

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ ННЦЗФН _____
(повна назва)

Кафедра _____ Штучного інтелекту _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

_____ Дослідження та розробка систем комп'ютерного зору на основі _____
глибинних нейронних мереж _____
(тема)

Виконав:
студент 2 курсу, групи _____ СШЗм-20-2 _____
Московка А.Д. _____
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки _____
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту _____
(повна назва спеціалізації)

Керівник _____ к.т.н., доц. Дейнеко А.О. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет _____ ННЦЗФН _____
(повна назва)
Кафедра _____ Штучного інтелекту _____
(повна назва)
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)
Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Системи штучного інтелекту (СШІ) _____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Московці Анастасії Денисівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та розробка систем комп'ютерного зору на основі глибинних нейронних мереж

затверджена наказом університету від 25 жовтня 20 21 р. № 168Стз

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20 ____ р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки систем комп'ютерного зору, keras documentation, набори даних kaggle

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Огляд існуючих підходів до задач комп'ютерного зору

2) Методологічні основи вирішення задач комп'ютерного зору

3) Навчання передтренованих нейронних мереж

4) Імітаційне моделювання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – приклад ознак Хаара на обличчі, Рисунок 2 – повнозв'язна нейронна мережа прямого поширення, Рисунок 3 – неповнозв'язна нейронна мережа, Рисунок 4 – самоорганізовна карта Кохонена, Рисунок 5 – Карта Кохонена, Рисунок 6 – приклад згортки матриць розмірністю 7×7 та 3×3, Рисунок 7 – Зв'язок кількості карт між згортковим і вхідним шаром, Рисунок 9 – загальна архітектура згорткової мережі, Рисунок 10 – приклад деформованих зображень, Рисунок 11 – архітектура мережі VGG16, Рисунок 12 – архітектура Xception, Рисунок 13 – блок MobileNet, Рисунок 14 – відмінність між MobileNetV1 та MobileNetV2, Рисунок 15 – архітектура MobileNetV2, Рисунок 16 – відмінність звичайного машинного навчання та переносу навчання, Рисунок 17 – виділення блоків для тонкого налаштування

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу	01.09.2021	виконано
2	Теоретичні дослідження	01.10.2021	виконано
3	Експериментальне моделювання та навчання моделі	25.10.2021	виконано
4	Оформлення пояснювальної записки	10.11.2021	виконано
5	Оформлення графічних матеріалів	20.11.2021	виконано
6	Попередній захист		
7	Захист перед ЕК		

Дата видачі завдання 01 вересня 20 21 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис) _____
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 70 с., 26 рис., 1 табл., 39 джерел.

ГЛИБИННІ НЕЙРОННІ МЕРЕЖІ, КОМП'ЮТЕРНИЙ ЗІР,
РОЗПІЗНАВАННЯ ОБРАЗІВ, ПЕРЕДНАВЧЕНІ НМ, ПЕРЕНОС
НАВЧАННЯ, MOBILENETV2, VGG16, XCEPTION

Метою кваліфікаційної магістерської роботи є дослідження та розробка систем комп'ютерного зору на основі глибинних нейронних мереж.

Об'єктом дослідження є глибокі нейронні мережі, використовувані для задач класифікації зображень

Предметом дослідження є методи вирішення задач комп'ютерного зору, зокрема задач, пов'язаних з розпізнаванням образів.

Методи навчання – аналіз існуючих підходів до задач комп'ютерного зору, аналіз релевантних джерел, імітаційне моделювання з метою оцінки сучасних архітектур глибинних нейронних мереж.

РЕФЕРАТ

Пояснительная записка: 70 с., 26 рис., 1 табл., 39 источников.

ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ,
РАСПОЗНАВАНИЕ ОБРАЗОВ, ПРЕДОБУЧЕННЫЕ НС, ПЕРЕНОС
ОБУЧЕНИЯ, MOBILENETV2, VGG16, XCEPTION

Цель квалификационной магистерской работы – исследование и разработка систем компьютерного зрения на основе глубоких нейронных сетей.

Объектом исследования являются глубокие нейронные сети, используемые для задач классификации изображений.

Предметом исследования являются методы решения задач компьютерного зрения, в частности задач, связанных с распознаванием образов.

Методы обучения – анализ существующих подходов к задачам компьютерного зрения, анализ релевантных источников, имитационное моделирование для оценки современных архитектур глубинных нейронных сетей.

ABSTRACT

Explanatory note: 70 p., 26 fig., 1 tabl., 39 sources.

COMPUTER VISION, DEEP NEURAL NETWORKS, IMAGE RECOGNITION, MOBILENETV2, PRETRAINED NN, TRANSFER LEARNING, VGG16, XCEPTION,

The purpose of the qualifying master's work is the research and development of computer vision systems based on deep neural networks.

The object of the research is deep neural networks used for image classification problems.

The subject of this research is methods for solving computer vision problems, in particular, problems related to pattern recognition.

Teaching methods - analysis of existing approaches to computer vision problems, analysis of relevant sources, simulation modeling for assessing modern architectures of deep neural networks.

ЗМІСТ

Вступ.....	9
1 Огляд існуючих підходів до задач комп'ютерного зору	10
1.1 Розпізнавання образів та комп'ютерний зір	10
1.2 Існуючі методи вирішення задач в області комп'ютерного зору	11
1.2.1 Метод Віоли-Джонса.....	12
1.2.2 Нейронні мережі	15
1.2.3 Самоорганізовані карти Кохонена.....	18
1.3 Глибинні нейронні мережі в задачах комп'ютерного зору	22
1.4 Постановка задачі дослідження.....	23
2 Методологічні основи вирішення задач комп'ютерного зору	25
2.1 Згорткові нейронні мережі.....	25
2.2 Навчання згорткової нейронної мережі.....	29
2.3 Розширення навчальної множини (data augmentation).....	32
2.4 Метрики	33
3 Навчання передтренованих нейронних мереж	36
3.1 Передтреновані нейронні мережі	36
3.2 Архітектура VGG16.....	37
3.3 Архітектура Xception.....	38
3.4 Архітектура MobileNet	39
3.5 Transfer learning	42
4 Імітаційне моделювання	48
4.1 Огляд набору даних та їх аугментація.....	48
4.2 Навчання мереж	49
4.2.1 Результати навчання моделі VGG16.....	50

4.2.2	Результати навчання моделі Xception.....	51
4.2.3	Результати навчання моделі MobileNetV2	53
4.5	Порівняння результатів навчання та тестування.....	54
	Висновки	56
	Перелік джерел посилання.....	57
	Додаток А Вихідний код для навчання мереж.....	62
	Додаток Б Відомість кваліфікаційної роботи магістра.....	70

ВСТУП

З розвитком технологій у сучасному світі все більше різноманітних програмних систем, що полегшують життя людей. Серед таких систем збільшується кількість складних програмних комплексів, що за принципом своєї дії подібні до механізму обробки інформації людиною.

Глибинні нейронні мережі застосовуються у багатьох сферах, наприклад у системах що спостерігають за зображеннями на камерах, фільтрують зображення або обробляють тексти. Такі сфери застосування мають одну спільну рису – образність вхідних даних, що не можуть бути оброблені без виділення характерних ознак.

Саме тому дослідження глибинних нейронних мереж є актуальним напрямком, адже існує безліч задач, що використовують такі абстрактні дані. З таких задач варто виділити задачі комп'ютерного зору, зокрема, задачі розпізнавання об'єктів на зображеннях.

В період коронавірусної пандемії, що почалась у 2019 році, людям необхідно ретельно придержуватися карантинних заходів для зменшення поширення хвороби. В цій роботі запропоновано систему, що може підвищити якість контролю дотримання маскового режиму, а саме – систему класифікації зображень за наявністю маски на обличчі.

Систему розроблено із застосуванням нових нейромережевих архітектур, які, в порівнянні з попередніми, є більш продуктивними.

Розроблену систему можна застосовувати у закладах охорони здоров'я для контролю придержання відвідувачами маскового режиму. Схожа система, наприклад, використовуються зараз у ряді лікарень, не відчиняючи двері поки людина не продемонструє на камеру наявність маски на обличчі.

1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ДО ЗАДАЧ КОМП'ЮТЕРНОГО ЗОРУ

1.1 Розпізнавання образів та комп'ютерний зір

Розпізнавання образів – це галузь науки, присвячена розробці принципів і побудові програмних і апаратних систем, мета яких – визначити, якому з наперед визначених класів належать об'єкт, що аналізується. Об'єктом може бути будь-який сигнал, явище, реальний об'єкт чи абстракція тощо [1].

Об'єкти визначаються обмеженою послідовністю характеристик або ознак. Ознаки – це дані, отримані в результаті вимірювання об'єктів, що використовуються для їх класифікації. Набір ознак об'єкта складає так званий вектор ознак. Цей вектор ознак можна виразити у вигляді точки в гіперпросторі ознак [2]. Кожне зображення зіставляється лише з одним вектором ознак, так само, як вектор ознак зіставляється лише з одним зображенням. Відповідність об'єкта одній з категорій називається класифікацією. Еталонні об'єкти, тобто об'єкти категорії яких вже відомі є основою для класифікації [3].

Образ – опис об'єкта, що використовується для вирішення задачі розпізнавання образів, співвідношення між об'єктом і одним із класів. Для того, щоб розв'язати задачу розпізнавання образів, усі можливі образи об'єкта необхідно розділити на множину категорій. Також необхідно вказати набір еталонних об'єктів. Кожен еталонний об'єкт належить до одного з цих класів. Також існують об'єкти невідомого класу. Задача розпізнавання образів – визначити, до якого класу належить цей об'єкт. Або розробити алгоритм виконання цієї операції над будь-яким об'єктом, що розглядається [4]. Чим точніше алгоритм визначає належність об'єкта класу, тим точніше буде система розпізнавання образів.

Комп'ютерний зір – це теорія і технологія створення штучних комп'ютерних систем для виявлення, класифікації та відстеження об'єктів. Вони отримують інформацію із зображень, які можуть бути представлені певним відеорядом, зображеннями або тривимірними даними з різних камер, відсканованими зображеннями тощо. Метою комп'ютерного зору є формування корисних висновків про реальні об'єкти та сцени на основі аналізу зображень, отриманих за допомогою датчиків [5].

Багато систем комп'ютерного зору реалізують такі функції:

- отримання зображень;
- попередня обробка;
- виділення деталей;
- детектування/сегментація;
- високорівнева обробка.

Комп'ютерний зір може генерувати змінене зображення або список значень певних параметрів зображення, таких як розмір об'єкта, колір, орієнтація камери, швидкість тощо [6].

Таким чином, основною відмінністю галузі комп'ютерного зору від галузі розпізнавання образів є принципова різниця задач, які вони вирішують. Якщо розпізнавання образів є методологією, яка виконує основні кроки – визначає основні ознаки між зображеннями та виконує класифікацію, тобто оцінює до якого класу («типу») належить зображений об'єкт, то комп'ютерний зір – комплексна система, задачею якої є прийняття рішень на основі обробки результатів, отриманих за допомогою розпізнавання образів.

1.2 Існуючі методи вирішення задач в області комп'ютерного зору

Для визначення напрямку роботи даному підрозділі пропонується розглянути методи, із застосуванням яких вирішуються задачі розпізнавання образів.

1.2.1 Метод Віоли-Джонса

Алгоритм Віоли-Джонса (Viola-Jones) успішно вирішував проблему розпізнавання обличчя, він є основним алгоритмом розпізнавання об'єктів на зображеннях. Алгоритм був розроблений Полом Віолою (Paul Viola) та Майклом Джонсом (Michael Jones) у 2001 році і заснований на використанні технології «ковзного вікна» та каскаду Хаара як функцій. Техніка «ковзного вікна» («sliding window») полягає у створенні рамки, яка рухається по зображенню [7]. Розмір рамки підбирається так, щоб ознакові фрагменти зображення попадали у рамку, після чого з використанням каскадів Хаара визначається чи є на такому зображенні обличчя

Каскад Хаара після навчання класифікує вхідне зображення та розділяє вхідні дані на дві категорії (залежно від того, чи містить зображення визначений об'єкт) [8]. Метод Віоли-Джонса є одним з найкращих методів розпізнавання об'єктів, оскільки порівняно з сучасними методами швидкість обробки даних і якість розпізнавання все ще дуже високі, а можливість помилкових спрацьовувань для цього класифікатора дуже низька. Перед розпізнаванням обличчя алгоритм перетворює зображення у відтінки сірого, щоб зменшити кількість даних і спростити обробку [9].

Принцип роботи алгоритму Віоли-Джонса такий:

Крок 1. Вихідне цифрове кольорове триканальне (RGB) зображення перетворюється у відтінки сірого (одноканальне зображення з $2^8 = 256$ рівнями яскравості) і перетворюється в матрицю яскравості, тобто інтегральне представлення зображення. Інтегральне представлення дозволяє швидко розрахувати загальну яскравість будь-якого прямокутника зображення і незалежно від розмірів прямокутника час обчислення залишається незмінним. Інтегральне представлення зображення являє собою матрицю, яка відповідає розміру вихідного зображення, кожен

елемент зберігає суму яскравостей усіх пікселів ліворуч і над елементом. Елементи матриці розраховуються за такою формулою:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j), \quad (1.1)$$

де $I(i, j)$ – яскравість пікселя вихідного зображення.

Кожен елемент матриці $L(x, y)$ є сумою пікселів у прямокутнику від $(0, 0)$ до (x, y) , тобто значення кожного пікселя (x, y) дорівнює сумі значень пікселів, що розташовані лівіше та вище за цей. Розрахунок матриці вимагає лінійного часу, пропорційного кількості пікселів у зображенні, тому інтегральне зображення прораховується за один прохід.

Крок 2. На отриману матрицю яскравості накладається одна з квадратних масок (ознак Хаара). Ознака Хаара – це сукупність прямокутних областей зображень, які примикають одна до одної і поділені на дві групи. Ознак Хаара існує досить багато: різноманітні комбінації областей різної ширини та висоти в різних положеннях на зображенні. Маска проходить по зображенні, змінюючи положення та розмір. Ознака f – це відображення $f: X \rightarrow D_f$, де X – множина досліджуваних об'єктів, D_f – множина допустимих значень ознаки f . Якщо задані ознаки f_1, \dots, f_n , то вектор ознак $x = (f_1(x), \dots, f_n(x))$ називається ознаковим описом об'єкта $x \in X$. Ознаки можна ідентифікувати за самим об'єктом, тоді множина $X = D_{f_1} \dots D_{f_n}$ називається простором символів. Приклади ознак Хаара та їх положення на зображенні показано на рисунку 1.1.

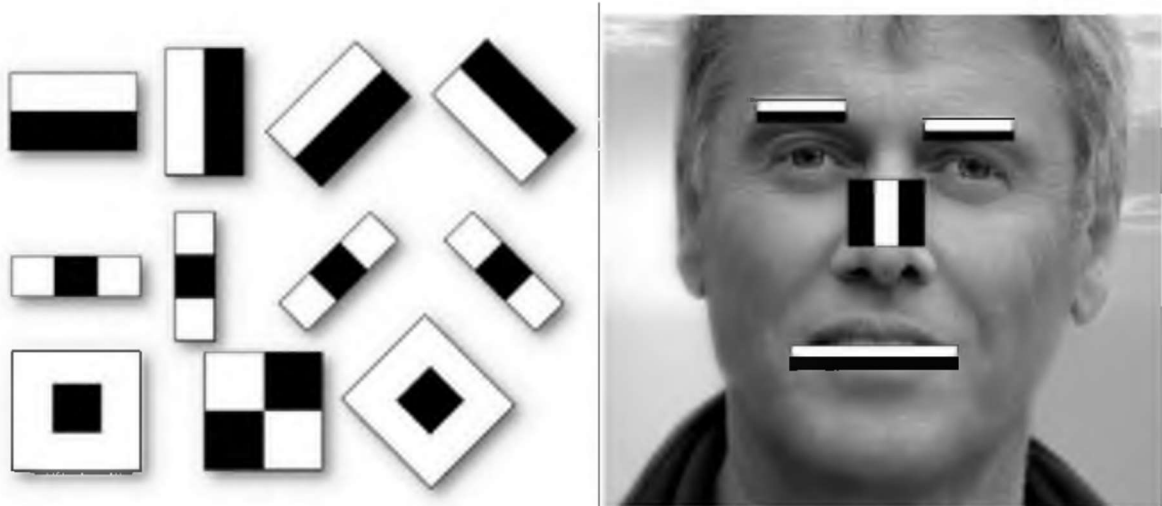


Рисунок 1.1 – Приклад ознак Хаара на обличчі [7]

Ознаки Хаара використовують той факт, що на зображеннях обличчя брови характеризуються темними горизонтальними лініями, носова перегородка – світлішими вертикальними лініями тощо.

Крок 3. Значення ознаки дорівнює різниці між сумою пікселів у білому прямокутнику та чорному прямокутнику. У розрахунках використовується поняття інтегральних зображень. Отже, цифрові значення яскравості тих комірок у матриці, які потрапляють під білу частину маски додаються, а значення, які потрапляють під чорну частину – віднімаються. Якщо хоча б в одному випадку різниця між білою і чорною областями перевищує певний поріг, то ця область зображення обробляється далі. Якщо ця умова не виконується, область зображення не містить обличчя.

Крок 4. Виконується повторення з кроку 2 з новою маскою, але тільки для області зображення, яка пройшла перший тест. Область, яка пройшла всі стадії, – це обличчя.

Однак алгоритм Віоли-Джонса в основному використовується для розпізнавання облич в анфас або облич з невеликими кутами нахилу (приблизно до 30°). Якщо кут нахилу перевищує 30° або є обличчя, направлені вгору, вниз або вбік (положення в профіль), ймовірність

виявлення різко падає. З огляду на вимоги, ця особливість методу не дозволяє виявляти обличчя, що повернені під деяким кутом, що значно ускладнює (вимагає нового класифікатора) або унеможлиблює використання алгоритму Віоли-Джонса в сучасних виробничих системах. Іншим недоліком цього алгоритму є те, що розмічені зразки зображень не завжди можливо підготувати заздалегідь.

1.2.2 Нейронні мережі

Актуальність досліджень штучних нейронних мереж зумовлена можливістю використання нейронних мереж у різних галузях. Широта задач, що вирішуються нейронними мережами, багато в чому пояснюється можливістю навчання, яка на порядок підвищує ефективність роботи системи.

Нейронні мережі можуть сильно відрізнятися одна від одної, але у них є деякі спільні риси. Основою кожної нейронної мережі є прості елементи, які імітують роботу нейронів мозку. Нейрони визначаються їхнім поточним станом, збудженим чи загальмованим, і з'єднуються з іншими нейронами через синапси або просто зв'язки [10].

Кожен вихід кожного нейрона має багато вхідних синапсів. Відповідно до вхідного сигналу нейрон формує вихідний сигнал, який передається наступному нейрону. Якщо кожен нейрон в ШНМ з'єднаний з кожним нейроном, тобто сигнал поширюється лише в напрямку від вхідного шару до вихідного, без рекурсії, то така мережа називається повнозв'язною НМ прямого поширення (рисунок 1.2).

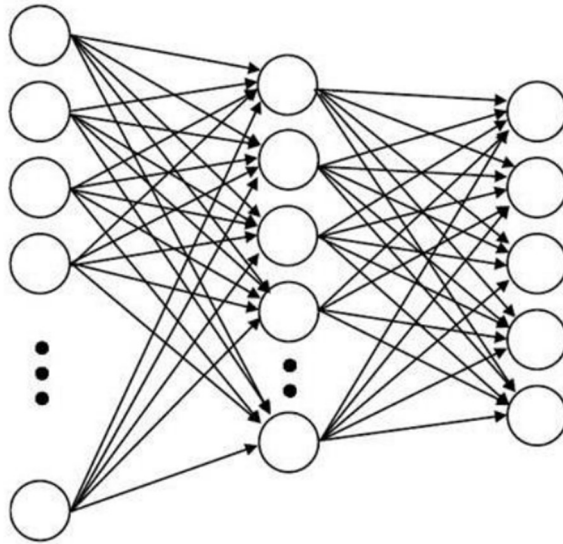


Рисунок 1.2 – Повнозв'язна нейронна мережа прямого поширення

Існують також неповнозв'язні нейронні мережі (часто їх називають перцептронами). Їх поділяють за кількістю шарів і характером з'єднання нейронів. У нейронній мережі з прямими зв'язками нейрони певного шару можуть бути з'єднані лише з нейронами попереднього шару. У нейронній мережі з перехресними зв'язками дозволені лише з'єднання в межах одного шару. Класична неповнозв'язна нейронна мережа представлена на рисунку 1.3.

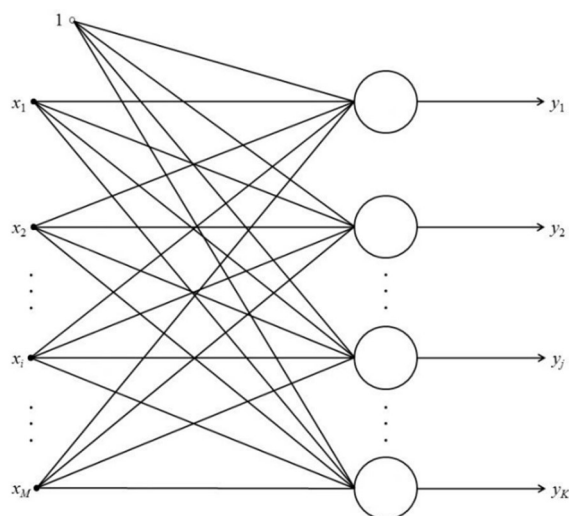


Рисунок 1.3 – Неповнозв'язна нейронна мережа

При використанні нейронної мережі для розв'язання задач вибирається стандартна конфігурація нейронної мережі. Якщо задача не може бути зведена до жодного відомого типу нейронної мережі, необхідно вирішити складну задачу синтезу нової конфігурації [10]:

- встановити класифікацію нейронних мереж;
- провести аналіз існуючих нейронних мереж;
- розробити основні критерії відбору нейронних мереж для побудови моделі;
- визначити основні характеристики для визначення якості моделі на основі нейронної мережі.

Основною характеристикою нейронної мережі є модель. Нейронну мережу можна охарактеризувати за типами нейронів, які використовуються в мережі, структурою мережевої моделі, способом навчання мережі та задачами, які мережа вирішує. Розглядаючи проблеми, що вирішуються нейронними мережами, можна визначити широкий спектр задач для обробки та аналізу даних: розпізнавання та класифікація об'єктів, прогнозування, керування, аналіз, стиснення даних тощо.

Відомі нейронні мережі можна розділити за типами структур нейронів на:

- гомогенні (однорідні);
- гетерогенні.

Гомогенні мережі складаються з нейронів одного типу з єдиною функцією активації, а гетерогенну мережу входять нейрони з різними функціями активації. Активаційна функція нейрона визначає нелінійне перетворення здійснюване нейроном. Існує безліч активаційних функцій. Найпоширеніші з них [10]:

- лінійна функція;
- порогова функція;

- сигмоїдальна функція (логістична функція, гіперболічний тангенс та ін.);
- радіально-базисна функція передачі;
- інші функції передачі.

Етап проектування, під час якого виконується коригування вхідних зв'язків називається навчанням нейронної мережі, і від його виконання залежить здатність НМ розв'язувати задачі. . На цьому етапі, крім параметра якості підбору вхідних зв'язків, важливу роль відіграє час навчання. У більшості випадків ці два параметри обернено пропорційні. Нейронні мережі можна навчати з учителем або без нього. У першому випадку мережа отримує значення вхідних і правильних вихідних сигналів, а мережа коригує свої вхідні з'єднання за внутрішніми алгоритмами. У другому випадку вихід НМ формується незалежно, а зв'язки налаштовуються за алгоритмом, який враховує тільки вхідний сигнал.

1.2.3 Самоорганізовані карти Кохонена

Самоорганізована карта Кохонена – це змагальна нейронна мережа з навчанням без вчителя, яка виконує задачі візуалізації та кластеризації. Ідею створення мережі запропонував фінський учений Теуво Кохонен. Цей вид мережі є методом проектування багатовимірного простору в більш низький (зазвичай двовимірний) простір, а також використовується для вирішення таких проблем, як моделювання та прогнозування. Ідея мережі Кохонена заснована на аналогії з особливостями людського мозку. Кора головного мозку являє собою плоский і складчастий лист. Він має певні топологічні характеристики (ділянки, що відповідають за близькі частини тіла, прилягають одна до одної, і на цій двовимірній поверхні відбивається все зображення людського тіла) [11].

Мережа Кохонена відрізняється від багатошарової нейронної мережі, вона дуже проста: має два шари: вхідний і вихідний. Її також називають самоорганізовною картою (рисунок 1.4)

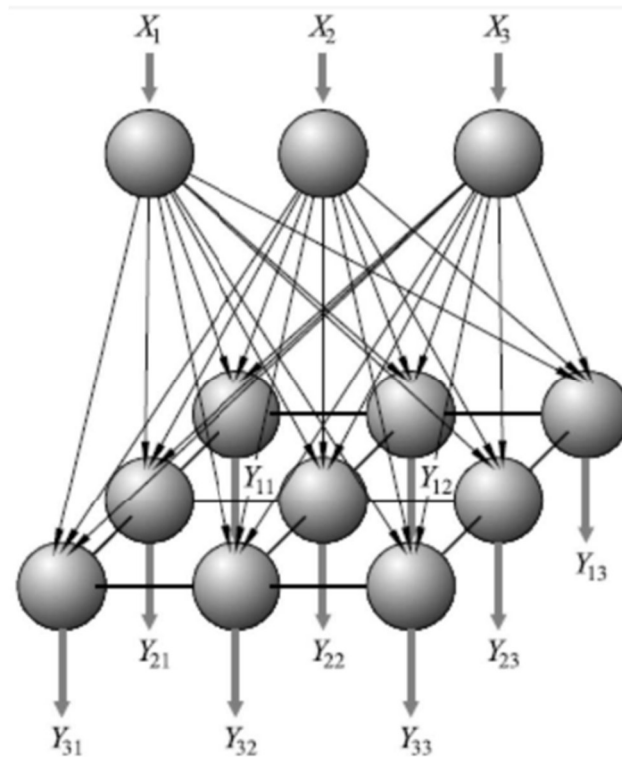


Рисунок 1.4 – Самоорганізовна карта Кохонена

Алгоритм роботи мережі [11].

Хай t – номер ітерації (ініціалізація відповідає значенню 0)

Ініціалізація.

Найпоширенішими є три способи встановлення початкової ваги вузла:

- завдання всіх координат випадковими числами;
- надання вектору ваги значення випадкового спостереження з вхідних даних;
- вибір векторів ваги з лінійного простору, натягнутого на головні компоненти набору вхідних даних.

Цикл.

– вибрати довільне спостереження $x(t)$ з множини вхідних даних;
 – знайти відстані від нього до векторів ваги всіх вузлів карти та визначити найближчий за вагою вузол $M_c(t)$. Це – ВМУ чи Winner. Умова на $M_c(t)$: $\|x(t) - w_c(t)\| \leq \|x(t) - w_i(t)\|$, для будь-якого $w_i(t)$, де $w_i(t)$ – вектор ваги вузла $M_i(t)$. Якщо знаходиться декілька вузлів, які задовольняють умові, ВМУ обирається випадковим чином серед них.

– визначити за допомогою функції h (функції сусідства) сусідів M_c та зміну їх векторів ваги.

Часто в якості функції сусідства використовується гауссівська функція:

$$h_{ci}(t) = \alpha(t) * e^{-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}}, \quad (1.2)$$

де $0 < \alpha(t) < 1$ – навчальний співмножник, що монотонно зменшується з кожною наступною ітерацією (тобто визначає наближення значення векторів ваги ВМУ і його сусідів до спостереження; чим більший крок, тим менше уточнення),

r_i, r_c – координати вузлів $M_i(t)$ та $M_c(t)$ на карті;

$\sigma(t)$ – співмножник, що зменшує кількість сусідів з ітерацією, монотонно зменшується;

– змінити вектор ваги за формулою

$$w_i(t) = w_i(t - 1) + h_{ci} * (x(t) - w_i(t - 1)) \quad (1.3)$$

Обчислення помилки карти виконується як середнє арифметичне відстаней між спостереженнями та векторами ваги відповідних їм ВМУ:

$$\frac{1}{N} \sum_{i=1}^N \|x_i - w_c\|, \quad (1.4)$$

де N – кількість елементів набору вхідних даних.

У цьому методі згенерована карта може бути представлена як набір шарів, кожен з яких є картою, створеною компонентом вихідних даних. Згенерований атлас можна використовувати для аналізу закономірностей між компонентами набору даних. Після формування карти отримуються набір вузлів, які можна відобразити у вигляді двовимірного зображення. При цьому кожен вузол карти може бути поставлений у відповідність з ділянкою на зображенні, чотирикутником або шестикутником, а його координати визначаються координатами відповідного вузла в сітці. Тепер для візуалізації залишилося лише визначити колір комірок цієї картинки. Тому використовується значення компонента. Найпростіший варіант – використовувати відтінки сірого. У цьому випадку комірка, що відповідає вузлу карти з найменшим значенням компонента або взагалі без запису, буде відображена чорним кольором, а комірка з записом із найбільшим значенням такого компонента – білим.

Приклад такої розфарбованої карти з використанням 7ми кольорів наведено на рисунку 1.5.

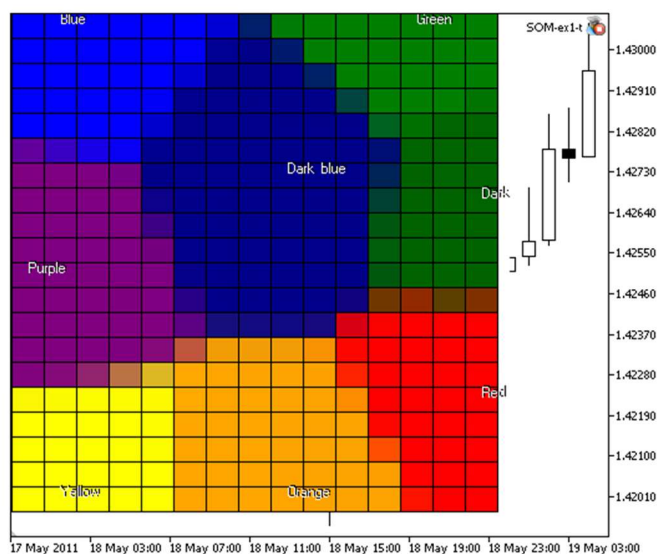


Рисунок 1.5 – Карта Кохонена

Отримані кольори разом утворюють карту, що відображає положення компонентів, взаємозв'язок між ними та взаємне розташування значень компонентів.

Кластером буде набір векторів, відстань між векторами всередині групи менша за відстань до сусідньої групи. Кластерну структуру при використанні алгоритму SOM можна відобразити шляхом візуалізації відстані між опорними векторами (вагами нейронів).

1.3 Глибинні нейронні мережі в задачах комп'ютерного зору

Серед методів вирішення задач класифікації зображень використовуються різні алгоритми. В даній роботі для вирішення цих задач запропоновано використовувати переднавчені згорткові нейронні мережі. Такі мережі мають високу якість у порівнянні з іншими способами розпізнавання образів на зображеннях.

Серед існуючих методів розпізнавання образів, можна виділити метод Віоли-Джонса. Метод, використовуючи певні маски, звані ознаками Хаара, шукає закономірності на зображеннях. Серед недоліків цього методу можна виділити той факт, що метод коректно працює лише з обличчями, розміщеними в анфас, що робить таку систему недостатньо гнучкою.

Також задачі класифікації вирішуються за допомогою самоорганізовних карт Кохонена. Такі карти в процесі зменшення розмірності зображення, отриманого на вході, виділяють певні ознаки і за допомогою них виконують класифікацію зображень. Головним недоліком такого підходу є залежність результату від вибору початкових налаштувань системи (вагових коефіцієнтів, функції сусідства).

Таким чином, через недостатню гнучкість класифікації для вирішення задач, що розглядаються у роботі, було обрано саме глибинні нейронні мережі.

Метою кваліфікаційної магістерської роботи є дослідження та розробка систем комп'ютерного зору на основі глибинних нейронних мереж.

Об'єктом дослідження є глибокі нейронні мережі, використовувані для задач класифікації зображень

Предметом дослідження є методи вирішення задач комп'ютерного зору, зокрема задач, пов'язаних з розпізнаванням образів.

Для створення глибинних нейронних мереж для вирішення задач розпізнавання образів було використано переднавчені згорткові нейронні мережі різних архітектур, для навчання яких були застосовані механізми переносу знань.

1.4 Постановка задачі дослідження

На сьогодні серед актуальних проблем постає проблема поширення вірусних захворювань. Медичні заклади обмежують можливість перебування на території людей, які не виконуються санітарні вимоги. Так, наприклад, у Харківській терапевтичній лікарні №7 біля входу розташовано пристрій, який зчитуючи зображення з камери відкриває двері лише за умови, що в об'єктиві камери знаходиться людина у коректно одягненій масці. До явних недоліків такої системи з точки зору користувача можна віднести головну особливість – обличчя повинне знаходитись досить близько до камери та бути нерухомим, а також система іноді вирішує, що одягнена не підборіддя маска влаштовує вимогам і пропускає людину.

Таким чином в рамках даної роботи було вирішено розробити аналог такої системи з урахуванням недоліків. Важливо виділити три класи для створюваної системи, які б визначали також і випадок некоректно одягненої маски, а також використовувати вибірку з різними відстанями між об'єктом класифікації та об'єктивом камери. Таке рішення збільшить можливий проміжок відстані та буде класифікувати обличчя більш гнучко.

Для дослідження пропонується застосувати декілька архітектур згорткових нейронних мереж та виконати їх порівняння, щоб визначити кращу з них, з використанням різних метрик оцінки якості класифікаторів.

Отже, було поставлено наступну задачу для даної роботи: зробити огляд існуючих методів для розв'язання задач комп'ютерного зору, оцінити їх з точки зору поставленої задачі класифікації зображень на три класи: люди в масках, люди без масок та люди з невірно одягненими масками. Розробити рішення на основі передтренуваних згорткових нейронних мереж та перенавчити його для заданої задачі за допомогою сучасних моделей. Виконати аналіз результатів навчання з урахуванням множини метрик оцінки та обрати серед існуючих архітектур найбільш точний класифікатор.

2 МЕТОДОЛОГІЧНІ ОСНОВИ ВИРІШЕННЯ ЗАДАЧ КОМП'ЮТЕРНОГО ЗОРУ

2.1 Згорткові нейронні мережі

Найкращою архітектурою штучних нейронних мереж для розпізнавання зображень є архітектура згорткових нейронних мереж. Вона використовує деякі особливості людського мозку, особливо зорову кору. У 1962 році експеримент групи американських нейрофізіологів виявив, що одна нервова клітина мозку реагує (або активується) лише візуальним сприйняттям межі в певному напрямку. Наприклад, деякі нейрони активуються, коли вони сприймають вертикальні межі, а інші – горизонтальні або діагональні. Нейрони, відповідальні за це, зосереджені у вигляді стрижневої архітектури для спільного формування візуального сприйняття. Найпростіший приклад у повсякденному житті — визначення унікальних ознак та виведення висновків про об'єкти на основі цих ознак: якщо можна побачити дві круглі деталі внизу об'єкта – це велосипед. Ця ідея лягла в основу роботи французького вченого Яна Лекуна, який представив архітектуру згорткових нейронних мереж у 1988 році [12].

Вхід згорткової нейронної мережі складається з цифрової матриці (якщо зображення у відтінках сірого) або набору матриць (зображення розділене на три шари: R-, G- і B- канали). Фільтр відповідає за виявлення певних характеристик зображення – це теж матриця, але з шуканими значеннями. Вибір архітектури згорткової нейронної мережі для розпізнавання активності має визначатися складністю обробки шарів, кількістю шарів і швидкістю отримання результатів. Виходячи з цього, слід вибрати найкращий алгоритм з найменшими втратами швидкості.

Згорткова нейронна мережа складається з декількох шарів, кількість шарів залежить від її архітектури та задачі, що розв'язується [13]. Перший шар завжди є згортковим шаром, від якого цей тип мережі отримав свою

назву. Це так звана карта ознак – матриця результату, яка є результатом операції згортки матриці фільтра та частини матриці зображення, а розмір еквівалентний матриці фільтра (рисунок 2.1). Кожен елемент результату обчислюється як скалярний добуток матриці фільтра та підматриці такого ж розміру (частини зображення), використовуючи такий вираз:

$$C_{i,j} = \sum_{u=0}^{m_x-1} \sum_{v=0}^{m_y-1} A_{i+u,j+v} B_{u,v}, \quad (2.1)$$

де A і B – матриці розмірністю $n_x \times n_y$ та $m_x \times m_y$ відповідно;

C – матриця розмірністю $(n_x - m_x + 1) \times (n_y - m_y + 1)$.

Чим більше значення у матриці результату, тим більша схожість між вихідним зображенням і фільтром (рисунок 2.1).

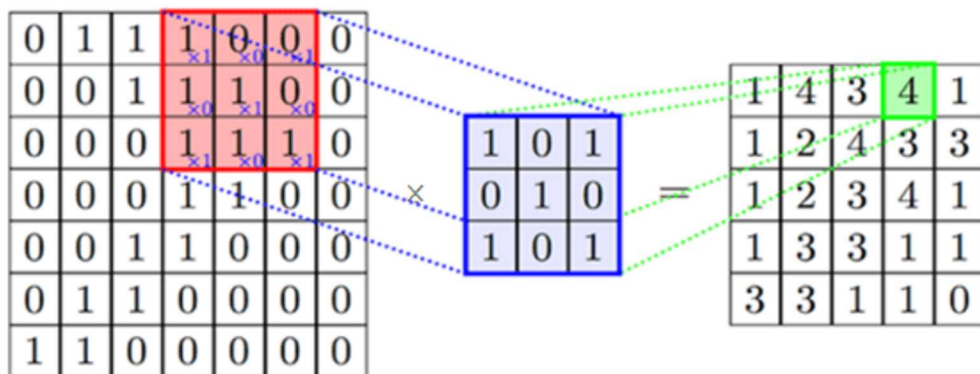


Рисунок 2.1 – Приклад згортки матриць розмірністю 7×7 та 3×3

Важливо розуміти, що кількість каналів фільтра має відповідати кількості каналів вхідного зображення. Тобто при обробці трьох матриць зображень має бути три матриці для кожного каналу.

Кількість карт визначається вимогами завдання. Якщо взяти велику кількість карт, покращиться якість розпізнавання, але збільшиться складність обчислень [14]. У більшості випадків рекомендується

співвідношення один до двох: кожне зображення попереднього шару (наприклад, перший згортковий шар, попереднім є вхідний шар) пов'язане з двома зображеннями згорткового шару (рисунок 2.2).

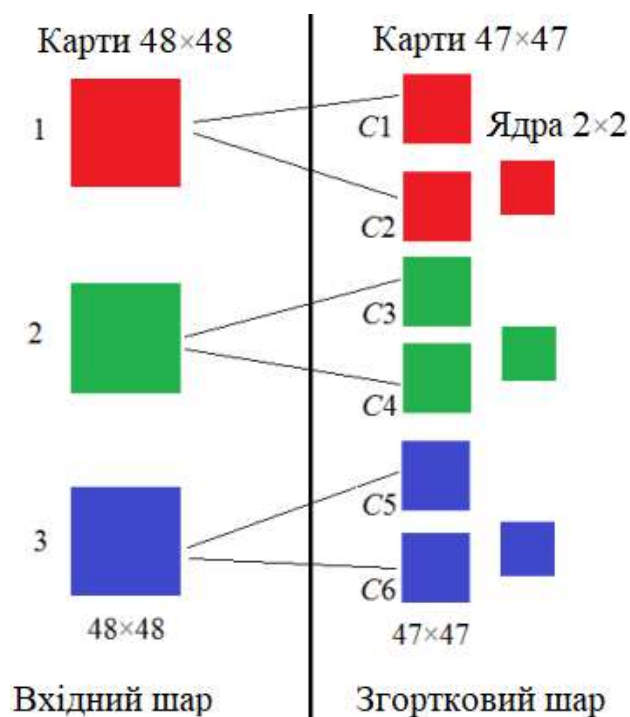


Рисунок 2.2 – Зв'язок кількості карт між згортковим і вхідним шаром

Чим більше згорткових шарів проходить зображення і чим далі воно переміщається по мережі, тим складніші особливості визначаються в картах активації. Шар підвибірки, як і шар згортки, також має карти, але їх кількість така ж як і на попередньому (згортковому) шарі. Мета цього шару – зменшити розмірність карт попереднього шару. Якщо після попередньої операції згортки були відображені деякі ознаки, то немає необхідності виконувати таку детальну обробку зображення далі, тому воно ущільнюється до менш детального. Крім того, фільтрація непотрібних частин допомагає не перевчитися.

У процесі сканування ядром підвибіркового шару карти попереднього шару, скануюче ядро не перетинається на відміну від

згорткового шару. Найчастіше кожна карта має ядро розміром, що дозволяє зменшити попередні карти згорткового шару в 2 рази. Вся карта ознак поділяється на комірки елемента, з яких вибираються максимальні значення. Також найчастіше у підвибірковому шарі застосовується функція активації ReLU [15]. Завдяки ній картина, що формується за допомогою операції згортки, зазнає деяких змін, що дозволяє нейронній мережі чіткіше оцінювати ситуацію.

У процесі сканування ядром шару підвибірки карти попереднього шару, ядра сканування не перетинаються, як згорткові шари. У більшості випадків розмір ядра кожної карти становить 2×2 , що дозволяє зменшити попередню карту в згортковому шарі в 2 рази. Вся карта об'єктів розділена на комірки елементів 2×2 , з яких вибирається максимальне значення. Функція активації ReLU також часто використовується на рівні підвибіркового шару [15]. Завдяки цьому зображення, утворене операцією згортки, зазнає деяких змін, що дозволяє нейронній мережі більш чітко оцінити ситуацію.

Останній із типів шарів – повнозв'язний шар. У більшості випадків це багатошаровий перцептрон. Метою цього шару є класифікація, що виконує моделювання складної нелінійної функції та оптимізує її для покращення якості розпізнавання. Після виконання згортки зображення декілька раз та ущільнення за допомогою підвибірки система перебудовується від конкретної сітки пікселів з високою роздільною здатністю до більш абстрактних карт ознак. Як правило, на кожному наступному шарі кількість каналів збільшується, а розмірність зображення в кожному каналі зменшується. Нарешті, залишається великий набір каналів, які зберігають невелику кількість даних (навіть один параметр), які інтерпретуються як абстрактні поняття, знайдені в оригінальному зображенні.

Ці дані об'єднуються і передаються на звичайну повнозв'язну нейронну мережу, яка також може складатися з кількох шарів. У цьому

випадку повнозв'язний шар втрачає просторову структуру пікселів і має відносно невеликий розмір (відносно кількості пікселів у вихідному зображенні). На рисунку 2.3 представлена узагальнена схема роботи нейронної мережі з багатошаровим персептроном.

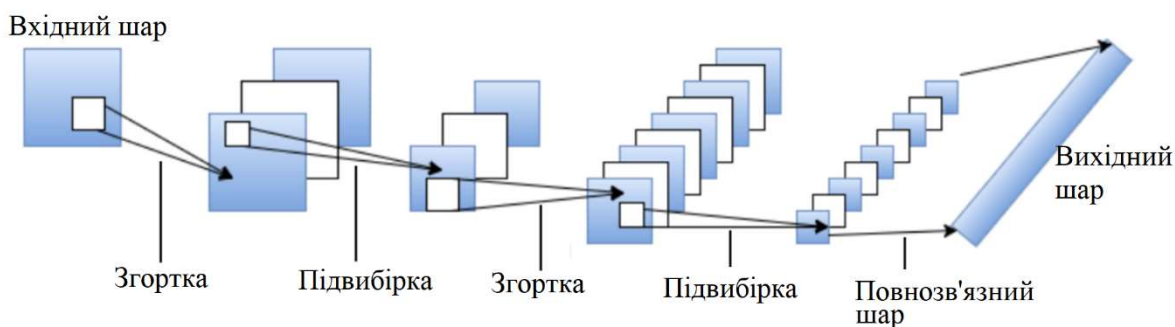


Рисунок 2.3 – Загальна архітектура згорткової мережі

Вихідний шар відповідає за формування ймовірності того, що вхідне зображення належить до певної категорії (з деякою вірогідністю). Для досягнення цього вихідний шар повинен містити кількість нейронів, що відповідає кількості класів. Використовується функція активації для подальшої зміни зваженого та підсумованого сигналу. Навчання проводиться класичним способом зворотного поширення помилки[12].

2.2 Навчання згорткової нейронної мережі

Навчання згорткової мережі [16] здійснюється за допомогою алгоритму зворотного поширення помилки, який використовується для навчання багатошарового персептрону. Складність навчання згорткових мереж полягає в тому, що в згортковому шарі нейрони мають роздільні ваги.

У багатошаровому персептроні можна визначити помилку нейрона j як:

$$\delta_j^l = \frac{\partial C}{\partial z_j^l}, \quad (2.2)$$

де z_j^l :

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l, \quad (2.3)$$

де $a_j^l = \sigma(z_j^l)$;

σ – функція активації.

Але оскільки розглядається згортковий шар, а не багат шаровий перспетрон, то замість z_j маємо $z_{x,y}$ і замість перемноження матриць виконується згортка:

$$z_{x,y}^{l+1} = w_{x,y}^{l+1} * \sigma(z_{x,y}^l) + b_{x,y}^{l+1} = \sum_a \sum_b w_{a,b}^{l+1} \sigma(z_{x-a,y-b}^l) + b_{x,y}^{l+1} \quad (2.4)$$

Помилка поширюється на попередній шар за такою формулою:

$$\begin{aligned} \frac{\partial C}{\partial z_{x,y}^l} &= \sum_{x'} \sum_{y'} \frac{\partial C}{\partial z_{x',y'}^{l+1}} \frac{\partial z_{x',y'}^{l+1}}{\partial z_{x,y}^l} \\ &= \sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} \frac{\partial \left(\sum_a \sum_b w_{a,b}^{l+1} \sigma(z_{x'-a,y'-b}^l) + b_{x',y'}^{l+1} \right)}{\partial z_{x,y}^l} \end{aligned} \quad (2.5)$$

Виконавши математичні перетворення отримаємо:

$$\sum_{x'} \sum_{y'} \delta_{x',y'}^{l+1} w_{x'-x,y'-y}^{l+1} \sigma'(z_{x,y}^l) = \delta_{x,y}^{l+1} * w_{-x,-y}^{l+1} \sigma'(z_{x,y}^l) \quad (2.6)$$

Оновлення ваги відбуватиметься за такою формулою:

$$\begin{aligned}
\frac{\partial C}{\partial w_{a,b}^l} &= \sum_x \sum_y \frac{\partial C}{\partial z_{x,y}^l} \frac{\partial z_{x,y}^l}{\partial w_{a,b}^l} \\
&= \sum_x \sum_y \delta_{x,y}^l \frac{\partial \left(\sum_{a'} \sum_{b'} w_{a',b'}^l \sigma(z_{x-a',y-b'}^l) + b_{x,y}^l \right)}{\partial w_{a,b}^l} \\
&= \sum_x \sum_y \delta_{x,y}^l \sigma(z_{x-a,y-b}^{l-1}) = \delta_{a,b}^l * \sigma(z_{-a,-b}^{l-1}) \\
&= \delta_{a,b}^l * \sigma(\text{ROT180}(z_{a,b}^{l-1}))
\end{aligned} \tag{2.7}$$

Узагальнивши формули, отримаємо, що помилка поширюється за формулою:

$$\delta_{x,y}^l = \delta_{x,y}^{l+1} * \text{ROT180}(w_{x,y}^{l+1}) \sigma'(z_{x,y}^l) \tag{2.8}$$

А оновлення ваг:

$$\frac{\partial C}{\partial w_{a,b}^l} = \delta_{a,b}^l * \sigma(\text{ROT180}(z_{a,b}^{l-1})) \tag{2.9}$$

де * – операція згортки,

$\text{ROT180}(w_{x,y}^{l+1}) = w_{-x,-y}^{l+1}$, ядро необхідно повернути на 180 градусів.

Шар підвибірки зазвичай не містить параметрів для навчання, тому помилка переходить до попереднього шару. Якщо в якості функції підвибірки використовується Max , то $\sigma_{x_{\max}, y_{\max}}^l = \sigma^{l+1}$, а помилка інших нейронів дорівнює 0.

У повнозв'язних шарах поширення помилки та корекція ваг відбувається подібно до багат шарового перцептрона [17].

2.3 Розширення навчальної множини (data augmentation)

Для навчання нейронної мережі важливо, щоб навчальні вибірки були достатньо збалансованими та репрезентативними [18]. Рисунок 2.4 демонструє приклад зображення задачі розпізнавання символів – у першому стовпці, так зване «ідеальне» зображення, в інших стовпцях – деформовані, але такі зустрічаються на практиці.



Рисунок 2.4 – Приклад деформованих зображень

Класифікатор, навчений лише на зображеннях, близьких до ідеалу, не бачить усіх можливих викривлень, і він буде помилятися на сильно спотворених зображеннях. Недостатня кількість даних призведе до перенавчання моделі. Щоб вирішити цю проблему на рівні моделі, пропонується використовувати коригування або зменшення кількості параметрів. На рівні роботи із даними найчастіше використовується аугментація [19].

Аугментація даних полягає у штучному збільшенні розміру вибірки шляхом перетворення даних. Її використання може зменшити перенавчання [20]. Перенавчання – це явище, при якому модель підбирає більшість

параметрів для вирішення задачі на наборі навчальних даних на шкоду узагальненню інформації.

Для перетворення зображень використовуються:

- масштабування;
- поворот;
- зміщення;
- зміна яскравості тощо.

З метою забезпечення стійкості моделі до просторових змін зображення (масштабування, вертикального або горизонтального зміщення, повороту), навчальні вибірки були штучно збільшені за рахунок вищезгаданої модифікації вихідних даних (аугментації зображення). Щоб не зберігати додаткові файли, набір даних розширюється в процесі навчання моделі. Для цього використовується ImageDataGenerator з бібліотеки Keras. Ідентифікація випадкової модифікації фотографій і масок забезпечується шляхом фіксації породжуючого елемента (seed) генератора псевдовипадкових чисел.

2.4 Метрики

Вибір правильних показників важливий для оцінки моделей машинного навчання. Для оцінки моделей машинного навчання в різних програмах можна використовувати різні показники. У цьому розділі описано ключові показники, щоб краще зрозуміти їх і визначити задачі, для яких вони можуть бути використані. У деяких випадках використання єдиного показника може не повністю відобразити проблему, яку необхідно вирішити, у зв'язку з цим необхідно визначити найбільш об'єктивну комплексну оцінку якості. 22

Задача класифікації поділяється на бінарні (якщо цільовий клас може приймати два значення) і мультикласові. У задачі класифікації навчальні приклади найкраще розподілити відносно рівномірно між класами [22], але

насправді майже всі набори даних у задачі бінарної класифікації є незбалансованими, що призводить до проблеми якісної оцінки класифікатора. Класична метрика акуратності (accuracy) може дати високе значення, причому фактично ефект моделі гірший, ніж передбачення, яке завжди визначає один клас.

Для оцінки якості розробленої нейромережі використано метрики оцінки правильності класифікації. Серед них:

- доля вірно знайдених об'єктів;
- точність;
- повнота;
- AUC.

Для оцінки метрик використовуються наступні значення: n_{TP} (True Positive) – число вірно виявлених об'єктів заданого класу, n_{FP} (False Positive) – кількість помилкових спрацьовувань, n_{FN} (False Negative) – кількість невиявлених об'єктів (пропусків) [23].

Для метрики точності (precision) використовується наступна формула:

$$precision = \frac{n_{TP}}{n_{TP} + n_{FP}}. \quad (2.10)$$

Для метрики повноти:

$$recall = \frac{n_{TP}}{n_{TP} + n_{FN}}. \quad (2.11)$$

Одним із способів оцінити модель загалом, не прив'язуючись до конкретного порога, є AUC-ROC (або ROC AUC) – площа (Area Under Curve) під кривою помилок (Receiver Operating Characteristic curve).

ROC (receiver operating characteristic) – графік, що показує залежність правильно класифікованих об'єктів позитивного класу від хибно позитивно класифікованих об'єктів негативного класу. Іншими словами, співвідношення True Positive Rate (Recall) та False Positive Rate.

True Positive Rate (TPR) показує, який відсоток серед усіх зображень класу правильно передбачений моделлю.

$$TPR = \frac{n_{TP}}{n_{TP} + n_{FN}} \quad (2.12)$$

False Positive Rate (FPR): який відсоток серед усіх зображень інших класів неправильно передбачений моделлю.

$$FPR = \frac{n_{FP}}{n_{FP} + n_{TN}} \quad (2.13)$$

На підставі ROC розраховується показник AUC (Area under the curve), який є загальною площею під кривою залежності вірних та невірних передбачень класифікаційної моделі. Чим ближче до 1 отримані значення, краще система розпізнає класи.

Усі наведені вище метрики використовуються у випадку бінарної класифікації. Для мультикласової класифікації використовується середнє значення для всіх класів окремо.

3 НАВЧАННЯ ПЕРЕДТРЕНОВАНИХ НЕЙРОННИХ МЕРЕЖ

3.1 Передтреновані нейронні мережі

Використання глибоких нейронних мереж у задачах навчання з учителем успішно зайняло місце в комп'ютерному зорі, мовленні та широкому спектрі задач обробки природної мови. Глибокі нейронні мережі показали свою перевагу в таких ключових задачах, наприклад таких, як задачі класифікації зображень [24]. Зокрема, розвиток архітектур згорткових нейронних мереж зменшив рівень помилок класичного набору даних ImageNet, що використовується для завдань класифікації зображень, він містить понад 14 мільйонів зображень, позначених на 1000 категорій. Таким чином, AlexNet [25] досяг майже 50% зниження частоти помилок у 2012 році, а потім цей показник був покращений в архітектурах VGG [26], ResNet [27] та DenseNet [28] у 2014 році. Тому сучасна практика вирішення задач комп'ютерного зору базується на використанні глибинних нейронних мереж, що є більш перспективним методом вирішення задач виділення ознак. Системи вищезгаданих архітектур згорткових нейронних мереж дуже різні, але в цілому їх можна розділити на кілька функціональних блоків:

- блок згорткових шарів;
- блок шарів об'єднання;
- блок повнозв'язних шарів.

У більшості випадків виходом нейронної мережі задачі класифікації є ймовірність приналежності певному класу для вхідного зображення, отриманого за допомогою функції softmax (softmax), яка є узагальненням логістичної функції в багатовимірному випадку. Мережа, попередньо навчена на наборі даних ImageNet, має параметри повнозв'язних блоків шарів, які визначено оптимальними для вирішення задачі поділу конкретних класів набору даних. Це зазвичай не дає цінної інформації для задачі кластеризації у загальному вигляді, в якій зображення може не мати

класу. За цих умов найкращим рішенням є використання вектора ознак з входу блоку повнозв'язних шарів. По суті, це буде набір відповідей ознакових фільтрів згорткових шарів [24].

3.2 Архітектура VGG16

VGG16 – це складна нейронна мережа, яка містить 16 шарів. Мережа і двох частин: Перша частина мережі виділяє характерні ознаки у зображенні. Складається з блоків, що чергуються, згортки і підвиборки. [30]

Спочатку йдуть два блоки згортка-згортка-підвибірка, а потім три блоки згортка-згортка-згортка-підвибірка. Розмір вузла згортки у всіх шарах 3x3. При підвибірці проводиться вибір максимального значення квадрата розміром 2x2 [30]. Друга частина відповідає за класифікацію об'єкта на зображенні за виділеними на попередньому етапі ознаками. За умовчанням ця частина містить 3 повнозв'язкові шари. У перших двох шарах 4096 нейронів, а в останньому – 1000 нейронів (за кількістю класів об'єктів) [30]. На вхід мережа VGG16 отримує зображення розміром 229x229 пікселя, 3 канали кольору (червоний, зелений і синій). На виході мережа видає ймовірності, що у зображенні той чи інший клас об'єкта [30]. У мережі є 1000 класів зображень. Структура мережі представлена рисунку 3.1.



Рисунок 3.1 – Архітектура мережі VGG16

3.3 Архітектура Xception

Xception, екстремальна версія Inception, є архітектурою, заснованою на розділених по глибині згорткових шарах. З цією модифікованою роздільною згорткою по глибині це навіть краще, ніж InceptionV3 [31]. Xception запропонував не хто інший, як сам Франсуа Шолле, творець і адміністратор бібліотеки Keras. Стандартні модулі Inception замінені архітектурою Xception, яка є розширенням архітектури Inception [32].

Це проста і модульна архітектура. На рисунку 3.2 зображено архітектуру такої мережі.

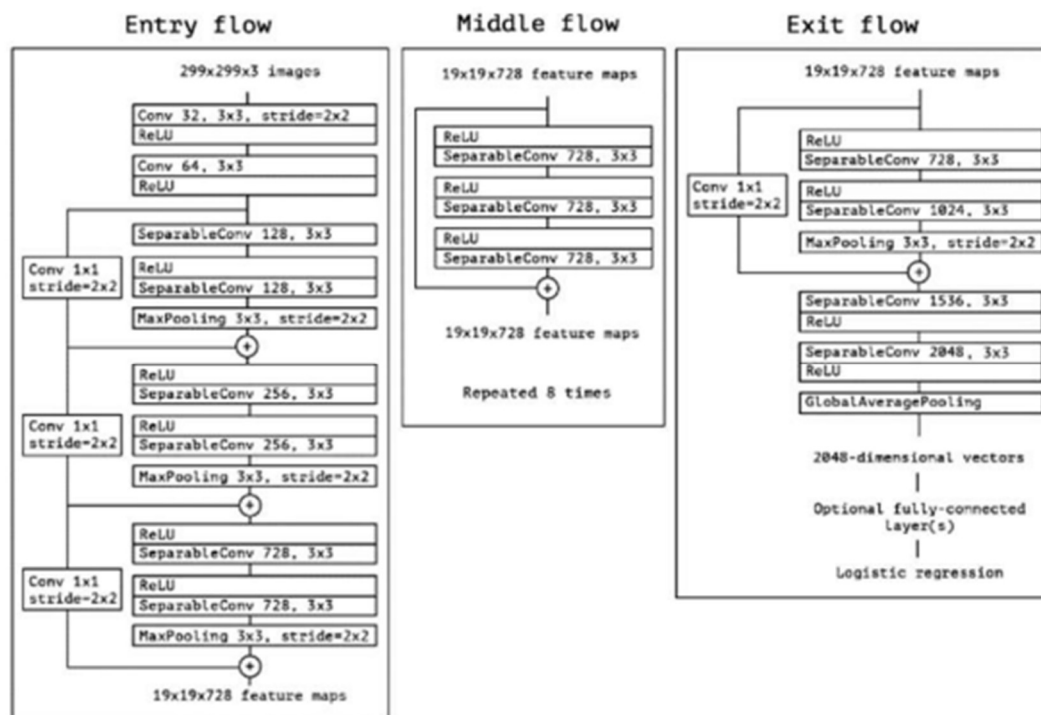


Рисунок 3.2 – Архітектура Xception

Архітектура має 36 шарів згортки, які утворюють основу для вилучення функцій з мережі. Ці 36 шарів структуровані на 14 модулів, усі з яких, за винятком першого та останнього модулів, мають залишкові лінійні зв'язки навколо них. Роздільні по глибині шари згортки складатимуть

лінійну стопку із залишковими з'єднаннями. В результаті легше визначити та змінити архітектуру.

При використанні бібліотек високого рівня, таких як Keras або TensorFlow Slim, потрібно лише від 30 до 40 рядків коду, подібно до архітектури VGG16, але на відміну від архітектур, таких як Inception V2 або Inception V3, які набагато складніше визначити.

Як видно з рисунку 3.2, модифікована глибоко розділена згортка, яку можна побачити, є SeparableConv. Ці SeparableConvs розміщуються в архітектурі глибокого навчання і розглядаються як початкові модулі. Також є залишкові з'єднання (або ярлики/пропуски), розміщені для всіх потоків, спочатку запропонованих ResNet [32].

3.4 Архітектура MobileNet

MobileNet – передова архітектура згорткової нейронної мережі, що дозволяє розпізнавати об'єкти на зображенні з високою точністю та низьким споживанням ресурсів. Розроблено Google. Дана архітектура знаходиться у відкритому доступі, що уможливорює її використання у випускній кваліфікаційній роботі. Проект підтримується та активно розвивається. На даний момент випущено версію MobileNetV2. Програмне забезпечення, що використовує цю архітектуру, має низькі системні вимоги і тому може виконуватися на пристроях з обмеженими ресурсами, такими як смартфони або комп'ютери (наприклад, Raspberry PI). Архітектура MobileNets відрізняється від традиційних згорткових нейронних мереж використанням глибинної згортки. Загальна ідея глибинної сепарабельної згортки полягає в тому, щоб розбити згортку на дві стадії: спочатку виконується глибинна згортка 3×3 ; потім слідує потокова згортка 1×1 .

Дане рішення дозволяє зменшити фактичну кількість параметрів нейронної мережі. На рисунку 3.3 ліворуч зображено блок звичайної згорткової мережі, а праворуч – базовий блок MobileNet

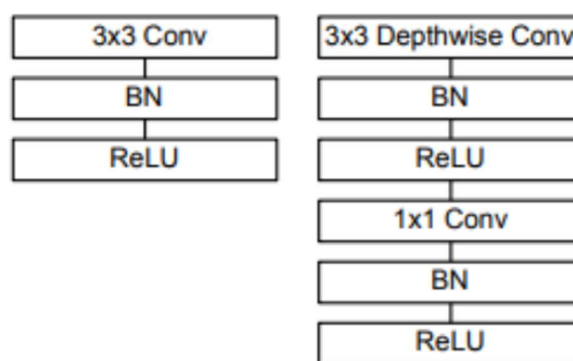


Рисунок 3.3 – Блок MobileNet

Згорткова частина мережі MobileNet складається з одного звичайного згорткового шару зі згорткою 3×3 на початку і тринадцяти блоків, зображених праворуч на рисунку, з числом фільтрів, що поступово збільшується, і просторової розмірністю тензора, що знижується. Особливістю даної архітектури є відсутність max pooling-шарів. Замість них для зниження просторової розмірності використовується згортка з параметром stride, що дорівнює 2. Двома гіперпараметрами архітектури MobileNet є α (множник ширини) і ρ (множник глибини або множник роздільної здатності). Множник ширини відповідає за кількість каналів у кожному шарі. Наприклад, $\alpha=1$ дає нам архітектуру, описану вище, а $\alpha=0.25$ — архітектуру із зменшеним у чотири рази числом каналів на виході кожного блоку. Чисельник дозволу відповідає за просторові розміри вхідних тензорів. Наприклад, $\rho=0.5$ означає, що висота та ширина feature map, що подається на вхід кожному шару буде зменшена вдвічі. Обидва параметри дозволяють змінювати розміри мережі. При зменшенні α і ρ знижується точність розпізнавання, але водночас збільшується швидкість роботи та зменшується споживана пам'ять [33].

MobileNetV2 – наступне покоління нейромереж цього сімейства, яке дозволяє досягати приблизно такої ж точності розпізнавання при ще більшій

швидкості роботи. Основний блок цієї мережі загалом схожий на попереднє покоління, але має низку ключових особливостей (рисунок 3.4).

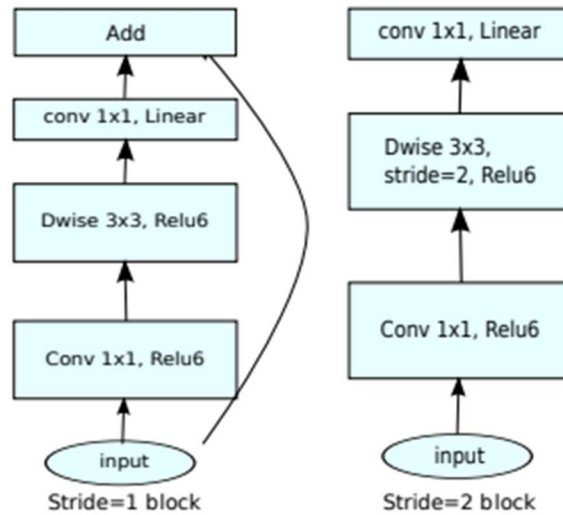


Рисунок 3.4 – Відмінність між MobileNetV1 та MobileNetV2

Як і в MobileNetV1, тут є блоки згортки з кроком 1 (на рисунку зліва) і з кроком 2 (на рисунку справа). Блоки з кроком 2 призначені для зниження просторової розмірності тензора і, на відміну блоку з кроком 1, немає residual connections.

Блок MobileNet, званий авторами розширювальним блоком згортки, складається з трьох шарів [34]:

- спочатку йде pointwise convolution з великою кількістю каналів, званий expansion layer.

На вході цей шар приймає тензор розмірності $D_f * D_f * C_{in}$, а на виході видає тензор $D_f * D_f * (t * C_{in})$, де t – новий гіперпараметр, названий рівнем розширення expansion factor). Цей шар створює відображення вхідного тензора у просторі великої розмірності;

- потім йде depthwise convolution з ReLU6-активацією. Цей шар разом з попереднім, по суті, утворює вже знайомий блок MobileNetV1.

На вході цей шар приймає тензор розмірності $D_f * D_f * (t * C_{in})$, а на виході видає тензор $\left(\frac{D_f}{s}\right) * \left(\frac{D_f}{s}\right) * (t * C_{in})$, де s – крок згортки (stride), адже depthwise convolution не змінює кількість каналів;

– наприкінці йде 1×1 -згортка з лінійною функцією активації, що знижує кількість каналів.

На вході такий шар приймає тензор розмірності $\left(\frac{D_f}{s}\right) * \left(\frac{D_f}{s}\right) * (t * C_{in})$, а на виході видає тензор $\left(\frac{D_f}{s}\right) * \left(\frac{D_f}{s}\right) * C_{out}$ де C_{out} — кількість каналів на виході блоку.

Загальний вигляд архітектури наведено на рисунку 3.5.

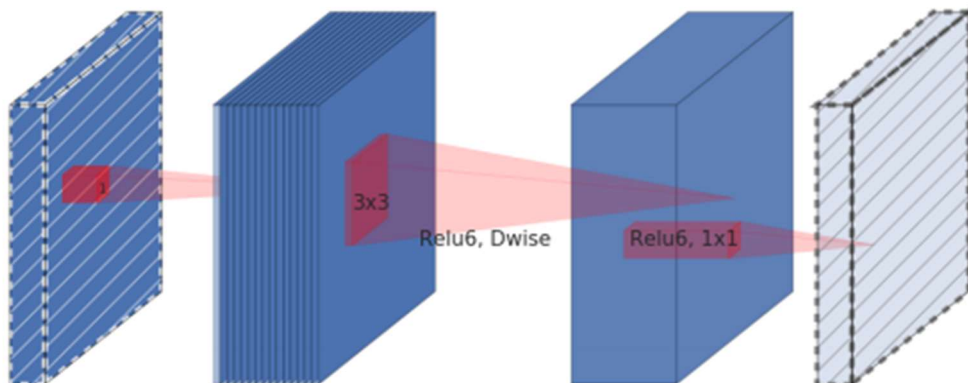


Рисунок 3.5 – Архітектура MobileNetV2

Фактично, саме третій шар у цьому блоці, званий bottleneck layer, є основною відмінністю другого покоління MobileNet від першого [34].

3.5 Transfer learning

Людський мозок може передавати знання між задачами. Так людина застосовує відповідні знання зі свого досвіду, коли стикається з новими завданнями, подібними до тих, з якими мала справу раніше. Причому, чим

ближче завдання до попереднього досвіду, тим легше людині буде з ним впоратися.

Навпаки, основні алгоритми машинного навчання (traditional machine learning) розраховані на певні специфічні задачі: класифікація котів і собак, класифікація чисел тощо. Схему показано на рисунку 3.6 (ліворуч).

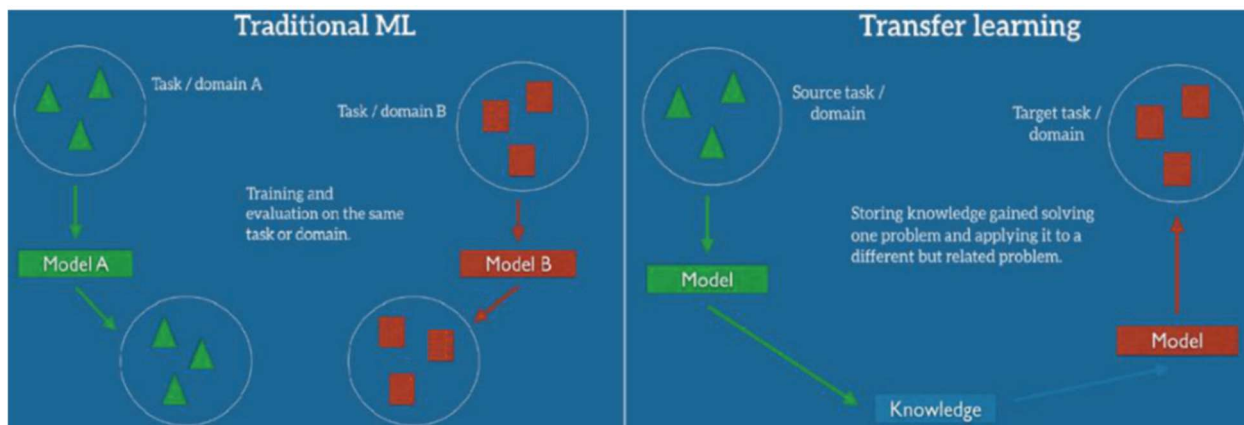


Рисунок 3.6 – Відмінність звичайного машинного навчання та переносу навчання

Transfer learning, або феномен передачі знань – спроба змодельовати цю характеристику людського мозку в математиці. Концепція передачі знань полягає в тому, щоб передати знання, отримані в одному або кількох початкових завданнях, і використовувати їх для покращення навчання в поточному завданні. Схему показано на рисунку 3.6 (праворуч). Завдяки цьому машинне навчання може бути настільки ж ефективним, як і навчання людини [35].

Метою трансферного навчання є використання знань, отриманих раніше з вихідного завдання, для підвищення якості навчання поточного (цільового) завдання. Тому передача знань може мати позитивний вплив на такі показники навчання:

– початкова продуктивність, що може бути досягнута в поточному завданні, з використанням апріорної інформації (передачу знань) для вибору початкових параметрів моделі;

– час, необхідний для навчання поточного завдання, враховуючи передані дані;

– кінцевий рівень продуктивності, якого можна досягти в поточному завданні.

При використанні методу передачі знань виникають ситуації, коли вихідне завдання не пов'язане тісно з поточним завданням або цей зв'язок не враховується повністю. Тоді продуктивність може залишатися на тому ж рівні, а потім і знижуватися. Це явище називається негативним обміном (negative transfer). Тому одним із завдань дослідників у сфері трансферного навчання є отримання позитивних результатів у передачі знань між подібними завданнями та усунення негативного обміну між непов'язаними завданнями. Трансферне навчання має особливість, яка відрізняється від класичного багатозадачного навчання нейронних мереж: воно може передавати інформацію строго від початкової задачі до поточної, тоді як у багатозадачному навчанні різні моделі можуть вирішувати інформацію в будь-якому напрямку.

Використання трансферного навчання для підготовки попередньо навчених згорткових нейронних мереж можна здійснити двома способами [36]:

1) finetuning the convnet (тонка настройка мережі): замість довільної ініціалізації мережі беруть ваги передбаченої на великій вибірці мережі. У процесі навчання не тільки перенавчається класифікатор для нового набору даних, але також відбувається точне налаштування ваги мережі за допомогою зворотного розповсюдження помилки;

2) convnet as fixed feature extractor: при ініціалізації попередньої мережі «заморожують» ваги всіх шарів, крім останнього повнозв'язного

шару. Цей шар замінюється на новий із випадковими вагами, і тільки він навчається.

Стратегії перенесення навчання залежить від різних чинників, але найважливішими є два: розмір нового набору даних та її схожість із вихідним набором даних. Якщо врахувати, що характер роботи ДСНС більш універсальний на ранніх шарах і більш тісно пов'язаним з конкретним набором даних на наступних шарах, можна виділити чотири основні сценарії [37]:

– новий набір даних менший за розміром і аналогічний за змістом вихідного набору даних. Якщо обсяг даних невеликий, то немає сенсу проводити тонке налаштування ДСНС через надмірне припасування. Оскільки дані схожі з первісними, можна припускати, що відмінні риси в ДСНС будуть релевантними і для цього набору даних. Тому оптимальним рішенням є навчання лінійного класифікатора відмітною ознакою СНР;

– новий набір даних відносно великий та аналогічний за змістом вихідного набору даних. Оскільки у нас більше даних, можна не турбуватися про надмірне припасування, якщо ми спробуємо провести тонке налаштування всієї мережі;

– новий набір даних менший за розміром і суттєво відрізняється за змістом від вихідного набору даних. Оскільки обсяг даних невеликий, буде цілком достатньо лише лінійного класифікатора. Так як дані суттєво відрізняються, краще навчати класифікатор не з вершини мережі, де містяться конкретніші дані. Натомість краще навчити класифікатор, активувавши його на ранніх шарах мережі;

– новий набір даних відносно великий і суттєво відрізняється за змістом вихідного набору даних. Оскільки набір даних дуже великий, можна собі навчити всю ГСНС з нуля. Проте на практиці часто все одно виявляється вигідніше використовувати для ініціалізації ваг із заздалегідь навченої моделі. У цьому випадку ми будемо мати достатній обсяг даних для тонкого налаштування всієї мережі.

Так наприклад в роботі [38] використано метод тонкого налаштування ШНС. Останній фрагмент попередньо навченої мережі розморожується для навчання. На прикладі мережі VGG16 виділено останній блок переднавченої частини виділено на рисунку 3.7.



Рисунок 3.7 – Виділення блоків для тонкого налаштування

Як зазначають автори роботи, Точність роботи мережі на тестових даних становить 95,21% і зросла порівняно з мережею без тонкого налаштування на 0,83%. Невеликий приріст пояснюється тим, що новий набір даних не сильно відрізняється від набору даних ImageNet і згорткова частина нейронної мережі VGG16 добре виділяє характерні ознаки морських суден. Якби набір даних містив об'єкти, які не входять у вихідний набір даних для навчання, то ефективність тонкого налаштування мережі на такому наборі даних була б значно вищою.

Таким чином для імітаційного моделювання було обрано метод виділення ознак, тобто вирішено «заморозити» навчання усіх шарів мереж, окрім останніх (рисунок 3.8)

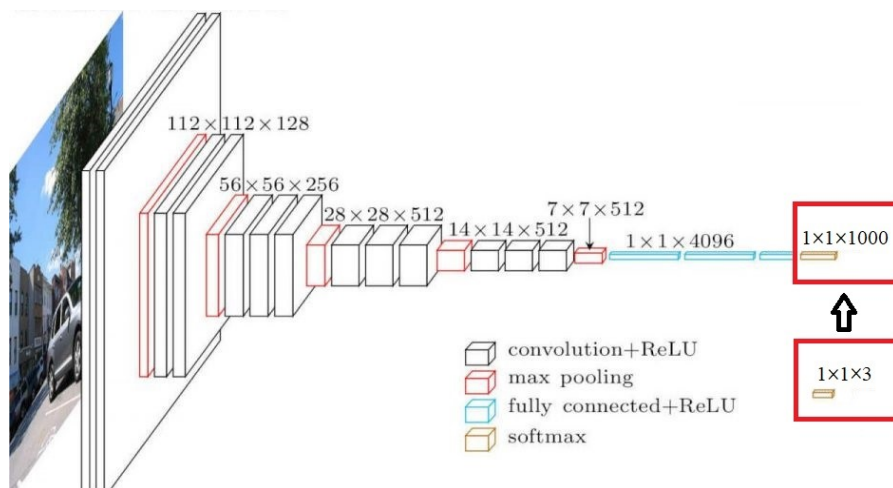


Рисунок 3.8 – Зміни в архітектурі ШНС на прикладі VGG16

Таке рішення дозволяє скоригувати вже навчену неймережу для конкретної задачі. В даному випадку важливим є корекція саме останнього шару так як для навчання усіх розглянутих в роботі переднавчених ШНС використовувалася база ImageNet, яка має 1000 класів об'єктів. В задачі, що вирішується в даній роботі необхідно розділяти зображення лише на три класи, тому останній шар мережі повинен містити лише три нейрони.

4 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ

4.1 Огляд набору даних та їх аугментація

В якості набору даних для навчання трьох нейронних мереж було обрано набір даних [39].

Перший набір даних містить 2079 зображень, розділених на три класи:

- обличчя без масок;
- обличчя з некоректно одягненими масками;
- обличчя з масками, що закривають рот і ніс.

Було створено дві вибірки для тренування та для тестування. Приклади цих зображень наведено на рисунках 4.1 – 4.2.



Рисунок 4.1 – Приклади зображень з тестової вибірки



Рисунок 4.2 – Приклади зображень з навчальної вибірки

Для збільшення вибірки, кожне зображення було оброблене з використанням ImageDataGenerator бібліотеки keras. Для обробки було визначено наступні параметри:

- rescale=1/255, зміна масштабу, необхідно, так як використовувані мережі можуть працювати з зображеннями розміром 255×255;
- zoom_range=0.3, множник випадкового збільшення зображення;
- rotation_range=30, відсоток максимального випадкового повороту;
- width_shift_range=0.2, shift_range=0.2, частка зменшення зображення в висоту та ширину;
- brightness_range=[0.4, 1.5], обмеження зміни яскравості зображення;
- horizontal_flip=True, чи повертати зображення по горизонтальній осі.

4.2 Навчання мереж

У наведеному в додатку А програмному коді для навчання мереж використовується файл train.py.

В цьому файлі визначено функцію, що навчає кожну з мереж. Перед виводом функції створюються вже навчені групи шарів для кожної мережі, їм задаються параметри: початкові ваги, що відповідають вагам навчених на датасеті ImageNet, та розміри зображень вхідних даних 224×224 з трьома каналами кольорів).

Після цього для кожного типу мереж викликається функція навчання.

Спочатку вона створює модель мережі на основі поданих значень. Вона вмикає можливість навчання для усіх навчених шарів, після чого додає шар класифікатору з трьома нейронами (кількість класів) з функцією активації softmax.

Після цього створюються об'єкти налаштування, що використовуються в процесі навчання – параметри збереження (зберігати тільки модель, що має найвищу точність) та параметри зупинки – якщо точність не збільшилась протягом восьми епох навчання, навчання зупиняється і мережа вважається навченою.

Модель компілюється з вказанням у параметрах метрик, які використовуватимуться для оцінки якості мережі.

Після компіляції модель навчається. В якості вхідних даних вказуються передоброблені набори даних для навчання та валідації. Серед інших параметрів – кількість епох навчання (обрано 200, так як використовується механізм ранньої зупинки) та створені об'єкти налаштування.

Після навчання моделі мереж зберігаються на диск.

4.2.1 Результати навчання моделі VGG16

Модель, подана на навчання, виглядає наступним чином (рисунок 4.3).

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14714688
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 3)	75267
=====		
Total params: 14,789,955		
Trainable params: 75,267		
Non-trainable params: 14,714,688		

Рисунок 4.3 – Модель VGG16

Навчання проводилось протягом 14 епох. Під час навчання було отримано наступні значення метрик оцінки, відображені на графіках на рисунку 4.4.

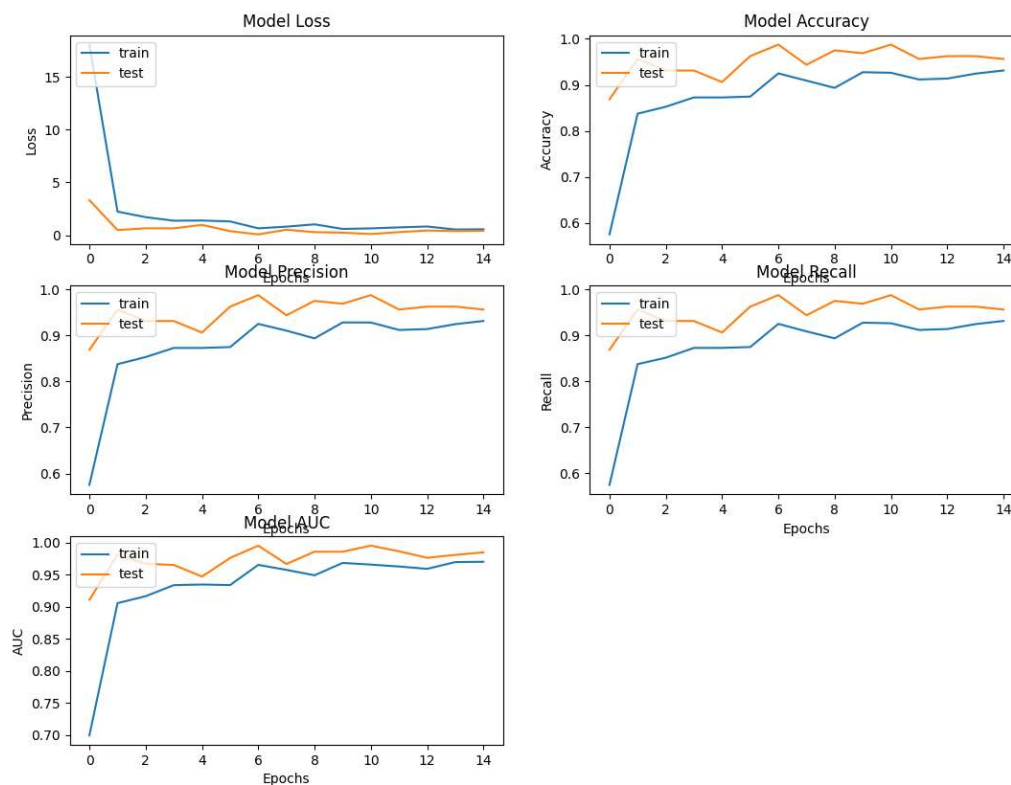


Рисунок 4.4 – Навчання мережі з архітектурою VGG16

З графіків можна зробити висновок, що хоча на першому кроці було вже отримане значення точності близько 0.8, мережа повинна була навчатися ще як мінімум 5 кроків для досягнення задовільної точності в 0.9.

4.2.2 Результати навчання моделі Xception

Модель, подана на навчання, виглядає наступним чином (рисунок 4.5).

```

=====
Layer (type)                Output Shape                Param #
=====
xception (Functional)      (None, 7, 7, 2048)         20861480
flatten (Flatten)          (None, 100352)             0
dense (Dense)               (None, 3)                   301059
=====

Total params: 21,162,539
Trainable params: 301,059
Non-trainable params: 20,861,480

```

Рисунок 4.5 – Модель Xception

Навчання проводилось потягом 41 епохи. Під час навчання було отримано наступні значення метрик оцінки, відображені на графіках на рисунку 4.6.

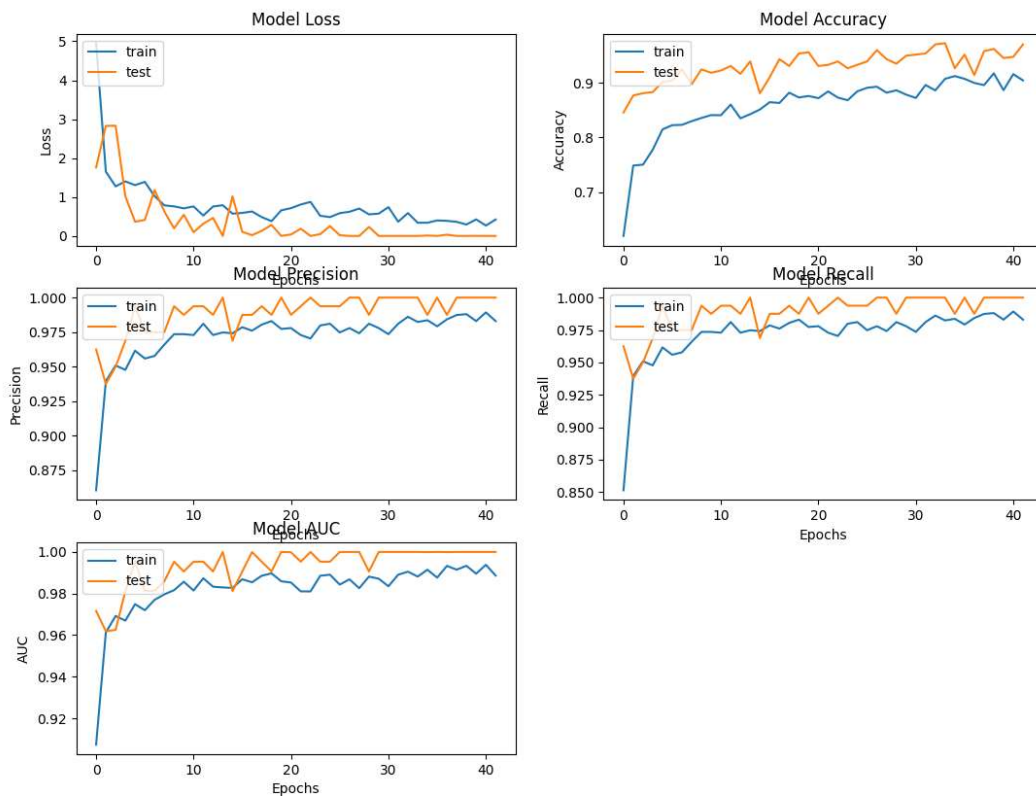


Рисунок 4.6 – Навчання мережі з архітектурою Xception

Як видно з графіків на другій епосі було отримано вже достатньо високі значення побічних метрик та значення точності приблизно 0,7. Після цього мережа навчалася для покращення точності, причому інші показники також в середньому підвищувались.

4.2.3 Результати навчання моделі MobileNetV2

Модель, подана на навчання, виглядає наступним чином (рисунок 4.7).

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2257984
flatten (Flatten)	(None, 62720)	0
dense (Dense)	(None, 3)	188163

=====
Total params: 2,446,147
Trainable params: 188,163
Non-trainable params: 2,257,984

Рисунок 4.7 – Модель MobileNetV2

Навчання проводилось потягом 16 епох. Під час навчання було отримано наступні значення метрик оцінки, відображені на графіках на рисунку 4.8.

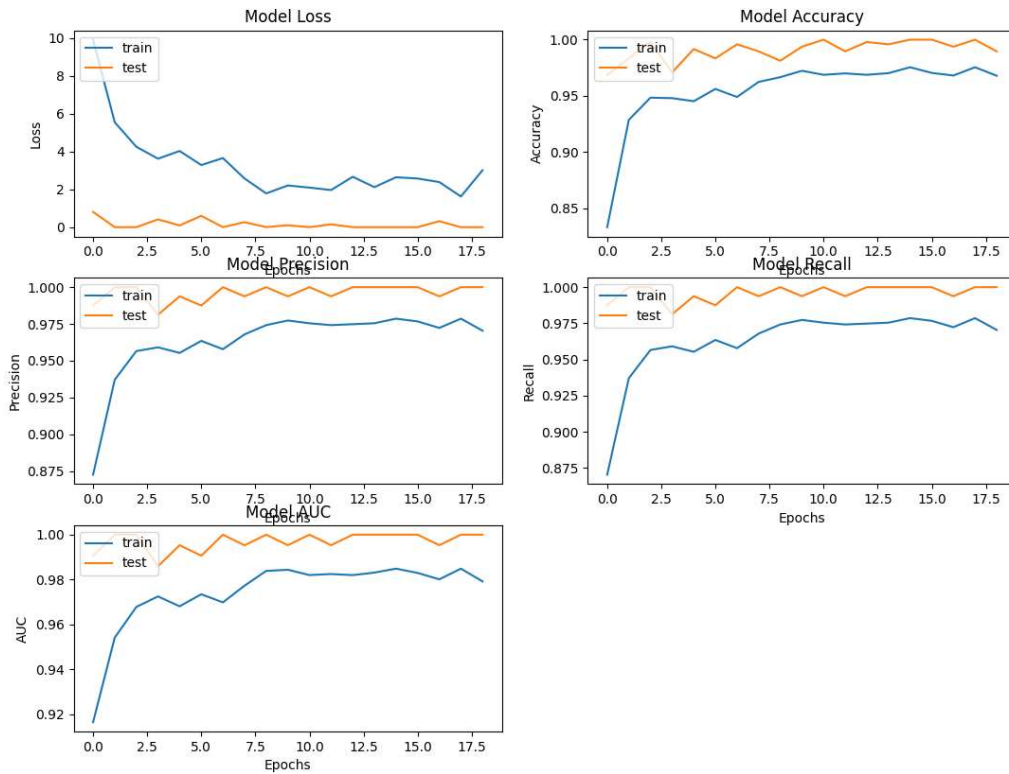


Рисунок 4.8 – Навчання мережі з архітектурою MobileNetV2

Як видно з графіків високі значення (вище 0.9) метрик було отримано вже після третьої епохи, що свідчить про високу навчальну здатність у такої архітектури нейромережі.

4.5 Порівняння результатів навчання та тестування

Після навчання трьох мереж, вони були оцінені із застосуванням повного датасету (2079) для більшої точності обчислень.

Завдяки чому були отримані значення п'яти метрик, для зручності порівняння наведені у вигляді таблиці (табл. 4.1)

Таблиця 4.1 – значення метрик оцінки якості мереж

Архітектура	VGG16	Xception	MobileNetV2
Втрати (loss)	0.18894	0.05746	1.60225
% вірних відповідей (accuracy)	0.97499	0.97371	0.98188
Точність (precision)	0.97546	0.99711	0.98413
Повнота (recall)	0.97499	0.99711	0.98413
AUC	0.990603	0.99819	0.988801
Час на обробку	1046s	273s	58s

З даних, наведених у таблиці можна зробити висновок щодо якості розглянутих у роботі архітектур. Очевидно, що найгіршою з розглянутих мереж, є архітектура VGG16, через великий час обробки поданих зображень. З точки зору часу обробки кращою є MobileNetV2, причому вона має і найбільшу точність відносно інших мереж. Xception виграє з точки зору повноти розглянутих класів, тобто вона класифікує зображення більш рівномірно по класам.

Такими чином, для низки задач, що потребуються більшої точності саме в розпізнавання окремих класів краще обрати Xception. В інших випадках, особливо, коли критичним є час обробки пропонується обирати MobileNetV2.

ВИСНОВКИ

В ході даної роботи досліджено методи вирішення задач комп'ютерного зору.

В першому розділі, присвяченому аналізу існуючих методів вирішення задач комп'ютерного зору розглянуто спектр задач, що відносяться до цього теоретичного аспекту. Було розглянуто метод Віоли-Джонса, самоорганізовані мережі Кохонена та глибинні нейронні мережі як методи розв'язання таких задач.

У другому розділі розглянуто основні аспекти використання згорткових нейронних мереж для вирішення задач класифікації зображень. Розглянуто основні особливості ЗНМ, особливості їх навчання та методи оцінки якості класифікаторів на основі нейромереж.

Третій розділ присвячено сучасним переднавченим нейронним мережам. Виділено три архітектури побудови переднавчених нейронних мереж та викладено особливості методів перенесення знань для задач класифікації.

В четвертому розділі виконано імітаційне моделювання розглянутих раніше переднавчених нейронних мереж. Виконано передобробку набору зображень, створено моделі нейромереж та виконано перенесення навчання з урахуванням розглянутої задачі визначення наявності маски на обличчі. Зроблені висновки щодо ефективності кожної з моделей.

Отримані в результаті роботи системи можуть бути використані для ідентифікації випадків порушення маскового режиму та, наприклад, недопущення таких осіб до оздоровчих закладів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Тормозов В. С. Анализ методов распознавания образов и машинного обучения для распознавания визуальных образов. Информационные технологии XXI века: Сборник научных трудов / ред. В. В. Воронин. Хабаровск: Тихоокеанский государственный университет, 2019. С. 354-359.

2 Мерков А. Б. Распознавание образов. Введение в методы статистического обучения. М.: Едиториал УРСС. 2011. 256с.

3 Воронцов К. В. Комбинаторные оценки качества обучения по прецедентам. Доклады Академии наук. 2004. №. 2. С. 175-178.

4 Мокшин В. В., Сайфудинов И. Р., Кирпичников А. П., Шарнин Л. М. Распознавание образов транспортных средств на основе эвристических данных и машинного обучения. Вестник Казанского технологического университета. 2016. №5. С. 130-137

5 Шапиро Л., Стокман Дж. Компьютерное зрение. М.: БИНОМ. Лаборатория знаний. 2006. Т. 752. С. 2.

6 Манюкова Н. В. Компьютерное зрение как средство извлечения информации из видеоряда *Математические структуры и моделирование*. 2015. №. 4 (36).

7 Полковникова Н. А. Исследование методов и алгоритмов компьютерного зрения на основе свёрточных и рекуррентных нейронных сетей *Эксплуатация морского транспорта*. 2020. №. 3 (96). С. 154.

8 Чанг Буй Тхи Тху, Хоанг Фан Нгок, Спицын В.Г. Распознавание лиц на основе применения метода Виолы-Джонса, вейвлет-преобразования и метода главных компонент *Известия Томского политехнического университета*. 2012. Т. 320. № 5. С. 54-59.

9 Шешкус А. В. и др. Исследование композиции алгоритмов компьютерного зрения, базирующихся на машинном обучении, для решения задачи локализации и классификации объектов. *Труды*

Института системного анализа Российской академии наук. 2019. Т. 69. №. 1. С. 29-36.

10 Греков И. С., Заворотна М. Г. Класфікація нейронних мереж : дис. – Кременчуцький льотний коледж, 2019.

11 Анисимова Э. С. Самоорганизующиеся карты Кохонена в задачах кластеризации. *Актуальные проблемы гуманитарных и естественных наук.* 2014. №. 9.

12 Маринич П. С., Борискевич А. А. Выбор архитектуры нейронной сети для задачи распознавания физической активности ББК 32.811. 4 К57 Редакционная коллегия: ВК Конопелько (руководитель конференции), ВЮ Цветков, ЛА Шичко. – 2021. – С. 9.

13 Recommendation I. G. 983.1: 'Broadband Optical Access Systems Based on Passive Optical Networks (PON) ITU: Geneva, Switzerland. – 1998.

14 Recommendation I. G. 989.2: 40-Gigabit-Capable Passive Optical Networks (NG-PON2): Physical Media Dependent (PMD) Layer Specification ITU: Geneva, Switzerland. – 2019.

15 Recommendation I. G. 989.3: 40-Gigabit-Capable Passive Optical Networks 2 (NG-PON2): Transmission Convergence (TC) Layer Specification Under Study. Geneva, Switzerland – 2015.

16 Аксютин Е. М., Белов Ю. С. Обзор архитектур и методов машинного обучения для анализа больших данных *Электронный журнал: наука, техника и образование.* 2016. №. 1. С. 134-141.

17 Гришанов К. М., Белов Ю. С. Модель сверточной нейронной сети в задачах машинного зрения. *Электронный журнал: наука, техника и образование.* 2017. №. СВ1. С. 100-106.

18 Weiss G. M., Provost F. The effect of class distribution on classifier learning. 2001.

19 Гайер А. В., Шешкус А. В., Чернышова Ю. С. Аугментация обучающей выборки «на лету» для обучения нейронных сетей Труды

Института системного анализа Российской академии наук. 2018. Т. 68. №. S1. С. 150-157.

20 Image data preprocessing URL: <https://keras.io/api/applications> (дата звернения: 29.11.2021).

21 Горячкин Б. С., Чечнев А. А. Анализ чувствительности метрик бинарной классификации к дисбалансу данных *E-Scio*. – 2021. – №. 4 (55). – С. 23-34.

22 Макиенко Д. О., Селезнев И. А., Сафонов И. В. Влияние дисбаланса классов в обучающей выборке на качество классификации для задачи определения литотипов по фотографиям полноразмерного керна *Информационные технологии и нанотехнологии (ИТНТ-2020)*. 2020. С. 574-581.

23 Ерохин Д. Ю., Ершов М. Д. Современные сверточные нейронные сети для обнаружения и распознавания объектов // *Цифровая обработка сигналов*. 2018. №. 3. С. 64.

24 Кузнецов А. С., Семенов Е. Ю., Матросова Л. Д. Кластеризация изображений при использовании предобученных нейронных сетей *International Journal of Open Information Technologies*. 2019. Т. 7. №. 7.

25 Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012.

26 Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition *arXiv preprint arXiv:1409.1556*. 2014.

27 He K. et al. Deep residual learning for image recognition/ *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. С. 770-778.

28 Huang G. et al. Densely connected convolutional networks *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. С. 4700-4708.

29 Харлов В. В. Применение нейронных сетей для распознавания туберкулёза по флюорографии *Фундаментальные и прикладные научные исследования: актуальные вопросы, достижения и инновации*. 2019. С. 13-19.

30 Keras Documentation. URL: <https://keras.io/applications/> (дата звернения: 29.11.2021).

31 Tsang S. H. Xception With Depthwise Separable Convolution Better Than Inception-v3 (Image Classification). *Towards Data Science*. – 2018.

32 Cleetus L., Sukumar R., Hemalatha N. Computational Prediction of Disease Detection and Insect Identification using Xception model *bioRxiv*. – 2021.

33 Забашта А. Ю., Мильченко М. А., Гришина Е. А. Обнаружение объектов в видеопотоке и построение их траектории, алгоритм центроидного отслеживания. *Ростовский научный журнал*. 2019. №. 3. С. 238-248.

34 Transfer Learning: как быстро обучить нейросеть на своих данных URL: <https://habr.com/ru/company/binarydistrict/blog/428255/> (дата звернения: 29.11.2021).

35 Богатырева А. А., Виноградова А. Р., Тихомирова С. А. Исследование способности к transfer learning сверточных нейронных сетей, обученных на ImageNet. *Международный журнал прикладных и фундаментальных исследований*. 2019. №. 7. С. 106-111.

36 Torch Contributors, PyTorch Documentation 2017. URL: <http://pytorch.org/docs/0.3.0/index.html> (дата звернения: 29.11.2021).

37 Trivedi A. Deep learning part 2: transfer learning and fine-tuning deep convolutional neural networks. 2016. URL: <https://blog.revolutionanalytics.com/2016/08/deep-learning-part-2.html> (дата звернения: 29.11.2021).

38 Конарев Д. И., Гуламов А. А. Повышение точности предварительно обученных нейронных сетей путём тонкой настройки

Материалы конференции Информационные технологии в управлении. 2020.
С. 209.

39 Face mask detector(mask, not mask, incorrect mask) URL:
<https://www.kaggle.com/spandanpatnaik09/face-mask-detectormask-not-mask-incorrect-mask> (дата звернения: 29.11.2021).