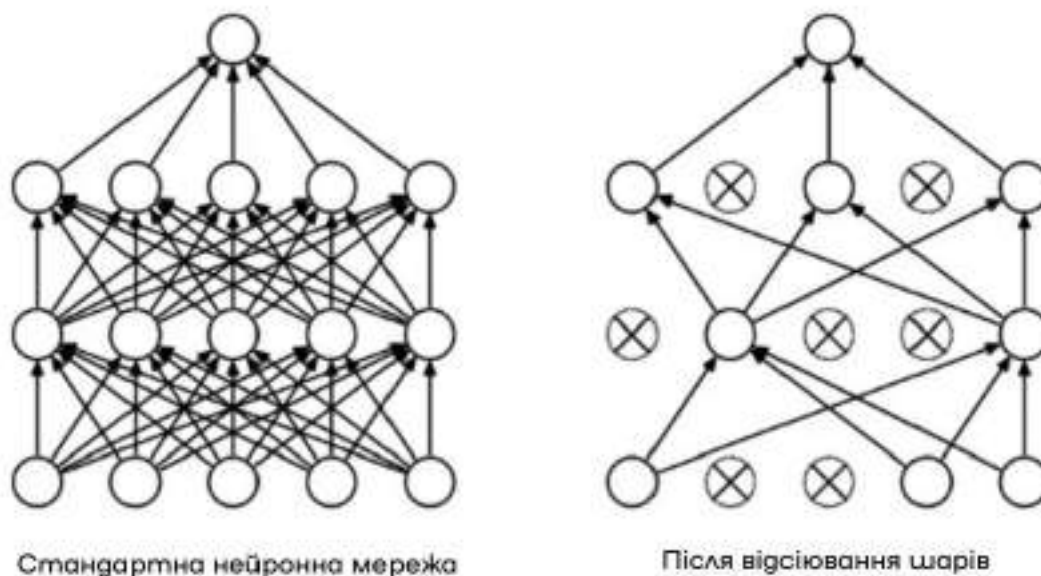


Рисунок 1.2 — Стандартна нейронна мережа та вона ж після відсіювання шарів

Відсіюваний шар представляє собою шар у нейронній мережі, який з визначеною ймовірністю анулює нейрони, іншими словами, ігнорує їх під час тренувального прогону. Це дозволяє знизити ризик перенавчання в глибоких нейронних мережах, оскільки нейрони не формують адаптаційних зв'язків між собою, а замість цього визначають глибші структури в навчальних даних. Під час



використання методу відсіву, певні вузли мережі встановлюються у нульове значення під час навчання, тобто вони виключаються з мережі. Відсів дозволяє вилучити їх вплив як на прогноз, так і на зворотне розповсюдження.[8] Внаслідок цього при кожному навчальному прогоні формується нова, трохи змінена архітектура мережі, і мережа навчається робити точні прогнози без певних вхідних даних. При застосуванні відсіву до шару також вказується ймовірність відсіву, яка визначає кількість вузлів у шарі, що будуть встановлені рівними 0. Наприклад, якщо вхідний шар має десять вхідних значень, ймовірність відсіву 10% означає, що один випадковий вхід буде встановлено рівним нулю під час кожного навчання. У випадку прихованих шарів ця логіка застосовується до прихованих вузлів. Таким чином, ймовірність відсіву 10% означає, що 10% вершин не будуть використані в кожному проході. Оптимальна ймовірність відсіву також сильно залежить від типу шару. З різних досліджень випливає, що для вхідного шару оптимальною є ймовірність відсіву, близька до одиниці. З іншого боку, для прихованих шарів оптимальна ймовірність, близька до 50%, призводить до кращих результатів.[8]

Однією з головних переваг відсіву є його здатність покращувати продуктивність узагальнення нейронних мереж. Перенавчання відбувається, коли модель стає занадто спеціалізованою у вивченні деталей навчальних даних, що призводить до поганої продуктивності на невидимих даних. Випадкове вимкнення нейронів під час навчання перешкоджає спільній адаптації нейронів і змушує мережу вивчати більш надійні та узагальнені ознаки. Така регуляризація допомагає моделі краще працювати з новими, ще не баченими даними, що призводить до покращення загальної продуктивності. Цей метод забезпечує ефективну техніку регуляризації без потреби в складній архітектурі. У традиційних методах регуляризації, таких як L1 і L2 регуляризація, введення додаткових параметрів або обмежень може стати обчислювально дорогим і складним у налаштуванні. З іншого боку, метод відсіювання відносно простий у застосуванні і вимагає мінімальних змін в існуючій мережевій архітектурі. Ця простота робить його

привабливим варіантом для покращення продуктивності моделей глибокого навчання без необхідності значних архітектурних модифікацій.[8]

Реалізувати відсіювання відносно просто, особливо в сучасних фреймворках глибокого навчання, таких як TensorFlow і PyTorch. Додавання шарів відсіву до архітектури нейронної мережі вимагає мінімальних зусиль, що дозволяє легко інтегрувати їх в існуючі моделі. Більше того, незважаючи на випадковість, що вноситься під час навчання, відсів не призводить до значного збільшення обчислювальних витрат порівняно з іншими методами регуляризації. Така ефективність дозволяє дослідникам і практикам експериментувати з відсівом без значних обчислювальних витрат.

Відсіювання може діяти як форма зменшення шуму під час навчання, роблячи модель більш стійкою до зашумлених або недосконалих навчальних даних. Випадкове відсіювання нейронів ефективно запобігає надмірному пристосуванню мережі до шуму, присутнього в даних. Така стійкість до зашумлених даних може бути особливо корисною в реальних умовах, коли якість даних може бути неоптимальною.

### **Розширення даних**

Щоб отримати з малого набору даних великий можна його штучно розширити. Процес доповнення даних включає повторну модифікацію заданих прикладів за допомогою різних методів з метою створення нових екземплярів для навчання на основі наявних зображень.[15] Такі методи доповнення даних для зображень включають в себе:

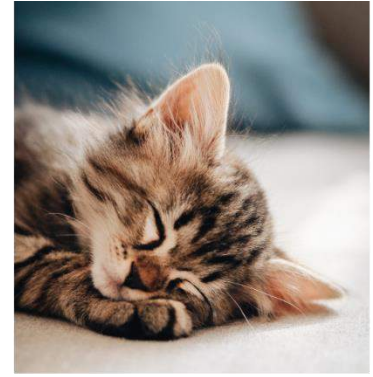
- перевертання в різні сторони;
- віддзеркалення;
- випадкові зміни розміру;
- випадкові обрізання;
- накладати та змішувати кілька зображень;
- зміни яскравості, контрастності або колірного каналу;
- зміна різкості або розмиття;
- видалення частини початкового зображення.



Початкове зображення



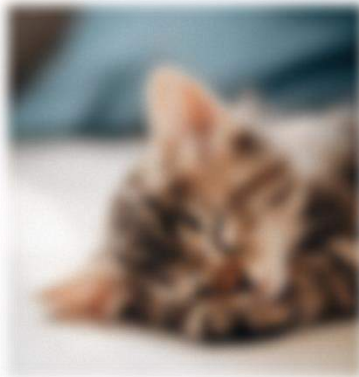
Перевернуте



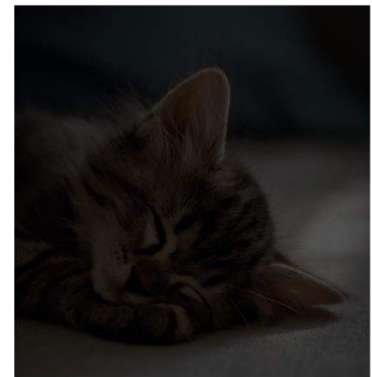
Віггзеркалене



Змінено розмір



Розмите



Змінено яскравість

Рисунок 1.3 — Методи зміни зображення

Це хороший метод для підвищення точності класифікації зображення. Доповнення даних володіє додатковою перевагою у вигляді регулятора, даючи дані моделі з різних перспектив. Його слід використовувати щоб підвищити точність моделі, запобігти перенавчанню, очистити необроблений набір даних та у випадках коли початковий набір даних занадто малий.[15] Використання методу розширення даних може бути розглянуто в контексті всіх застосувань машинного навчання, де отримання якісних даних виявляється трудомістким завданням. Безперечно, цей підхід також може сприяти підвищенню надійності та продуктивності моделі. Збирання та маркування наборів даних у медицині є витратним та тривалим процесом. Для перевірки даних потрібен експерт у цій галузі. Використання геометричних та інших трансформацій може сприяти створенню надійних та точних моделей машинного навчання. Наприклад, у випадку класифікації пневмонії можна

впроваджувати випадкове обрізання, масштабування, розтягування та трансформації кольорового простору для покращення ефективності моделі. Однак слід бути уважними з деякими видами доповнень, оскільки вони можуть призвести до небажаних результатів. Наприклад, для набору даних рентгенівських знімків не рекомендується застосовувати випадкове обертання та відображення вздовж осі x.

### Рання зупинка

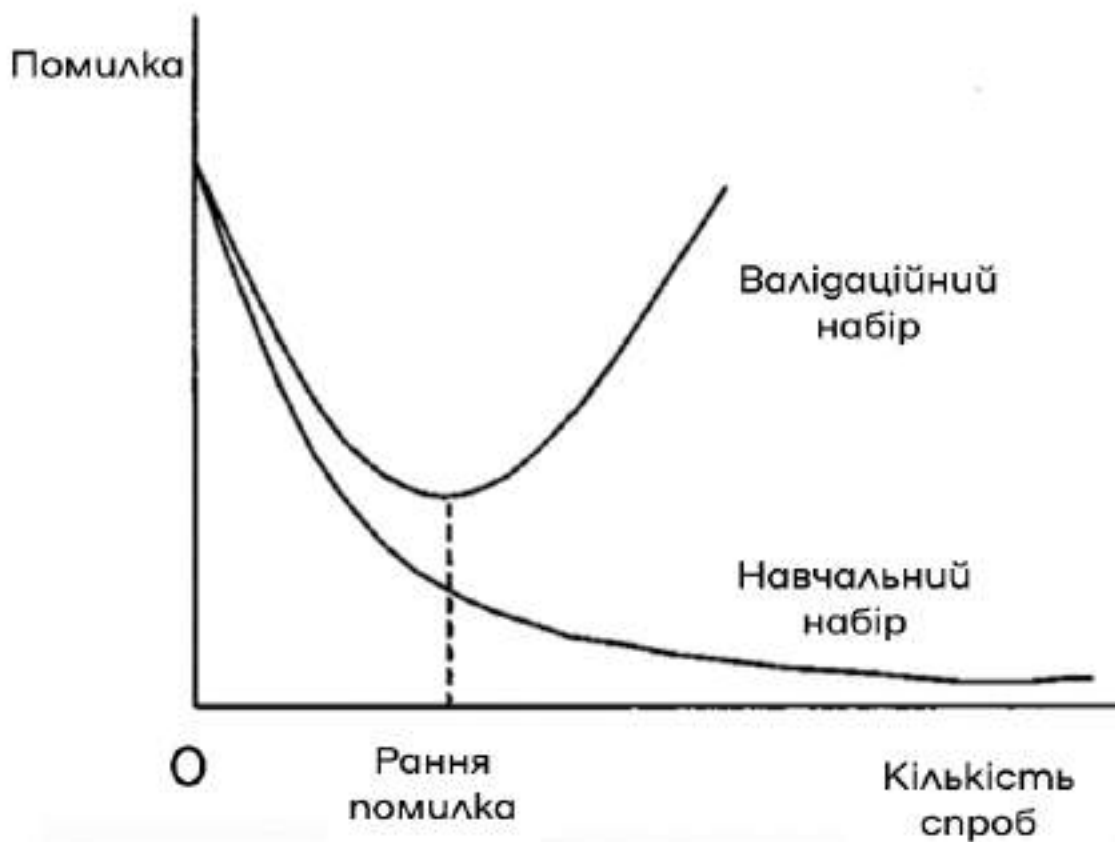


Рисунок 1.4 — Принцип роботи методу ранньої зупинки

При навчанні великої мережі настає момент, коли вона перестає узагальнювати і починає досліджувати статистичний шум у тренувальному наборі даних. Це явище, відоме як перенавчання, призводить до збільшення помилки узагальнення, роблячи модель менш ефективною для прогнозування нових даних. Завдання полягає в тому,

щоб тренувати мережу достатньо довго, щоб вона засвоїла відображення від входів до виходів, але не надто довго, щоб уникнути перенасичення навчальних даних.[5]

Один із способів вирішення цієї проблеми - розглядати кількість епох навчання як гіперпараметр, тренувати модель кілька разів з різними значеннями та обирати той варіант, що показує найкращу продуктивність на тренувальному або тестовому наборі даних. Однак недолік цього підходу полягає в тому, що він вимагає тренування та відкидання кількох моделей, що може бути неефективним з точки зору обчислювальних ресурсів та забирати значний час, особливо для великих моделей, що працюють з обширними наборами даних протягом тривалого періоду, такого як кілька днів або тижнів. Рання зупинка - це метод регуляризації, який покращує точність класифікації зображень, навмисно зупиняючи навчання, коли зростають втрати валідації.[5] Навчання зупиняється, оскільки тренування моделі на занадто великій кількості епох іноді призводить до перенавчання. У методі ранньої зупинки кількість епох стає гіперпараметром, що налаштовується. Ми безперервно зберігаємо найкращі параметри під час навчання, і коли ці параметри більше не змінюються протягом декількох епох, ми зупиняємо навчання. Цей метод настільки простий у використанні, наприклад, з найпростішим тригером, що немає жодних причин не використовувати його при навчанні нейронних мереж.[5]

Використання ранньої зупинки може стати фундаментом для сучасних підходів до тренування глибоких нейронних мереж. Аналізування графіків продуктивності моделі у режимі реального часу або після тривалого періоду навчання дозволить визначити, наскільки зашумленим є процес навчання для вашої конкретної моделі та набору даних. Це важливо для визначення оптимального тригера дострокової зупинки. Неодноразове повторення застосування ранньої зупинки призведе до надлишкового адаптування моделі до набору даних для валідації. Краще використовувати цей метод лише після визначення всіх інших гіперпараметрів моделі.

Або можна застосувати різні поділи навчального набору даних на навчальний та валідаційний набори кожного разу при використанні ранньої зупинки.[6]

### Перенос навчання

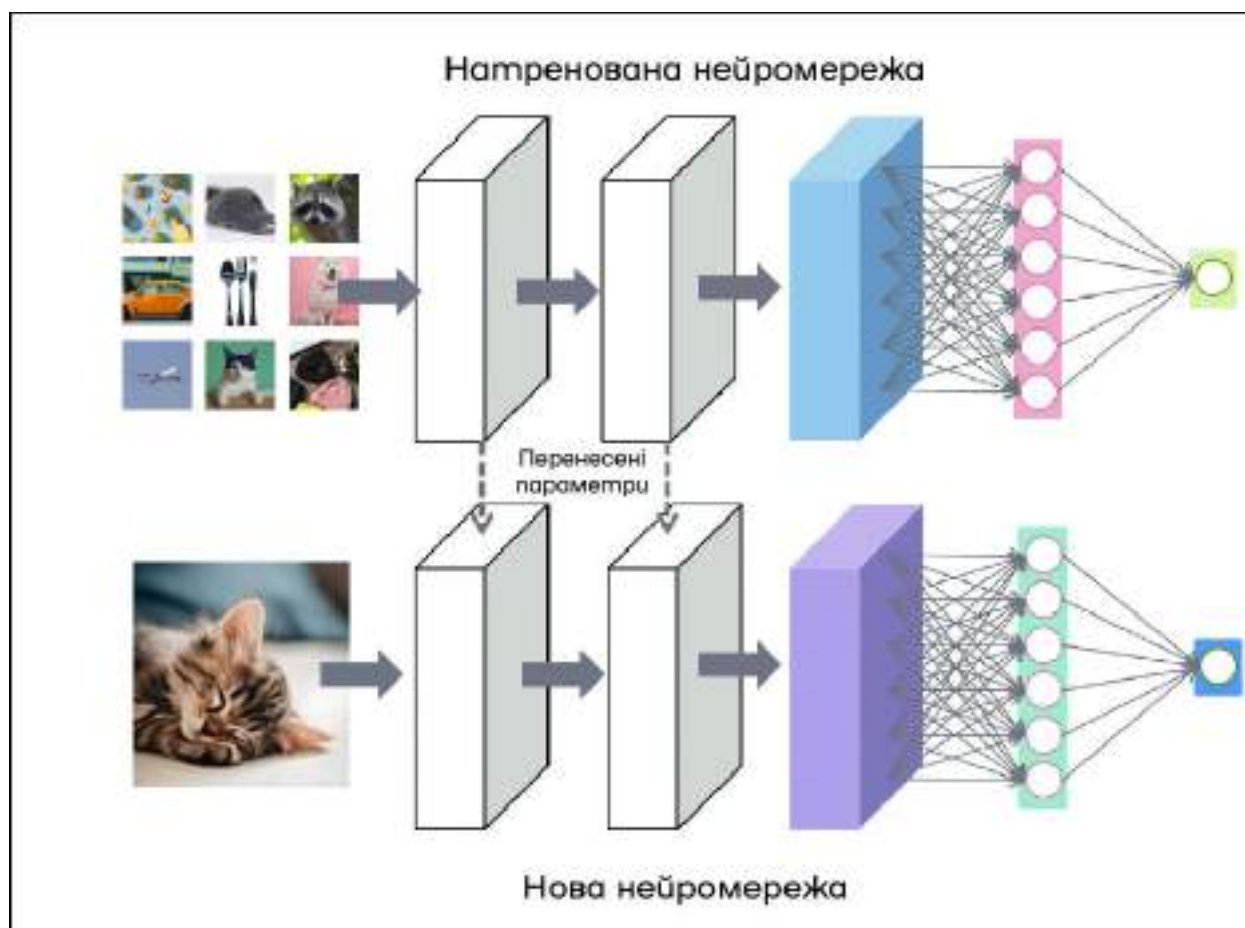


Рисунок 1.5 — Принцип роботи переносу навчання

Навчання масштабної нейронної мережі вимагає багато часу та сил. Тому щоб зекономити важливі ресурси, можна взяти вже навчену модель на великому наборі даних і використати лише потрібну частину. Вона вже вміє розпізнавати значну сукупність ознак об'єктів. Ця процедура дозволяє використовувати навчену модель для визначення ваг або виявлення ознак. Зазвичай перенос навчання застосовується для більш точного налаштування нейронної мережі. У випадку коли набору даних не вистачає для навчання моделі, це допоможе збільшити кількість даних. Великою перевагою цього методу є сильне скорочення часу навчання. У повторно взятих

шарах, ваги змінюються під нову конкретну задачу і можуть слугувати за початок навчання. Це називається ініціалізації ваг. Згорткові шари, які розташовані ближче до вхідного шару, спроможні вивчати низькорівневі ознаки, в той час як шари у середині моделі фокусуються на складних абстрактних ознаках, які об'єднують характеристики, витягнуті з більш низькорівневих шарів. Шари, що знаходяться ближче до виходу, інтерпретують об'єднані ознаки у контексті завдання класифікації. З врахуванням цього, можна визначити рівень деталізації для виділення ознак із попередньо навченої моделі. Наприклад, якщо нове завдання суттєво відрізняється від класифікації об'єктів на фотографіях, то буде доцільним виведення даних з попередньо навченої моделі після кількох перших шарів. У випадку, коли нове завдання схоже на класифікацію об'єктів на фотографіях, можна розглядати використання вихідних даних з шарів, що знаходяться на глибшому рівні у моделі, або навіть вихідних даних повністю пов'язаного шару перед вихідним шаром.[10]

Для виконання методу, до згорткової частини додають шар, який відповідає за класифікацію потрібних об'єктів. Потім іде навчання мережі на доступному набору даних. Попередньо навчену модель можна використовувати як самостійний інструмент для вилучення ознак.[11] Цей метод є невід'ємною частиною навчання для багатозадачних моделей. Найчастіше перенос навчання використовується для комп'ютерного зору або розпізнавання мови.

### **1.3 Перспективи використання запропонованих методів у майбутньому**

Запропоновані методи покращення ефективності нейронних мереж створюють значний потенціал для підвищення їхньої продуктивності та розширення областей їхнього використання. В першу чергу, розвиток оптимізованих структур та алгоритмів навчання відкриває можливості для створення більш ефективних та швидших нейронних мереж. Використання розглянутих методів може суттєво покращити їх області застосування у великому спектрі завдань, від розпізнавання образів до

автономних систем. Такий розвиток стане кроком вперед у досягненні більшої точності та надійності у використанні нейронних мереж у майбутніх технологічних вирішеннях.

Однією з ключових переваг Перенесення навчання є здатність моделі використовувати знання, накопичене на одному завданні, для покращення продуктивності на новому завданні. Це особливо корисно в умовах обмежених ресурсів, коли навчання нейронних мереж з нуля може вимагати великої кількості даних та обчислювальної потужності. Використання перенесення навчання відкриває можливість використання попередньо навчених моделей для швидкого та ефективного навчання на нових завданнях. Ще однією перспективою є розширення областей застосування нейронних мереж завдяки перенесення навчання. Відомо, що навчання на великих обсягах даних є важливим етапом у розвитку моделей глибокого навчання. Однак, застосовуючи перенесення навчання, можна досягти вражаючих результатів навіть у випадках обмежених наборів даних. Це робить нейронні мережі більш доступними та ефективними для різноманітних галузей, включаючи медицину, фінанси, та інші.

Попри досягнення у використанні рання зупинка, існує простір для подальших вдосконалень та досліджень. Однією з перспектив може бути розробка адаптивних методів рання зупинка, які змінюють своєрідність критеріїв зупинки під час різних етапів навчання. Також, важливо дослідити можливість комбінування рання зупинка із іншими методами регуляризації та оптимізації для досягнення ще кращих результатів.

Передбачається, що майбутні напрямки розвитку включатимуть більш продуктивні алгоритми налаштування гіперпараметрів, які забезпечать швидше та ефективніше налаштування гіперпараметрів. Також, зростання обчислювальної потужності дозволить більш детально досліджувати простори гіперпараметрів, що призведе до знаходження оптимальних конфігурацій для різних задач.



Очікується, що майбутні вдосконалення в області розширення даних включатимуть розвиток більш складних та інтелектуальних методів, зокрема використання глибокого навчання для створення більш реалістичних доповнених даних. Також, зростання обчислювальної потужності дозволить більш детально досліджувати і оптимізувати методи розширення даних для конкретних завдань. Ці інновації підкреслюють важливість та перспективи використання розширення даних у майбутньому для досягнення високої продуктивності нейромереж в умовах обмежених даних.

В майбутньому можна очікувати подальше розширення застосування відсіюваного шару в різних архітектурах нейронних мереж. Додаткові дослідження будуть спрямовані на оптимізацію параметрів відсіюваного шару, зокрема ймовірності вимкнення нейронів та моменту зрізу. Розуміння впливу відсіюваного шару на різні типи даних та завдань є важливим аспектом подальших досліджень. В цьому контексті, відсіюваний шар залишається актуальним та перспективним напрямком в розвитку нейромереж та машинного навчання загалом.

## **2. РОЛЬ ЦИФРОВИХ ФІЛЬТРІВ У ВИРШЕННІ ЗАВДАНЬ ОБРОБКИ ЗОБРАЖЕНЬ**

### **2.1 Опис видів цифрових фільтрів**

Фільтр низьких частот призначений для приглушення високочастотних сигналів та пропускання лише низькочастотних. У контексті обробки зображень, відмінні коливання частоти, як правило, спостерігаються на краях, що вказує на високий рівень шуму. Використання фільтра низьких частот дозволяє виділити низькочастотний сигнал зображення та приглушити високочастотний шум.[16] До прикладів таких фільтрів належать середній, медіанний та Гаусів фільтри. Згладжування зображення здійснюється зменшенням різниці між значеннями пікселів, що досягається усередненням сусідніх пікселів за допомогою ядра фільтра. Це дозволяє зберігати низькочастотну інформацію та зменшувати високочастотну.[17]

Фільтр високих частот, натомість, є основою для методів підвищення різкості зображень. Зберігання високочастотної інформації, відсікаючи низькочастотну, сприяє підвищенню контрастності між сусідніми областями з різницею у яскравості.[20] Ядро фільтра високих частот зазвичай призначене для підвищення яскравості центрального пікселя в порівнянні з його сусідами. Такі фільтри можуть бути пасивними чи активними, використовуючи різні компоненти для створення та підсилення сигналу. Вони широко використовуються для аналізу та класифікації текстур на зображеннях, виділяючи їхні особливості та структурні характеристики.[21]

### **2.2 Фільтр Гауса**

Фільтр Гауса - це один з найпоширеніших лінійних фільтрів для обробки зображень. Він базується на розподілі Гауса (нормальному розподілі) і використовується для згладжування та видалення шуму з зображень.[22] Фільтр Гауса діє, згідно з принципом конволюції, де для кожного пікселя вхідного зображення

обчислюється нове значення шляхом зваженого середнього значення пікселів у навколишньому середовищі, де ваги визначаються розподілом Гауса.[23] Розподіл Гауса є неперервним розподілом, що має щільність розподілу ймовірностей  $f(x)$ , виражену у вигляді:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$(\sigma > 0, -\infty < x < \infty)$$

де  $\mu$  - середнє значення, а  $\sigma$  - стандартне відхилення розподілу.

Відповідна функція розподілу ймовірностей  $F(x)$  представлена інтегралом:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}\left(\frac{u-\mu}{\sigma}\right)^2} du$$

$$(\sigma > 0, -\infty < x < \infty)$$

Результат обчислення заміщує вихідне значення цього пікселя. І це повторюється для всіх пікселів на зображенні. Фільтр Гауса розмиває потрібну область і відсікає шум з більш високими частотами. Це лінійний фільтр, який зменшує шум і ефективно розмиває краї.[23] Він створюється, як матриця при цифровій обробці зображень, проходячи через кожен піксель виділеної ділянки. Фільтр Гауса використовує ядро (фільтруючий масив), в якому центральний піксель має найбільшу вагу, а ваги пікселів зменшуються зі збільшенням відстані від центру ядра. Ця зміна ваги призводить до згладжування та зменшення впливу шуму. Цей розподіл також називається нормальним, а випадкова величина, що має такий розподіл, вважається нормально розподіленою.[24]

Оператор генерує шаблон значень, які потім застосовуються до груп пікселів на зображенні. Ці шаблонні значення визначаються 2D рівнянням Гауса:

$$\frac{1}{2\pi\sigma^2} \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\}$$

Сигма визначає ступінь розмиття:

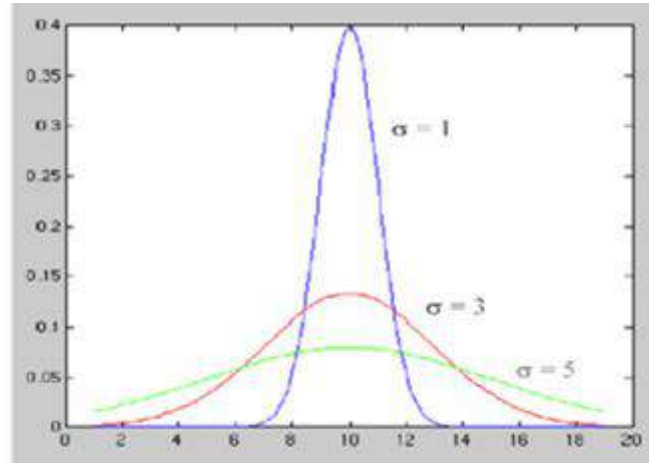


Рисунок 2.1 — Графік розподілу Гауса

Гаусова фільтрація є важливим інструментом в області обробки зображень, особливо для зменшення шуму. Застосовуючи гаусове ядро, фільтр надає центральним пікселям більшої ваги, ніж навколишнім областям, ефективно зменшуючи шум і зберігаючи структуру зображення. Важливий параметр  $\sigma$  контролює область застосування фільтра і подальше згладжування. Гаусові фільтри чудово видаляють випадкові, малопомітні шуми на зображенні, що робить їх життєво важливими у багатьох програмах обробки зображень.



Рисунок 2.2 — Застосування фільтра Гауса на зображенні. а) початкове зображення, б) ядро розміром 10x10, в) ядро розміром 20x20

Серед багатьох лінійних згладжувальних фільтрів, що використовуються для зменшення шуму, фільтр Гауса виділяється як найважливіший інструмент. Він використовує ваги, отримані з розподілу Гауса, для ефективного зменшення шуму на зображеннях. Однак перед його використанням важливо попередньо обробити зображення і видалити всі викиди або піки. Хоча він добре справляється з випадковим шумом, та все ж має свої обмеження. Він має тенденцію змішувати шум з результатом, спричиняючи невибіркве згладжування по краях.

<b>1</b>	<b>4</b>	<b>7</b>	<b>4</b>	<b>1</b>
<b>4</b>	<b>16</b>	<b>26</b>	<b>16</b>	<b>4</b>
<b>7</b>	<b>26</b>	<b>41</b>	<b>26</b>	<b>7</b>
<b>4</b>	<b>16</b>	<b>26</b>	<b>16</b>	<b>4</b>
<b>1</b>	<b>4</b>	<b>7</b>	<b>4</b>	<b>1</b>

Рисунок 2.3 — Ядро розміром 5x5

Отже, фільтр Гауса є популярним лінійним фільтром для зменшення шуму при обробці зображень. У поєднанні з іншими методами фільтрації зображень фільтр Гауса стає цінним інструментом, що покращує якість зображення і дає візуально приємні результати.

Застосовано фільтр Гауса на зображенні перед скануванням штучним інтелектом.

Зображено tiger із ймовірністю 74.91912189550615 %  
Зображено tiger із ймовірністю 75.03061212341882 %  
Зображено tiger із ймовірністю 74.36124765561162 %  
Зображено man із ймовірністю 74.62294523678256 %  
Зображено man із ймовірністю 74.58556915734938 %  
Зображено man із ймовірністю 75.65434626488704 %  
Зображено car із ймовірністю 75.42550282250559 %  
Зображено car із ймовірністю 74.75818720940683 %  
Зображено car із ймовірністю 74.1184270253627 %

Рисунок 2.4 — Класифікація об'єкта на зображенні із використанням фільтра Гауса

Після попередньої обробки зображення фільтром Гауса ефективність нейронної мережі зросла в середньому на 2.52%. Це вже непоганий результат, але можна використовуючи два різних фільтра можна добитися більшого зростання ефективності.

### 2.3 Медіанний фільтр

Медіанний фільтр відмінно справляється із видаленням шуму, оскільки він ігнорує викиди або надмірно великі або малі значення пікселів. Цей фільтр також добре підтримує границі об'єктів та деталі на зображенні, оскільки він не розмиває їх, як це може бути при застосуванні інших фільтрів.[26] Медіанний фільтр діє, шукаючи медіану пікселів в області, що оточує центральний піксель зображення.

Медіанний фільтр чудово справляється з шумом “сіть та перець”. Алгоритм фільтрації сканує все зображення, використовуючи невелику матрицю (наприклад, 3x3), і перераховує значення центрального пікселя, просто беручи медіану всіх значень всередині матриці.

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Сусідні значення:

115, 119, 120, 123, 124,  
125, 126, 127, 150

Медіанне значення: 124

Рисунок 2.5 Принцип роботи Медіанного фільтра

Основна ідея полягає в тому, що медіана є статистичною мірою, яка менше схильна до впливу викидів та шуму, порівняно з середнім значенням. Коли медіанний фільтр застосовується до пікселя, він упорядковує значення пікселів у вказаній області за зростанням та вибирає середнє значення.[27] Це значення стає новим значенням центрального пікселя. В результаті, значення, що виходять за рамки норми або вважаються шумом, не мають значення для медіанного фільтра та ігноруються. Він допомагає зменшити вплив шуму на результати виявлення та робить зображення більш придатними для подальшого аналізу. Таким чином, він видаляє деякі піки, спричинені шумом, особливо імпульсний шум. Це відбувається тому, що окремі пікселі шуму з екстремальною інтенсивністю, наприклад, чорно-білі, не виживають після медіанної фільтрації. Ще однією перевагою медіанного фільтра є те, що він не створює нових значень пікселів, оскільки повторно використовує лише існуючі значення пікселів у вікні.[28] Крім того, на відміну від інших фільтрів, він видаляє шум без втрати крайової інформації, як показано нижче. Останнє зауваження щодо медіанного фільтра полягає в тому, що його можна використовувати, коли гаусівський

фільтр не може вирішити проблему. На рис. показано візуальне порівняння між гаусовим та медіанним фільтрами.



Рисунок 2.6 — Застосування медіанного фільтра на зображенні. а) початкове зображення, б) ядро розміром 14x14, в) ядро розміром 21x21

Видно, що коли рівень шуму занадто високий, хоча кількість шумових пікселів зменшується зі збільшенням розміру фільтра Гауса, вони все ще існують на зображенні. Медіанний фільтр, з іншого боку, вже видаляє більшість шумових пікселів при розмірі фільтра 3 x 3. Застосовуючи більший розмір фільтра, медіанний фільтр ще більше виключає пікселі шуму, але при цьому втрачає багато інформації про структуру зображення та деталі зображення. Обчислюючи медіанне значення околиці, а не середнє, медіанний фільтр має дві основні переваги над середнім фільтром:

1. Медіана є більш надійним середнім значенням, ніж середнє, і тому один дуже нерепрезентативний піксель в околі не вплине на значення медіани суттєво.
2. Оскільки медіанне значення насправді має бути значенням одного з пікселів в околиці, медіанний фільтр не створює нових нереалістичних значень пікселів, коли фільтр перетинає край. З цієї причини медіанний фільтр набагато краще зберігає різкі краї, ніж середній фільтр.

Медіанний фільтр є потужним інструментом у фільтрації зображень при виявленні об'єктів. Він видаляє шум, робить точнішим контур об'єктів та поліпшує якість



зображень. Застосування медіанного фільтра є важливим кроком у точному та надійному виявленні об'єктів на зображеннях.

Особливість Медіанного фільтра полягає в тому, що значення центрального пікселя буде замінено значенням, яке присутнє в навколишніх пікселях.[28] Це відрізняється від фільтра Гауса, який використовує середнє значення, де викиди можуть сильно спотворити середнє значення, що призводить до майже повної відсутності зменшення шуму

Він дуже добре зберігає краї зображення. Це дуже важливо для комп'ютерного зору, оскільки краї важливі для таких речей, як розрізнення людини і фону у відео або визначення меж смуги руху для безпілотного автомобіля.

Оскільки фільтр Гауса бере середнє значення навколо пікселя, після сканування всіх пікселів зображення кожен з них стає сумішшю всіх кольорів навколо нього, і в кінцевому підсумку розмиває краї. Оскільки медіанна фільтрація використовує медіану, пікселі на краю об'єкта на зображенні стають значеннями, які вже присутні на цій ділянці зображення.

Медіанний фільтр залишає чіткий поділ між областями зображення, тоді як фільтр Гауса більше розмиває. Оптимізація виявлення країв є важливою концепцією для обробки зображень і комп'ютерного зору, навіть якщо оптимізації здаються незначними.

Перед скануванням штучним інтелектом зображення піддалося медіанній фільтрації.

Зображено tiger із ймовірністю 75.65972441526424 %

Зображено tiger із ймовірністю 75.69983191894065 %

Зображено tiger із ймовірністю 74.88782246591434 %

Зображено man із ймовірністю 75.11369020627123 %

Зображено man із ймовірністю 74.8693687938397 %

Зображено man із ймовірністю 74.61236841113514 %

Зображено car із ймовірністю 74.67698153407282 %

Зображено car із ймовірністю 74.02152974812945 %

Зображено car із ймовірністю 75.66694157717433 %

Рисунок 2.7 — Класифікація об'єкта на зображенні із використанням медіанного фільтра

В результаті попередньої обробки зображення цим фільтром, середня ефективність нейронної мережі зросла на 2.71%. Результат є задовільним, але використання двох різних фільтрів може призвести до ще значного підвищення продуктивності.

## 2.4 Фільтр середнього значення

Фільтрація за середнім значенням (або фільтр середнього значення) - це один із типів фільтрів, які використовуються в цифровій обробці сигналів і обробці зображень.[29] Ідея фільтрації за середнім значенням полягає в тому, щоб замінити значення кожного пікселя на зображенні середнім значенням його сусідів, включно з ним самим.[30] Це призводить до усунення значень пікселів, які не є репрезентативними для їхнього оточення. Фільтрація за середнім значенням зазвичай вважається фільтром згортки. Як і інші згортки, він ґрунтується на ядрі, яке представляє форму і розмір області, з якої буде зроблено вибірку при обчисленні середнього значення. Часто використовується квадратне ядро  $3 \times 3$ , як показано на рисунку 2.5, хоча для більш серйозного згладжування можна використовувати ядра більшого розміру (наприклад,  $5 \times 5$  квадратів).

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

**3x3**

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

**5x5**

1/n <sup>2</sup>	1/n <sup>2</sup>	—	1/n <sup>2</sup>
1/n <sup>2</sup>	1/n <sup>2</sup>	—	1/n <sup>2</sup>
1/n <sup>2</sup>	1/n <sup>2</sup>	—	1/n <sup>2</sup>
1/n <sup>2</sup>	1/n <sup>2</sup>	—	1/n <sup>2</sup>

**nхn**

Рисунок 2.8 — Приклади квадратного ядра для різної інтенсивності згладжування

Один піксель з дуже нерепрезентативним значенням може суттєво вплинути на середнє значення всіх пікселів у його околиці. Коли область фільтрації перетинає край, фільтр інтерполює нові значення для пікселів на краю і таким чином розмиває цей край. Це може бути проблемою, якщо на виході потрібно отримати чіткі краї.

Середня фільтрація - це широко застосовувана техніка в галузі обробки зображень, яка слугує важливим інструментом для зменшення шуму та згладжування. Використовуючи принципи просторової фільтрації, середня фільтрація допомагає усунути випадкові варіації або шум, присутні на зображенні, зберігаючи при цьому його основні характеристики.[30] У цьому контексті ми розглянемо механіку, переваги та обмеження усередненої фільтрації.

Фундаментальна концепція фільтрації за середнім значенням елегантно проста: кожен піксель зображення замінюється середнім значенням, отриманим із сусідніх пікселів, обмежених певним вікном або ядром. Розмір цього ядра визначає ступінь згладжування, причому більші ядра спричиняють потужніші ефекти згладжування за потенційну ціну втрати дрібних деталей.[31]



Рисунок 2.9 — Застосування фільтра середнього значення на зображенні. а) початкове зображення, б) ядро розміром 20x20, в) ядро розміром 30x30

По суті, середня фільтрація передбачає заміну значення кожного пікселя середнім значенням його сусідніх пікселів, включаючи сам піксель. Цей процес виявляється ефективним для видалення значень пікселів, які значно відрізняються від їхнього оточення. Концептуально схожа на згортку, фільтрація середнього значення обертається навколо ядра, яке визначає форму і розмір області, що охоплюється для обчислення середнього значення. Хоча часто використовується стандартне квадратне ядро  $3 \times 3$ , використання більших ядер - наприклад,  $5 \times 5$  квадратів - може інтенсифікувати згладжування для суттєвішого зменшення шуму.

Існує кілька ітерацій згладжування середнього значення, серед яких - порогове усереднення. Тут згладжування залежить від умови: значення центрального пікселя змінюється лише тоді, коли різниця між його початковим значенням і розрахованим середнім перевищує заданий поріг. Такий підхід ефективно зменшує шум, зберігаючи при цьому більше деталей зображення, ніж традиційна фільтрація за середнім значенням.

З точки зору простоти обчислень, фільтрація за середнім значенням є взірцем. Його проста реалізація та ефективність роблять його легкодоступним

інструментом.[29] Однак, середня фільтрація не є універсальною; зображення з гострими краями або складними деталями можуть зазнати розмиття країв через процедуру усереднення. Особливо ефективна проти гаусівського шуму або шуму "сіль-перець", середня фільтрація використовує контекст навколишніх пікселів, щоб зменшити випадкові значення шуму.

Хоча середня фільтрація займає своє місце в попередній обробці зображень для подальших складних завдань, таких як виявлення країв або розпізнавання об'єктів, її важливість підкреслюється в процесах відновлення та покращення зображень. Зокрема, в таких галузях, як медична візуалізація, застосування середньої фільтрації сприяє підвищенню точності діагностики та покращенню візуальної якості.

Таким чином, середня фільтрація є нескладним, але потужним методом обробки зображень. Його майстерність у зменшенні шуму і згладжуванні текстури робить його універсальним для різних застосувань. Тим не менш, розумний вибір розміру ядра залишається ключовим для досягнення балансу між зменшенням шуму і збереженням життєво важливих характеристик зображення. При розумному застосуванні середня фільтрація значно підвищує якість зображення і полегшує його всебічний аналіз. Незважаючи на його обчислювальну ефективність, необхідно визнати його обмеження; альтернативні згорткові фільтри, такі як фільтр Гауса, пропонують чіткі компроміси між зменшенням шуму і збереженням деталей.

Перед скануванням штучним інтелектом на зображення було застосовано фільтр середнього значення. На рисунку 2.10 показано результати розпізнавання та класифікації об'єктів.

Зображено tiger із ймовірністю 74.64999679862575 %

Зображено tiger із ймовірністю 74.85500524633309 %

Зображено tiger із ймовірністю 75.28633838675202 %

Зображено man із ймовірністю 75.71672773510777 %

Зображено man із ймовірністю 75.93508899053965 %

Зображено man із ймовірністю 74.80186737696926 %  
 Зображено car із ймовірністю 74.74823051568707 %  
 Зображено car із ймовірністю 74.42453041949229 %  
 Зображено car із ймовірністю 74.06220083950183 %

Рисунок 2.10 — Класифікація об'єкта на зображенні із використанням фільтра середнього значення

Після попередньої обробки зображення фільтром Собеля вдалося досягти середнього підвищення ефективності нейронної мережі на 2.63%. Результат достатній, проте використання двох різних фільтрів може призвести до подальшого збільшення ефективності.

## 2.5 Фільтр Собеля

Фільтр Собеля, винайдений Ірвіном Собельом і Гарі М. Фельдманом у 1968 році, є важливим інструментом в обробці зображень і комп'ютерному зорі. Його основне призначення полягає у виявленні країв зображень, відіграючи фундаментальну роль у різних алгоритмах виявлення країв. Покращуючи краї зображення, фільтр Собеля робить значний внесок у вирішення таких завдань, як виявлення об'єктів, сегментація зображень та покращення країв. Оператор Собеля - це класичний оператор виявлення країв першого порядку, який знаходить контраст за допомогою процесу, схожого на диференціювання.[32] Тут ми визначаємо величину ребер шляхом згортки двох шаблонів 3x3 із зображенням рівня сірого. Фільтр обчислює не лише величину ребер, але й їхній напрямок. Суть оператора Собеля полягає в його здатності обчислювати градієнт функції інтенсивності в кожній дискретній точці зображення. За своєю суттю, фільтр Собеля геніально розроблений як компактний фільтр з цілочисельним значенням, який працює як у горизонтальному, так і у вертикальному напрямках зображення.[32] Оператор використовує шаблон 3x3 по горизонталі, а потім по вертикалі:

-1	0	1	1	2	1
-2	0	2	0	0	0
-1	0	1	-1	-2	-1

Рисунок 2.11 — Вертикальна та горизонтальна маски оператора Собеля

В отриманому зображенні використовується метод піфагора для створення величини ребер та інверсний тангенс для отримання напрямку ребер.

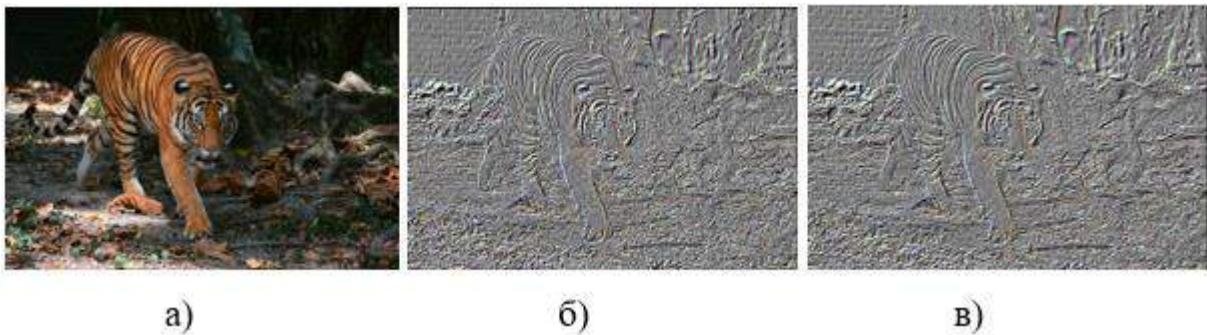


Рисунок 2.12 — Застосування фільтра Собеля на зображенні. а) початкове зображення, б) ядро розміром 5x5, в) ядро розміром 7x7

Ці ядра згортки відіграють фундаментальну роль у процесі згортки зображення. Коли ядра проходять зображення піксель за пікселем, вони виконують множення між значеннями пікселів і відповідними коефіцієнтами ядра.[34] Отримані результати потім підсумовуються, отримуючи значення відфільтрованого пікселя. Горизонтальне ядро Собеля найкраще ідентифікує вертикальні краї, тоді як вертикальне ядро Собеля вправно виділяє горизонтальні краї. Після завершення згорток фільтр Собеля обчислює величину градієнта, використовуючи теорему Піфагора:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Обчислення амплітуди дає уявлення про швидкість зміни інтенсивності та наявність країв по всьому зображенню. Більше того, орієнтація цих країв визначається проти годинникової стрілки відносно напрямку максимального контрасту, що простягається від чорного до білого. Використання тільки фільтра Собеля не дасть

чудових результатів, оскільки на зображенні можуть бути шуми. Але встановлення порогового значення на виході може покращити результат.

Коли ми застосовуємо цю маску до зображення, вона виділяє вертикальні краї. Вона просто працює як похідна першого порядку і обчислює різницю інтенсивностей пікселів в області країв. Оскільки центральний стовпчик дорівнює нулю, він не містить початкових значень зображення, а обчислює різницю значень правого та лівого пікселів навколо цього краю. Також центральні значення першого та третього стовпчиків дорівнюють 2 та -2 відповідно. Це надає більшого вагового віку значенням пікселів навколо області краю. Це збільшує інтенсивність країв, і вони стають виразнішими порівняно з оригінальним зображенням.

Таким чином, фільтр Собеля відіграє основоположну роль в обробці зображень і комп'ютерному зорі. Використовуючи можливості фільтра Собеля, фахівці можуть заглибитися в розуміння структури зображення і витягти безцінну інформацію з візуальних даних.

Застосування фільтру Собеля до зображення для попередньої обробки зображення:

```
Зображено tiger із ймовірністю 75.06411836681222 %
Зображено tiger із ймовірністю 74.06559690233587 %
Зображено tiger із ймовірністю 75.73174134835325 %
Зображено man із ймовірністю 74.39528323508497 %
Зображено man із ймовірністю 75.36962418966955 %
Зображено man із ймовірністю 74.9738431401081 %
Зображено car із ймовірністю 74.70055413592634 %
Зображено car із ймовірністю 74.96656404685422 %
Зображено car із ймовірністю 74.03335201728272 %
```

Рисунок 2.13 — Класифікація об'єкта на зображенні із використанням фільтра

Собеля



Використання фільтру Собеля для попередньої обробки зображення перед подачею їх на нейронну мережу призвело до збільшення ефективності на 2.5%. Однак, використання двох різних фільтрів може виявитися більш ефективним у порівнянні з використанням лише одного.

## **2.6 Висновок**

Розглянуто такі методи, як фільтр Собеля, фільтр Гауса, медіанний фільтр та фільтр середнього значення. Ці методи служать різним цілям, від зменшення шуму і згладжування зображень до виявлення та покращення країв. Фільтр Собеля став критично важливим інструментом для виявлення країв, ефективно виділяючи краї зображення шляхом згортки зображення з горизонтальними та вертикальними ядрами. З іншого боку, фільтр Гауса чудово справлявся з шумозаглушенням завдяки своєму підходу зваженого усереднення на основі розподілу Гауса. Середній фільтр пропонує просте, але ефективне рішення для зменшення шуму, замінюючи кожен піксель середнім значенням його сусідів.

### **3. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ НЕЙРОННИХ МЕРЕЖ З КОМБІНАЦІЯМИ ЦИФРОВИХ ФІЛЬТРІВ**

#### **3.1 Метод дослідження**

Для написання нейронних мереж використовують різні мови програмування, і вибір залежить від конкретного випадку та особистих вподобань. Наприклад у Python є багато бібліотек, таких як TensorFlow, PyTorch, Keras, які спрощують розробку та навчання моделей. Для використання нейромережі в браузері без додаткових додатків зручно використовувати JavaScript. Для оптимізації та використання нейронних мереж у вбудованих системах, іграх або додатках з високою продуктивністю зручна мова C++. Java підійде для нейронних мереж великого масштабу, особливо у сфері підприємництва. Дослідницькі та статистичні аналітики використовують R для роботи з нейронними мережами. Matlab також має інструменти для нейромереж та глибокого навчання.

Проаналізувавши доступні мови програмування було обрано Python через ряд його переваг. Зручність та лаконічний синтаксис. Відсутність необхідності в фігурних дужках для визначення блоків коду полегшує читання та підтримку кодової бази. Змінні найменування мають зрозумілу семантику, що полегшує розуміння призначення кожної змінної. Крім того, Python використовує відступи як частину свого синтаксису, що робить код більш читабельним. Легкість читання коду прискорює процес розробки, тестування та підтримки штучного інтелекту. Дослідження нейронної мережі на основі мови програмування Python дозволяє вивчати її ефективність у розпізнаванні об'єктів на фотографіях в різних умовах. Для цього проводяться експерименти з трьома типами фотографій: оригінальними, зашумленими та розмитими. Результати дослідження можуть надати інформацію про те, наскільки надійно та стійко працює нейронна мережа в різних умовах.

Бібліотеки разом із розширеннями та допоміжними інструментами роблять Python потужним середовищем для розробки та застосування нейронних мереж у різних областях. Однією з найвідоміших і широко використовуваних бібліотек для глибокого навчання є TensorFlow. Розроблена компанією Google, ця бібліотека надає високорівневі та низькорівневі інтерфейси для розробки та навчання нейронних мереж. TensorFlow дозволяє легко створювати складні моделі та ефективно їх оптимізувати. PyTorch використовує динамічне визначення графа, що дозволяє розробникам швидко маніпулювати структурою моделі та відстежувати зміни. Keras є високорівневим інтерфейсом для роботи з TensorFlow, а також може використовуватися з іншими бекендами. Завдяки простоті використання, Keras став популярним вибором для початківців та швидкої розробки прототипів моделей. MXNet є ще однією бібліотекою для глибокого навчання, яка надає гнучкість та ефективність. Вона підтримує гнучке розподілене навчання та використовує гібридний фреймворк для оптимізації продуктивності.

Для проведення дослідження обрано Keras в якості високорівневого інтерфейсу поверх TensorFlow. Keras надає чіткий та інтуїтивний API для роботи з різними типами шарів та моделей.[35] Це дозволяє легко змінювати архітектуру вашої нейромережі, додаючи або видаляючи шари, змінюючи їх параметри, а також використовуючи готові шаблони для створення різних типів моделей. Він також дозволяє конструювати складні архітектури нейромереж. Можна створювати моделі з кількома входами і виходами, використовувати спільні шари та поділ внутрішніх обчислень. Це робить Keras потужним інструментом для високорівневого моделювання різних архітектур.

### **3.2 Дослідження підвищення ефективності нейронної мережі за допомогою поєднання цифрових фільтрів**

Основною метою даного дослідження є аналіз впливу застосування різних комбінацій фільтрів (фільтрів Гауса, середнього значення, медіанного фільтра та фільтра Собеля) на якість розпізнавання об'єктів за допомогою нейронної мережі. Досліджуються різні сценарії комбінування цих фільтрів для зменшення впливу шуму та розмитості на вхідні дані.

Після проходження кожної фотографії через нейронну мережу розраховується ймовірність правильного розпізнавання об'єктів на кожному зображенні. Аналіз результатів надає можливість визначити, наскільки добре модель пристосовується до різних умов та якість її роботи при різних типах вхідних даних. Результати експериментів аналізуються шляхом порівняння покращень у якості розпізнавання об'єктів на фотографіях з використанням різних комбінацій фільтрів.

Для оцінки ефективності моделі, обирається три фотографії з різних категорій. Перші - оригінальні, для оцінки базової здатності розпізнавання. Другі - зашумлені, для визначення стійкості до випадкових артефактів та завад. Треті - розмиті, для вивчення можливості мережі працювати з нечіткими зображеннями.



Рисунок.3.1 — Зображення тигра: а) початкове зображення; б) зашумлене зображення; в) розмите зображення



Рисунок 3.2 — Зображення людини: а) початкове зображення; б) зашумлене зображення; в) розмите зображення



Рисунок 3.3 — Зображення автомобіля: а) початкове зображення; б) зашумлене зображення; в) розмите зображення

Дослідження проводилося шляхом поєднання двох фільтрів та перевірки. Із 4 фільтрів було сформовано 6 можливих пар:

1. Фільтр Гауса та фільтр середнього значення;
2. Фільтр Гауса та медіанний фільтр;
3. Фільтр Гауса та фільтр Собеля;
4. Фільтр середнього значення та медіанний фільтр;
5. Фільтр середнього значення та фільтр Собеля;
6. Фільтр Собеля та медіанний фільтр.

Перед перевіркою комбінацій, нейронній мережі було надано зображення для розпізнавання об'єктів.

На рисунку 3.2.1 нейромережа розпізнала тигра на 73.45%.

На рисунку 3.2.2 нейромережа розпізнала тигра на 71.42%.

На рисунку 3.2.3 нейромережа розпізнала тигра на 72.87%.

На рисунку 3.2.4 нейромережа розпізнала людину на 72.89%.

На рисунку 3.2.5 нейромережа розпізнала людину на 71.15%.

На рисунку 3.2.6 нейромережа розпізнала людину на 70.58%.

На рисунку 3.2.7 нейромережа розпізнала автомобіль на 72.34%.

На рисунку 3.2.8 нейромережа розпізнала автомобіль на 73.76%.

На рисунку 3.2.9 нейромережа розпізнала автомобіль на 72.23%.

Потім нейронна мережа сканувала зображення з використанням усіх комбінацій окремо.

Таблиця 1 — Результати досліджень

	1	2	3	4	5	6	7	8	9
Комбінація №1	78.2%	81.23%	80.43%	79.67%	80.41%	81.99%	78.13%	80.74%	80.92%
Комбінація №2	79.14%	80.68%	81.12%	80.21%	81.14%	80.76%	79.54%	81.69%	81.34%
Комбінація №3	80.73%	81.95%	80.68%	78.98%	81.57%	81.23%	80.66%	80.05%	80.09%
Комбінація №4	78.92%	80.12%	81.87%	79.36%	80.89%	80.35%	78.76%	81.46%	81.56%
Комбінація №5	80.08%	80.35%	80.21%	80.47%	80.26%	80.54%	80.89%	80.57%	80.63%
Комбінація №6	78.45%	81.78%	81.45%	79.82%	81.32%	81.78%	79.29%	81.01%	80.30%

### 3.3 Аналіз проведеного дослідження

Дослідження показує, що використання комбінацій різних фільтрів перед введенням даних до нейронної мережі може значно покращити її здатність до розпізнавання об'єктів навіть в умовах шуму та розмитості. Зокрема, комбінації фільтрів Гауса та медіанного фільтра, а також фільтрів середнього значення та фільтра Собеля виявилися найефективнішими у подоланні негативного впливу зовнішніх факторів. Дані результати можуть мати важливе практичне застосування для оптимізації роботи нейронних мереж в умовах реального середовища зображень.

Після проведення ретельного дослідження впливу різних комбінацій фільтрів на роботу нейронної мережі, наші висновки дозволяють визначити оптимальну стратегію для покращення процесу розпізнавання об'єктів. Виявлено, що використання комбінації Фільтра Собеля та медіанного фільтра має ряд переваг, які варто розглянути. Використання Фільтра Собеля дозволяє виділяти контури та ключові особливості на зображенні, що забезпечує нейронній мережі більш точний та релевантний вхідний сигнал. Медіанний фільтр ефективно зменшує вплив шуму на зображення. Це особливо важливо в умовах реального середовища, де зображення може бути піддане різним видам перешкод. Застосування Медіанного фільтру виявляється ефективним у збереженні важливих деталей та структури зображення, що може бути важливим для точного розпізнавання. Всі ці фактори взаємодіють для створення надійної та стабільної системи розпізнавання. Наявність гармонійної комбінації фільтрів сприяє оптимальній роботі нейронної мережі навіть в умовах викликів реального світу. Загалом, використання Фільтра Собеля та медіанного фільтра виявляється важливим стратегічним вибором для покращення продуктивності та точності роботи нейронної мережі у реальних умовах. Ця комбінація може бути ключовою в розвитку ефективних та надійних систем комп'ютерного зору та обробки зображень.

Поєднання фільтра Гауса і фільтра середнього значення у попередній обробці зображень для штучного інтелекту може виявитися не ефективним через втрату важливих деталей та надмірне вирівнювання. Гаусівський фільтр може розмити важливі деталі, а фільтр середнього значення може додатково згладити зображення, вносячи додаткову втрату інформації. Поєднання цих фільтрів може призвести до надмірного вирівнювання контурів та рівномірного зменшення контрасту. Такий підхід може бути невдалим, оскільки завданням є не тільки зменшення шуму, але й збереження важливих деталей.

Поєднання фільтра Гауса та медіанного фільтра може не завжди призводити до високих результатів у попередній обробці зображень для штучного інтелекту. Є кілька ключових аспектів, які можуть пояснити цю ситуацію. По-перше, фільтр Гауса є лінійним фільтром, що означає, що його вплив на пікселі залежить від лінійного перетворення їх значень. З іншого боку, медіанний фільтр є нелінійним та замінює значення пікселя на медіану його сусідів. Поєднання цих різних за природою фільтрів може створювати неочікувані ефекти, які можуть впливати на роботу штучного інтелекту. Вплив на текстурні особливості також важливий, оскільки фільтр Гауса зменшує градієнт текстурних особливостей, тоді як медіанний фільтр може зробити зображення менш чітким. Поєднання цих фільтрів може впливати на визначеність текстур на зображенні, що може мати важливе значення для роботи з штучним інтелектом, особливо в завданнях розпізнавання об'єктів.

Поєднання фільтра середнього значення та медіанного фільтра у попередній обробці зображень для штучного інтелекту може демонструвати низьку ефективність з кількох причин, що стосуються характеристик цих фільтрів та їх впливу на зображення. По-перше, фільтр середнього значення використовує усереднення значень пікселів у визначеній області, розмиттячи контури та текстурні особливості зображення. Це може вплинути на різкість та деталізацію зображення, що є критичним для задач розпізнавання об'єктів штучним інтелектом. По-друге, медіанний фільтр



видаляє значення пікселів відповідно до їхнього ранжування, і він є більш робастим до викидів або шуму в порівнянні з фільтром середнього значення. Однак він також може призводити до зменшення контрастності та деталізації важливих елементів на зображенні.

Використання комбінації фільтрів Гауса, медіанного та середнього значення для попередньої обробки зображень перед їхнім поданням на штучний інтелект може мати значущий вплив на якість обробки та розпізнавання об'єктів. Кожен із цих фільтрів привносить свої унікальні характеристики. Фільтр Гауса відповідає за розмиття та згладжування зображення, видаляючи високочастотний шум та сприяючи покращенню якості зображення. Медіанний фільтр, у свою чергу, виявляється ефективним у видаленні викидів і шуму, що може виникати на зображенні. Фільтр середнього значення використовує усереднення для створення ефекту загального розмиття та згладжування. Однак, як можна було побачити, поєднання цих фільтрів може викликати певні недоліки. Зокрема, комбінація фільтра Гауса та медіанного фільтра може вести до збереження недоречних артефактів та високочастотного шуму, які можуть негативно впливати на точність розпізнавання об'єктів нейронною мережею. Це може бути пов'язано із спробою збереження деталей, які медіанний фільтр не завжди ефективно видаляє.

## СТАРТАП-ПРОЄКТ

Всі представлені фільтри винайдені ще дуже давно, тому краще продавати вже навчену нейронну мережу із використаною комбінацією фільтрів. Дана нейронна мережа підійде для розпізнавання та класифікації об'єктів на зображенні. Це досить поширена область застосування. Одні з основних потенційних користувачів та їхні вигоди представлені в таблиці 3.2.1.

Цільова аудиторія	Застосування
Лікарям	Автоматизована діагностика на медичних зображеннях. Розпізнавання патологій та клітинних структур
Працівникам на заводі	Визначення дефектів чи несправностей на виробничих лініях.
Працівникам супермаркетів	Розпізнавання типів продуктів або пакувань на конвеєрі.
Працівникам охорони	Контроль доступу та безпеки за допомогою розпізнавання обличчя.
Екологам	Моніторинг екосистеми та захисту біорізноманіття.
Логістам	Автоматичне розпізнавання товарів на складах та в логістичних системах.
Туристам	Системи розпізнавання можуть надавати інформацію про туристичні об'єкти на основі фотографій, взятих туристами.

Працівникам електростанцій	Візуальний контроль елементів електростанцій та прогнозування можливих поломок.
----------------------------	---

Висновок: оскільки попит дійсно великий, конкурентів на ринку дуже багато. Особливо нейромережа із двома цифровими фільтрами в архітектурі буде вигідна для компаній, які планують інтегрувати штучний інтелект у сферу своєї діяльності.

## ДЖЕРЕЛА

1. Why\_neural\_networks\_are\_important [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: [https://futureside.com/artificial-neural-networks/#Why\\_neural\\_networks\\_are\\_important](https://futureside.com/artificial-neural-networks/#Why_neural_networks_are_important).
2. The importance of Artificial Neural Networks in realising Artificial Intelligence [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://medium.com/supervisionearth/the-importance-of-artificial-neural-networks-in-realising-artificial-intelligence-958428a24f44>.
3. What is a Neural Network? [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/what-is/neural-network/>.
4. Artificial Neural Networks. What they are & why they matter [Електронний ресурс] – Режим доступу до ресурсу: [https://www.sas.com/en\\_sa/insights/analytics/neural-networks.html](https://www.sas.com/en_sa/insights/analytics/neural-networks.html).
5. Early Stopping [Електронний ресурс] – Режим доступу до ресурсу: <https://paperswithcode.com/method/early-stopping>.
6. Using Early Stopping to Reduce Overfitting in Neural Networks [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://medium.com/data-science-365/using-early-stopping-to-reduce-overfitting-in-neural-networks-7f58180caf5b>.
7. How to Improve the Accuracy of Your Image Recognition Models [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/improve-image-recognition-model-accuracy-with-these-hacks/>.
8. What is the Dropout Layer? [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://databasecamp.de/en/ml/dropout-layer-en>.
9. A Gentle Introduction to Dropout for Regularizing Deep Neural Networks [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>.
10. A Gentle Introduction to Transfer Learning for Deep Learning [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>.
11. A Newbie-Friendly Guide to Transfer Learning [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.v7labs.com/blog/transfer-learning-guide>.
12. Simple Guide to Hyperparameter Tuning in Neural Networks [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://towardsdatascience.com/simple-guide-to-hyperparameter-tuning-in-neural-networks-3fe03dad8594>.
13. Applying Hyperparameter Tuning To Neural Networks [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://towardsdatascience.com/hyperparameter-tuning-neural-networks-101-ca1102891b27>.

14. Data Augmentation | How to use Deep Learning when you have Limited Data—Part 2 [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/>.
15. The Essential Guide to Data Augmentation in Deep Learning [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://www.v7labs.com/blog/data-augmentation-guide>.
16. Low Pass Filtering [Электронный ресурс] – Режим доступа до ресурсу: <https://www.nv5geospatialsoftware.com/docs/LowPassFilter.html>.
17. Low-Pass Filtering (Blurring) [Электронный ресурс] – Режим доступа до ресурсу: [https://cdn.diffractionlimited.com/help/maximdl/Low-Pass\\_Filtering.htm](https://cdn.diffractionlimited.com/help/maximdl/Low-Pass_Filtering.htm).
18. What Does a Low Pass Filter Do? Meaning, Uses, and Applications in Different Fields [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.hollyland.com/blog/tips/what-does-a-low-pass-filter-do>.
19. Low-Pass Filters [Электронный ресурс] – Режим доступа до ресурсу: <https://www.electricity-magnetism.org/low-pass-filters/>.
20. High Pass Filtering [Электронный ресурс] – Режим доступа до ресурсу: <https://www.nv5geospatialsoftware.com/docs/HighPassFilter.html>.
21. High-Pass Filtering (Sharpening) [Электронный ресурс] – Режим доступа до ресурсу: [https://cdn.diffractionlimited.com/help/maximdl/High-Pass\\_Filtering.htm](https://cdn.diffractionlimited.com/help/maximdl/High-Pass_Filtering.htm).
22. Lowpass Gaussian Filter [Электронный ресурс] – Режим доступа до ресурсу: <https://www.nanophys.kth.se/nanolab/afm/icon/bruker-help/Content/SoftwareGuide/Offline/ModifyCommands/Gaussian.htm>.
23. Estevao Gedraite. Investigation on the effect of a Gaussian Blur in image filtering and segmentation — 2011 — С. 2—3 — [https://www.researchgate.net/publication/261278360\\_Investigation\\_on\\_the\\_effect\\_of\\_a\\_Gaussian\\_Blur\\_in\\_image\\_filtering\\_and\\_segmentation](https://www.researchgate.net/publication/261278360_Investigation_on_the_effect_of_a_Gaussian_Blur_in_image_filtering_and_segmentation)
24. Gaussian Filter and Derivatives of Gaussian [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://hannibunny.github.io/orbook/preprocessing/04gaussianDerivatives.html#dimensional-gaussian-filter-1>.
25. Image Filtering Techniques in Image Processing — Part 1 [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://medium.com/@henriquevedoveli/image-filtering-techniques-in-image-processing-part-1-d03362fc73b7>.
26. Median Filtering: A New Insight. // Journal of Mathematical Imaging and Vision. – 2017. – №58. – С. 1–17.
27. FPGA Implementation of Median Filter Using an Improved Algorithm for Image Processing [Электронный ресурс] – Режим доступа до ресурсу: [https://www.academia.edu/16076355/FPGA\\_Implementation\\_of\\_Median\\_Filter\\_Using\\_an\\_Improved\\_Algorithm\\_for\\_Image\\_Processing?uc-g-sw=5718126](https://www.academia.edu/16076355/FPGA_Implementation_of_Median_Filter_Using_an_Improved_Algorithm_for_Image_Processing?uc-g-sw=5718126).

28. Median Filter [Электронный ресурс] – Режим доступа до ресурсу: [https://www.academia.edu/9895081/Median\\_Filter?uc-g-sw=5718126](https://www.academia.edu/9895081/Median_Filter?uc-g-sw=5718126).
29. Spatial Filters – Averaging filter and Median filter in Image Processing [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/spatial-filters-averaging-filter-and-median-filter-in-image-processing/>.
30. Tutorial 1: Image Filtering [Электронный ресурс] – Режим доступа до ресурсу: <https://ai.stanford.edu/~syyeong/cvweb/tutorial1.html>.
31. Mean filter, or average filter [Электронный ресурс] – Режим доступа до ресурсу: <http://www.librow.com/articles/article-5>.
32. Sobel Filtering [Электронный ресурс] – Режим доступа до ресурсу: <https://onlinedocs.microchip.com/pr/GUID-37AD5EEE-6FAB-48FC-89F6-CAA649534B2A-en-US-1/index.html>.
33. Sobel Operator [Электронный ресурс] – Режим доступа до ресурсу: [https://www.tutorialspoint.com/dip/sobel\\_operator.htm](https://www.tutorialspoint.com/dip/sobel_operator.htm).
34. How the Sobel Operator Works [Электронный ресурс] – Режим доступа до ресурсу: <https://automaticaddison.com/how-the-sobel-operator-works/>.
35. Keras [Электронный ресурс] – Режим доступа до ресурсу: <https://keras.io/>.
36. Multi-Class Classification Tutorial with the Keras Deep Learning Library [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/>.

## ДОДАТОК А

Розпізнавання та класифікація об'єктів за допомогою нейронної мережі на мові програмування Python:

```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.applications.vgg16 import preprocess_input
from keras.applications.vgg16 import decode_predictions
from keras.applications.vgg16 import VGG16

model = VGG16()

img1 = load_img('tiger.jpeg', target_size=(224, 224))
img2 = load_img('tigerblur.jpeg', target_size=(224, 224))
img3 = load_img('tigernoise.jpg', target_size=(224, 224))

def preprocess(image):
    image = img_to_array(image)
    image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
    image = preprocess_input(image)
    return image

def pred_modele(image):
    image = preprocess(image)
    y_pred = model.predict(image)
    label = decode_predictions(y_pred)
    label = label[0][0]
    return (label[1], label[2]*100)

img_list = [img1, img2, img3]

for i in range(3):
    print("На рисунку", i+1, ":", pred_modele(img_list[i])[0], 'із ймовірністю',
          round(pred_modele(img_list[i])[1], 2), '%')
```

## ДОДАТОК Б

Застосування фільтра середнього значення:

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

# Read the image
img = cv.imread('tiger.jpg')

# Apply blur
blur = cv.blur(img, (30, 30))

# Plot the images using matplotlib
plt.figure(figsize=(10, 5))

# Original image
plt.subplot(121), plt.imshow(cv.cvtColor(img, cv.COLOR_BGR2RGB)),
plt.title('Original')
plt.xticks([], plt.yticks([]))

# Blurred image
plt.subplot(122), plt.imshow(cv.cvtColor(blur, cv.COLOR_BGR2RGB)),
plt.title('Blurred')
plt.xticks([], plt.yticks([]))

# Show the plot
plt.show()
```



## ДОДАТОК В

Застосування фільтра Гауса:

```
import numpy as np
import scipy.ndimage as ndimage
import matplotlib.pyplot as plt

# Read the image using matplotlib
img = plt.imread('tiger.JPG')

plt.subplot(1, 2, 1)
plt.imshow(img, interpolation='nearest')
plt.title('Оригінальне зображення')

# Apply Gaussian filter to the image
img_filtered = ndimage.gaussian_filter(img, sigma=(10, 10, 0), order=0)

# Create a subplot with 1 row and 2 columns
plt.subplot(1, 2, 2)
plt.imshow(img_filtered, interpolation='nearest')
plt.title('Зображення після накладення фільтра Гауса')

# Display the filtered image
plt.imshow(img_filtered, interpolation='nearest')
plt.show()
```

## ДОДАТОК Г

Застосування медіанного фільтра:

```
import numpy as np
import scipy.ndimage as ndimage
import matplotlib.pyplot as plt

# Read the image using matplotlib
img = plt.imread('tiger.JPG')

# Create a subplot with 1 row and 2 columns
plt.subplot(1, 2, 1)
plt.imshow(img, interpolation='nearest')
plt.title('Оригінальне зображення')

# Apply Median filter to the image
img_filtered = ndimage.median_filter(img, size=(30, 30, 1))

# Create a subplot with 1 row and 2 columns
plt.subplot(1, 2, 2)
plt.imshow(img_filtered, interpolation='nearest')
plt.title('Зображення після накладення Медіанного фільтра')

# Display the subplots
plt.show()
```

## ДОДАТОК Д

Застосування фільтра Собеля:

```
import cv2
import matplotlib.pyplot as plt

# Read the original image
img = cv2.imread('tiger.jpg')
# converting because opencv uses BGR as default
RGB_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Create a figure to display images
plt.figure(figsize=(15, 5))

# Plot the original image
plt.subplot(1, 2, 1)
plt.imshow(RGB_img)
plt.title("Original Image")

# converting to gray scale
gray = cv2.cvtColor(RGB_img, cv2.COLOR_BGR2GRAY)
# convolute with sobel kernels
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=7) # x
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=7) # y

# Blend Sobel-x and Sobel-y images
blended = cv2.addWeighted(sobelx, 0.5, sobely, 0.5, 0)

# Plot the blended image
plt.subplot(1, 2, 2)
plt.imshow(blended, cmap='gray')
plt.title("Blended Sobel-x and Sobel-y")

# Display the figure
plt.show()
```

## ДОДАТОК Ж

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Радіотехнічний факультет

### **РАДІОЕЛЕКТРОНІКА В XXI СТОЛІТТІ**

Матеріали V Всеукраїнської науково-технічної конференції студентів  
та аспірантів «Радіоелектроніка в XXI столітті»  
10 – 12 травня 2023  
Київ, Україна

---

**ЗМІСТ**
**СЕКЦІЯ 1. ТЕЛЕКОМУНІКАЦІЯ, РАДІОЛОКАЦІЯ, НАВІГАЦІЯ.  
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ**

Аврамишин А. І. (керівник Лащевська Н. О.) Розпізнавання продуктів харчування з використанням згорткової нейронної мережі	8
Броварник С. А. (керівник Зінгер Я. Л.) Використання штучного інтелекту та машинного навчання для покращення роботи інерційних навігаційних систем	11
Бруско А. В. (керівник Мирончук О. Ю.) Визначення напрямку радіосигналу за допомогою кількох приймачів SDR	14
Гайдай А. В. (керівник Товкач І. О.) Алгоритм локалізації багатьох БПЛА на основі сенсорної мережі	16
Галкін Т. А. (керівник Катін П. Ю.) Аналіз мікрокомп'ютерів для побудови автоматизованої мережевої системи моніторингу і управління зрощенням	19
Гелетко О. Я. (керівник Адаменко В. О.) Формування ефективного променя ФАР за допомогою алгоритмів машинного навчання	23
Гладовський М. В. (керівник Неуймін О. С.) Мінідрон моніторингу обстановки ближнього радіусу дії	26
Голодяєва П. В., Ходаницький О. О., Гайдай А. В., Чмир Р. В. (керівник Товкач І. О.) Аналіз алгоритму оцінки місцеположення рухомого об'єкта за допомогою ансцентного фільтра Калмана	29
Yevdokimov D. O. (Supervisor Zinher Y. L.) Computer vision for detection, classification, and tracking of non-typical low-contrast targets against complex background surfaces	32
Желябовський А. Ю. (керівник Піддубний В. О.) Машинне навчання при ідентифікації металів вихрострумовою радіотехнічною системою	35
Запорожець Д. А. (керівник Вишневий С. В.) Придушення завад на цифрових зображеннях	38
Захарченко М. М. (керівник Шпилька О. О.) Розрахунок енергетичних параметрів каналу зв'язку для високошвидкісної передачі інформації з супутників низьких навколосемних орбіт	41
Іванченко Ю. В. (керівник Чмельов В. О.) Підходи щодо забезпечення ефективної роботи РЛС в різних умовах	44