

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
Кафедра авіаційних комп'ютерно-інтегрованих комплексів

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
Доктор технічних наук
професор

Синеглазов В. М.

“ _____ ” _____ 2020 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСНИКА ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ

“МАГІСТР”

Тема: Інтелектуальна медична система на основі нейронних мереж
RESNET

Виконавець: _____ Хоцянівський В.П.

Керівник: _____ Синеглазов В. М.

Консультанти з окремих розділів пояснювальної записки:

Охорона праці: _____ Козлітін О. О.

Охорона навколишнього
середовища: _____ Маджд С. М.

Нормоконтролер: _____ Тупіцин М. Ф.

Київ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп'ютерно-інтегрованих комплексів

Освітній ступінь: магістр

Спеціальність 151 “Автоматизація та комп'ютерно-інтегровані технології”

Освітньо-професійна програма "Комп'ютерно-інтегровані технологічні процеси і виробництва"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Синеглазов В.М.

“ _____ ” _____ 2020 р.

ЗАВДАННЯ

на виконання дипломної роботи студента

Хоцянівського Володимира Петровича

1. Тема проекту (роботи): “ Інтелектуальна медична система на основі нейронних мереж RESNET ”

2. Термін виконання проекту (роботи): з _____ 2020 р. до _____ 2020 р.

3. Вихідні дані до проекту (роботи): інтелектуальна медична система на основі нейронних мереж RESNET; час відпрацювання системи становить не більше 120 с; технічні параметри системи - відповідно до аналогів медичних систем.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

1.Формування навчальної множини для завдання обробки зображення;
2. Огляд методів трансферного навчання; 3. Система підтримки прийняття рішень на базі нейронних мереж ResNet. 4.Розробка інтелектуальної медичної системи та аналіз її характеристик шляхом моделювання

5. Перелік обов'язкового графічного матеріалу:

1. Моделі множини даних та формування спостережуваних ознак; 2. Алгоритми генерації синтетичних навчальних прикладів; 3. Структурна схема алгоритму контрастування; 4. Результати дослідження навчання мережі на різних наборах даних 5. Програмне забезпечення інтелектуальної системи медичної діагностики на основі нейронних мереж ResNet.

6. Календарний план-графік

№	Завдання	Термін виконання	Підпис
1	Підбір літератури	03.09.2020 – 10.09.2020	
2	Технічне завдання	16.09.2020 – 30.09.2020	
3	Вступ	01.10.2020 – 15.10.2020	
4	Формування навчальної множини для завдання обробки зображення	16.10.2020 – 29.10.2020	
5	Огляд методів трансферного навчання	30.10.2020 – 21.11.2020	

6	Система підтримки прийняття рішень на базі нейронних мереж ResNet.	22.10.2020 – 01.11.2020	
7	Розробка інтелектуальної медичної системи та аналіз її характеристик шляхом моделювання	02.11.2020 – 29.11.2020	
8	Висновки	29.11.2020 – 01.12.2020	

7. Консультанти зі спеціальних розділів

Розділ	Консультант (посада, П. І. Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці	Козлітін О. О.		
Охорона навколишнього середовища	Маджд С. М.		

8. Дата видачі завдання

01.09.2020р.

Керівник: зав.каф.АКІК _____ Синєглазов В.М.
(підпис)

Завдання прийняв до виконання _____ Хоцянівський В.П.
(підпис)

АНОТАЦІЯ

на дипломну роботу «Інтелектуальна медична система на основі нейронних мереж RESNET»

МЕДИЦИНА, ІНТЕЛЕКТУАЛЬНА СИСТЕМА, НАВЧАЛЬНА ВИБІРКА, ТРАНСФЕРНЕ НАВЧАННЯ, RESNET, НЕЙРОННІ МЕРЕЖІ

Пояснювальна записка: кількість сторінок – , кількість рисунків – , кількість використаних джерел – .

У роботі проведено аналіз проблематик формування навчальної вибірки, розглянута класифікація та стадії пухлин шуканої хвороби. Розглянуто проблему побудови медичних інформаційних систем для підтримки прийняття рішень, а також інформаційного супроводження лікувально-діагностичного процесу. Визначено шляхи підвищення рівня інформатизації процесу діагностики.

Автором роботи розроблено узагальнену структуру інтелектуальної медичної системи та проведені теоретичні та експериментальні дослідження з метою формування навчальної множини для задачі обробки зображень при навчанні згорткової нейронної мережі.

Об’єкт дослідження – процес діагностування захворювання з використанням обробки зображень.

Предмет дослідження – інтелектуальна система діагностування захворювання.

Мета дослідження – розробка програмного комплексу для поглибленої діагностики захворювань з використанням нейронних мереж RESNET.

Задачі на дослідження :

- аналіз можливих проблем при створенні програмного комплексу;
- розробка алгоритму системи;
- розробка системи підтримки прийняття рішень на основі нейронних мереж.

Актуальність теми дослідження – зниження впливу суб’єктивних факторів на достовірність діагностичного рішення лікаря при медичному обстеженні за рахунок автоматизації обробки зображень.

Новизна запланованих результатів – створення програмного комплексу, що дозволить зменшити час на обробку зображень та формування діагностичних висновків. Підвищення достовірності рішення.

Практична цінність - зниження впливу суб'єктивних факторів на достовірність діагностичного рішення лікаря при медичному обстеженні

Методи дослідження:

Емпіричні : аналіз інформації, анкетування, спостереження досліджуваного явища.

Теоретичні: синтез нейронних мереж, аналіз системи, розрахунок коефіцієнтів та параметрів системи.

ЗМІСТ

Вступ.....	9
ФОРМУВАННЯ НАВЧАЛЬНОЇ МНОЖИНИ ДЛЯ ЗАВДАННЯ ОБРОБКИ ЗОБРАЖЕННЯ	12
1.1 Проблеми, що виникають, при побудові алгоритмів машинного навчання.....	12
1.2 Моделі множини даних	14
1.3 Модель формування спостережуваних ознак.....	15
1.4. Критерії формування навчальної вибірки.....	17
1.5. Постановка задачі.....	21
1.6. Способи генерації навчальної множини.....	22
1.7. Можливі проблеми при формуванні навчальної множини.....	25
1.8. Способи додавання даних в навчальну множину.....	27
1.9. Алгоритми генерації синтетичних навчальних прикладів	31
РОЗДІЛ 2. ОГЛЯД МЕТОДІВ ТРАНСФЕРНОГО НАВЧАННЯ	48
2.1. Постановка задачі обробки зображень.....	48
2.2. Огляд трансферного навчання	50
2.3. Індуктивна передача навчання.....	58
2.4. Навчання трансуктивного трансферу.....	65
2.5. Неконтрольованна передача навчання.....	68
2.6. Обмеження трансферу і негативний трансфер.....	70
2.7. Застосування трансферного навчання.....	71
СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ RESNET.....	78
Огляд існуючих архітектурних рішень згорткових нейронних мереж для створення інтелектуальної медичної системи.....	78
Формування навчальної множини для задачі обробки зображень при навчанні згорткової нейронної мережі	87

Донавчання нейронної мережі з використання можливостей трансферного навчання.....	99
РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ МЕДИЧНОЇ СИСТЕМИ ТА АНАЛІЗ ЇЇ ХАРАКТЕРИСТИК ШЛЯХОМ МОДЕЛЮВАННЯ	102
4.1 Опис програмної складової інтелектуальної системи медичної діагностики.....	102
4.2 Лістинг програми.....	104
4.3 Опис роботи програми.....	116
ОХОРОНА НАВКОЛИШНЬОГО ПРИРОДНОГО СЕРЕДОВИЩА....	121
5.1 Вплив об'єкту дослідження на навколишнє середовище	121
5.2 Розрахунок впливу об'єкту дослідження на навколишнє середовище.....	124
5.3 Розрахунок утворення відходів на робочому місці оператора системи	127
5.4 Шляхи покращення ситуації, рекомендації чи заходи по зменшенню впливу.....	129
ОХОРОНА ПРАЦІ.....	136
6.1 Аналіз умов праці на робочому місці	136
6.2 Розробка заходів з охорони праці	138
6.3 Розрахунок захисного заземлення комп'ютера.....	141
6.4 Пожежна безпека робочого приміщення.....	143
6.5 Інструкція з охорони праці під час роботи з інтелектуальною медичною системою на основі нейронних мереж RESNET.....	144
Висновки.....	147
Список використаної літератури.....	148

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

СППР - система підтримки прийняття рішень

НМ – нейронна мережа

ЗНР - згорткова нейронна мережа

УЗД - ультразвукове дослідження

ІС - інтелектуальна система

ШНМ – штучна нейронна мережа

ІНС – інтелектуальна нейронна система

ПЗ – програмне забезпечення

МДВ – міра довіри вибірки

МДП – міра довіри до простору ознак

МДР – міра довіри до репризинтативності вибірки

МДО – міра довіри до обсягу вибірки

МДЕВ – міра довіри експертів до вибірки

МДІ – міра довіри інформаційності

МДЕП – міра довіри експертів до складу ознак

МДК – міра довіри до кількості

МДД – міра довіри до даних

ІРД – алгоритми штучного розмноження даних

GAN –Generative Adversarial Networks

DGAN – Deep Convolutional Generative Adversarial Networks

SVM - Support Vector Machine

GP – Гаусові процеси

MT-IVM - Multi-task Informative Vector Machine

MLN – логічні мережі Маркова

NLP - Natural Language Processing

TDA - алгоритм переданого дискримінаційного аналізу

САПР – система автоматизованого проектування

ВСТУП

З технічним розвитком засобів автоматизації загострюється потреба у створенні сучасного медично - діагностичного обладнання, яке для лікаря є необхідним вимірально-інформаційним інструментом отримання інформації про захворювання обстежуваного пацієнта з метою підтримки постановки діагнозу.

Використовувані в цій сфері стандартні системи надають лікарю тільки первинну інформацію у вигляді фрагментів даних про досліджуваний орган, що є основою постановки діагностичного висновку. Його формування здійснюється лікарем суб'єктивно (шляхом різних методів когнітивної психології: сприйняття, уявлення, пізнання, розуміння, пояснення, формування рішення і т.п.) в рамках рекомендованих діагностичних мінімумів виявлення того чи іншого захворювання. Спільно з цим, лікар постійно виконує операції управління при жорстких обмеженнях часу на обстеження, це зв'язано з його специфіками, що створює додатковий дискомфорт пацієнту. Все це, а також наявність таких суб'єктивних факторів, як обсяг, інтенсивність, стійкість уваги лікаря, знижують якість проведення обстеження, збільшують його терміни, підвищують ймовірність формування помилкового висновку.

Одним з напрямків підвищення ефективності діагностичного обстеження є включення до складу інструментальних засобів окремої комп'ютерної системи підтримки прийняття рішень (СППР), що дозволяє реалізувати частину операцій з серії психологічних перетворень, які виконуються лікарем. До них відносяться: виявлення діагностичних ознак захворювання і історії хвороби, облік даних попередніх обстежень, формування висновків за сукупністю знайдених патологій і ознак захворювань у вигляді діагнозу і ін. Автоматизація виконання цих операцій дозволить істотно знизити вплив на якість діагностичних висновків таких людських (суб'єктивних) факторів, як зменшення обсягу уваги, інтенсивності, стійкості тощо, так як ряд операцій з підтримки діагностичних висновків будуть передані машині.

Проведений аналіз відомих методик і апаратури показав наступні особливості - як фактори, що впливають на достовірність діагностичного висновку.

По-перше, це відсутність виведення на екран додаткової інформації про основні виявленні ознаки захворювань; обмеженість процедури фільтрації зображень, зміни яскравості і контрастності, кольору, що ускладнює виявлення візуальних ознак на зображеннях; неможливість запису на цифровий носій інформації в зручній для подальшої обробки формі; відсутність підтримки розпізнавання контурів та інших об'єктів на зображеннях.

По-друге, наявність тільки якісних ознак внутрішніх органів у вигляді їх зображень, відсутність можливості виявлення інтегральних інформативних ознак (кожна з яких відповідає сукупності ознак, що відносяться до однієї патології), відображених в кількісній або якісній шкалах оцінки. Як правило, їх формування здійснюється лікарем на підставі його професійного досвіду.

Поряд з цим відомо, що для підвищення достовірності діагностичного висновку необхідно спільно з візуально виявленими ознаками, також враховувати вплив великого числа патогенетичних чинників. Менш досвідчений лікар може пропустити деякі важливі деталі через їх велику кількість, що призведе до постановки помилкового діагнозу. При цьому навіть найкращі системи не дозволяють об'єктивно враховувати дані історії хвороби пацієнта, а також впливу патогенетичних чинників на розвиток захворювання. Таким чином, відсутність в цих системах функціональних підсистем з обліку таких даних і чинників при формуванні діагнозу, а також підсистем з підтримки прийняття рішень (СППР) є їх істотним недоліком.

Актуальність дослідження у зв'язку з цим обумовлена необхідністю зниження ймовірності помилки постановки діагнозу на основі рішення задач розробки і аналізу моделей, алгоритмів та засобів, використовуваних для комп'ютерної підтримки встановлення діагнозу захворювань щитовидної залози.

Виходячи з цього, пропонується, використовуючи досвід і знання в різних суміжних з медициною областях, розширити функціональні можливості стандартних систем на основі створення алгоритмічного забезпечення (у вигляді спеціальних моделей і алгоритмів) на базі комплексу програм для підтримки прийняття рішень з використанням персонального комп'ютера для формування лікарем правильних діагностичних висновків.

Одним з перспективних підходів до розробки цього забезпечення є застосування нейронної мережі, для класифікації, як основи для створення функціональної медичної системи.

РОЗДІЛ 1.ФОРМУВАННЯ НАВЧАЛЬНОЇ МНОЖИНИ ДЛЯ ЗАВДАННЯ ОБРОБКИ ЗОБРАЖЕННЯ

1.1. Проблеми, що виникають, при побудові алгоритмів машинного навчання.

Однією з проблем, що виникають, при побудові алгоритмів машинного навчання і, зокрема, алгоритмів класифікації (розпізнавання) образів є формування вихідних навчальних даних. При вирішенні складних практичних завдань машинного навчання етап підготовки і організації навчальних даних стає найважливішим, якщо не сказати самим важливим з огляду на загальну тенденцію до використання алгоритмів аналізу даних, що мають універсальну структуру (нейронні мережі, композиційні алгоритми і т. д.).

У той же час пошук об'єктів, попередня обробка і формування їх формалізованих описів (ознак) з подальшою підготовкою їх до використання для навчання тих чи інших алгоритмів вимагають значних витрат часу і людських ресурсів. Особливо істотно це проявляється при обробці зображень, звуку, тексту.

У зв'язку з активним розвитком глибоких нейронних мереж в останнє десятиліття питання формування безлічі навчальних даних приймають особливо важливе значення, оскільки в багатьох задачах глибокі нейронні мережі демонструють якість, істотно перевершує інші алгоритми машинного навчання, однак, щоб отримати подібний вииграш в якості, необхідно використовувати навчальне безліч дуже великого розміру (до декількох мільйонів зображень, при цьому навчання вимагає великого обсягу обчислювальних ресурсів.

В міру ускладнення вирішуваних завдань в галузі аналізу даних, створення систем штучного інтелекту потреби в подібних діях будуть рости.

Слід також зазначити, що в ряді завдань отримання представницьких

<i>КАФЕДРА АКІК</i>				<i>НАУ 20 19 32 000 ПЗ</i>			
<i>Розроб.</i>	<i>Хоцянівський В.П.</i>			<i>Інтелектуальна медична система на основі нейронних мереж RESNET</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Керівник</i>	<i>Синеглазов В.М.</i>						
<i>Консульт.</i>							
<i>Н.контр.</i>	<i>Тупіцин М. Ф.</i>						
<i>Зав.кафедри</i>	<i>Синеглазов В.М.</i>						
					<i>151 204М</i>		

навчальних вибірок утруднено і з об'єктивних причин. Зазвичай це пов'язано з необхідністю постановки і проведення експериментальних досліджень. Слід також зазначити, що в ряді завдань отримання представницьких навчальних вибірок утруднено і з об'єктивних причин. Зазвичай це пов'язано з необхідністю постановки і проведення експериментальних досліджень. У медицині часто об'єктивно присутні обмеження, пов'язані з необхідністю отримання навчальних даних для діагностики захворювань; при цьому підготовка таких даних може займати місяці і роки.

На жаль, питань формування навчальної множини приділяється недостатня увага, часто ці питання повністю ігноруються, недостатньо розвинена теоретична база, що пояснює явища, що виникають в процесі формування безлічі навчальних даних.

Визначимо такі поняття як навчальна множина і метод навчання.

Навчальна множина. Нехай ϵ безліч об'єктів X , безліч допустимих відповідей Y , і існує цільова функція $y^*: X \rightarrow Y$, значення якої $y_i = y^*(x_i)$ відомі тільки для кінцевої підмножини об'єктів $\{x_1, \dots, x_L\} \subset X$. сукупність пар $X^L = (x_i, y_i)_{i=1}^L$ називається навчальною множиною. Завдання навчання полягає в тому, щоб по навчальній множині X^L відновити залежність y^* , тобто побудувати вирішальну функцію $a: X \rightarrow Y$, яка наближала б цільову функцію $y^*(x)$, причому не тільки на об'єктах навчальної множини, але і на всій множині X [1].

Метод навчання – це відображення $\mu: (X \times Y)^L \rightarrow A$, яке безпідставне кінцевій навчальній множині $X^L = (x_i, y_i)_{i=1}^L$ і ставить у відповідність деяку вирішальну функцію $a: X \rightarrow Y$.

Також кажуть, що метод навчання будує вирішальну функцію a по навчальному множині X^L [1].

1.2. Моделі множини даних

В якості моделі множини даних може використовуватися імовірнісна модель і кейсова модель даних, модель даних заснована на застосуванні підходу перенесення навчання (Transfer Learning). Transfer Learning дозволяє використовувати накопичений при вирішенні однієї задачі досвід для вирішення іншої, аналогічної проблеми. Нейромережа спочатку навчається на великому обсязі даних, потім - на цільовому наборі.

Сучасне машинне навчання базується на ймовірнісній моделі даних.

Вважається, що навчальна множина $X^L = (x_i, y_i)_{i=1}^L$ є вибіркою з генеральної сукупності деяких об'єктів, при цьому вибірка повинна відображати основні властивості генеральної сукупності. Також вважається, що ймовірність появи об'єктів певного типу в навчальній вибірці дорівнює ймовірності появи даних об'єктів в генеральній сукупності.

Імовірнісна модель даних має ряд недоліків.

1. Результати в ймовірнісній моделі сильно залежать від співвідношення числа об'єктів різних типів у вибірці.

2. Вірогідну модель даних зручно використовувати, тільки якщо дані представляють собою однорідну сукупність, що невірно в більшості завдань машинного навчання.

У кейсові моделі даних весь простір можливих об'єктів X розбивається на деякі види даних (кейси). Кожен кейс характеризується функцією приналежності $f(x)$ (для кожного об'єкта визначає ступінь приналежності даного кейсу) і важливістю w . При цьому важливість кейса може змінюватися в процесі розробки системи в залежності від вимог замовника. Спрощено, кейс - це набір умов, яким задовольняє деяка різновид об'єктів з безлічі даних. Допускається довільна вкладеність кейсів. Всередині одного кейса можна використовувати стандартну вірогідну модель даних. Слід зауважити, що вибір кейсів в просторі об'єктів може виконуватися довільно і служить

виключно для зручності опису роботи системи. Тут доречна аналогія з стратифікацією вибірки в статистиці.

При використанні кейсові моделі при тестуванні алгоритмів розпізнавання стають більш зрозумілі загальні властивості системи (в яких випадках вона працює правильно чи помиляється), кейсова модель дозволяє добитися стійкості по відношенню до змін відносного числа об'єктів різних видів в множині. Також спрощується опис і розуміння явищ при навчанні алгоритму, так як в різних областях простору об'єктів навчальну множину може мати абсолютно різні властивості (наприклад, різну щільність даних), а імовірнісна модель дозволяє оцінити тільки властивості навчальної множини або системи в цілому.

1.3. Модель формування спостережуваних ознак

Навчальний об'єкт x_i визначається набором спостережуваних ознак f_1, \dots, f_q . Будемо вважати, що кожен з цих ознак обчислюється деяким невідомим нам алгоритмом A_i розрахунку ознаки (індивідуальним для кожної ознаки) на основі значень деяких змінних z_1, \dots, z_L (будемо називати їх генеруючі змінні). Ці змінні можна розділити на кілька видів.

Цільові змінні. Змінні, що визначають внутрішню структуру об'єкта і його належність до того чи іншого класу D_f . Знання значень цільових змінних гарантує вирішення задачі розпізнавання.

Змінні внутрішньо класових відмінностей. Змінні, що визначають зовнішні відмінності однакових з точки зору завдання об'єктів (зростання і колір одягу пішохода в задачі пошуку пішоходів). Знання значень даних змінних не забезпечує рішення задачі розпізнавання, однак може містити цінну інформацію для побудови моделі об'єкта. Не міняються з плином часу для одного і того ж об'єкта. Зазвичай, кількість можливих значення змінних даного виду підкоряються деяким обмеженням, які можуть бути враховані корисним чином. Зауважимо, що в залежності від завдання одна і та ж змінна може бути як цільовою змінною, так і змінною внутрікласовою відмінності.

Наприклад, пропорції обличчя людини будуть цільовими змінними в задачі визначення положення ключових точок особи, і змінними внутріклассової відмінності в задачі пошуку осіб на зображенні.

Зовнішні змінні. Змінні, які не залежать від самого об'єкта, але впливають на значення спостережуваних ознак (поворот камери, масштаб об'єкта, колір фону). Можуть довільно змінюватися з плином часу для одного і того ж об'єкта незалежно один від одного без особливих обмежень. Знання значень даних змінних не дає корисної інформації про сам об'єкт. Зазвичай в систему розпізнавання прагнуть закласти інваріантність до змін таких змінних (наприклад, дескриптори зображень намагаються зробити інваріантними до повороту і перенесення об'єкта інтересу, зміни масштабу і афінних перетворень).

Випадкові змінні. Змінні, значення яких не залежать від природи об'єкта і волі людини. Зазвичай застосовуються для опису спотворень і шумів на зображенні. На основі їх значень може бути задано деякий розподіл.

Будемо називати об'єднання множин цільових змінних і змінних внутріклассової відмінності внутрішніми змінними. На рис. 1.1 представлена модель формування спостережуваних ознак.

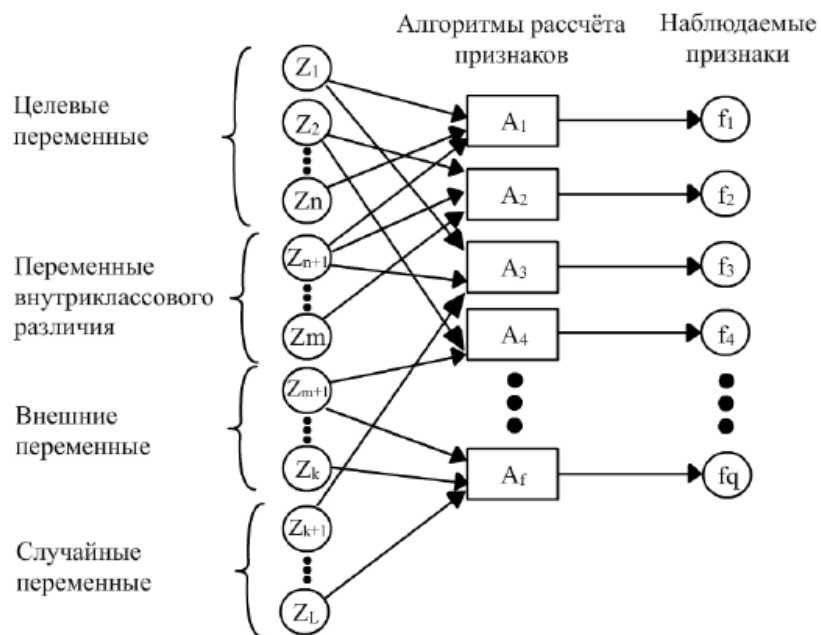


Рис. 1.1. Модель формування спостережуваних ознак

1.4. Критерії формування навчальної вибірки

На підставі аналізу термінів і різних процедур навчання процесам класифікації (розпізнавання образів) можна зробити висновок, що на майбутнє якість прийняття рішень впливають, з одного боку, якісний і кількісний склад навчальної вибірки і, з іншого боку, якісний і кількісний склад простору інформативних ознак. У свою чергу, навчальні вибірки характеризуються такими показниками, як репрезентативність (як приналежність генеральної сукупності [1, 2, 3]), обсяг і експертне довіру. Просторі ознак можна охарактеризувати статистичними показниками інформативності, експертним довірою до складу ознак і розмірністю.

Зазначені якісні та кількісні показники, що характеризують навчальні вибірки і простір ознак носять в основному емпіричний характер з явно вираженою нечіткістю визначень. Виходячи з цього, для опису вводяться показників з урахуванням ситуації, що термінології в області нечіткої логіки прийняття рішень і теорії впевнено для позначення цілісної характеристики навчальної вибірки введемо поняття міри довіри до навчальних здібностям вибірки (МДВ), а для позначення класифікаційної можливості простору ознак - міру довіри до просторі ознак (МДП). Показниками МДВ і МДП додамо властивість заходи довіри до прийнятих рішень Е. Шортліфа [4], визначивши область їх зміни в діапазоні від 0 до 1, в якому нулю відповідає повна недовіра до навчальної вибірки або складу інформативних ознак, а одиниці - повна довіра до них .

Повна довіра до навчальної вибірки і складу ознак означає, що існує потенційна можливість синтезу класифікаційних вирішальних правил, які ніколи «не помиляються».

Аналогічним чином для навчальної вибірки визначимо поняття міри довіри до репрезентативності вибірки МДР, заходи довіри до обсягу вибірки МДО, заходи довіри експертів до вибірці МДЕВ. Для простору ознак міра довіри до інформативною або інформаційної цінності МДІ, міра довіри

експертів до складу ознак МДЕП, міра довіри до розмірності (кількості) інформативних ознак МДК.

Залежно від медико-технічних можливостей завдання і розрахунок обраних показників може здійснюватися: групою висококваліфікованих експертів; за статистичними критеріями на вибірках різного, включаючи малого обсягу; з використанням змішаних стратегій (експерти, статистичні розрахунки, нечіткі конструкції і операції над ними).

З урахуванням введених визначень пропонується оцінку класифікаційних можливостей навчальних даних виробляти у відповідності з наступним методом.

1. Формуються навчальна і контрольна вибірки за алгоритмом, описаним вище, і на експертному рівні визначається склад показників $МДВ = \Phi_1$ (МДР, МДО, МДЕВ) і $МДП = \Phi_2$ (МДИ, МДЕП, МДК), де Φ_1 і Φ_2 – функціонали агрегації складових МДВ і МДП.

2. На експертному рівні визначається спосіб розрахунку кожної зі складових показників МДВ і МДП з наступного списку: експертний висновок, статистичні оцінки, змішана стратегія.

При виборі способів оцінки названих складових рекомендується дотримуватися наступних рекомендацій.

2.1. Якщо основну роботу виконують експерти (числова оцінка заходів довіри (функцій приналежностей і т.д.)), то з урахуванням складності розв'язуваної задачі відповідно до рекомендацій [4] визначається кількісний склад експертної групи та за результатами рішення текстових завдань визначається узгодженість її роботи з розрахунком коефіцієнта конкордації W . Якщо $W > 0.7$, то експертна група приступає до вирішення поставлених завдань. В іншому випадку склад групи якісно коригується.

2.2. Якщо для розрахунку заходів довіри використовуються елементи нечіткої логіки прийняття рішень модифікованої під рішення класифікаційних задач із застосуванням методів розвідувального аналізу, то використовуючи

рекомендації робіт [4], забезпечується синтез комбінованих вирішальних правил для розрахунку обраних складових з наступних загальних їх списку:

$$\{\text{МДР, МДО, МДЕВ, МДИ, МДЕП, МДК}\}. \quad (1.1)$$

2.3. Якщо в ході розвідувального аналізу вдається встановити гіперобласті перетину V_{Π} і об'єднання V_0 двох альтернативних класів, то формально показник заходи довіри до вибірці може бути оцінений виразом:

$$\text{МДВ}^* = 1 - \frac{V_{\Pi}}{V_0}, \quad (1.2)$$

де V_{Π} і V_0 – характеристики (значення функцій) перетину і об'єднання альтернативних класів ω_l и ω_r .

Показник МДВ^* може служити як для оцінки МДВ , так і використовуватися в складі додаткових показників МДР , МДО та МДЕВ з метою уточнення МДВ . Наприклад, цей показник може бути використаний експертами для уточнення своєї думки про величину МДЕВ .

2.4. Показник МДО може бути визначений при відомому обсязі навчальної вибірки шляхом використання формули, що застосовується для розрахунку обсягу цієї вибірки n_l для класу ω_l по заданій величині помилки класифікації або оцінки ймовірності правильного прийняття рішень.

Наприклад, використовуючи таблицю розрахунку обсягу навчальної вибірки n_l в залежності від складності розв'язуваної задачі S і обраній оцінці правильної класифікації P , легко вирішується зворотна задача з розрахунку $P = \text{МДО}$. Вважаючи n_l відомою величиною (число об'єктів навчальної вибірки реально формується для вирішення завдань навчання) і при заданій складності вирішального правила S визначаємо величину

$$\text{МДО} = f_0(n_l, S), \quad (1.3)$$

В роботі [4] описаний варіант розрахунку обсягу навчальної вибірки, як залежність виду:

$$n_v = f_{\Pi}(m, W_m, X, K_{\omega}, G^2), \quad (1.4)$$

де m – попереднє число можливих станів, W_m – різницю між максимальним і мінімальним значеннями спостережуваних ознак, G – величина помилки класифікації, $K\omega$ – табличне значення коефіцієнта.

Вирішуючи зворотну задачу при відомій величині n_v , отримуємо вираз для розрахунку G і через нього для $P = \text{МДО}$:

$$\text{МДО} = 1 - G = f_0(m, W_m, \bar{X}, K\omega, m, n_v). \quad (1.5)$$

У загальному вигляді при відомих, які обирають для конкретних завдань значень n_v , вирішуючи відповідні зворотні завдання, отримуємо статистичні оцінки МДО у вигляді залежностей:

$$\text{МДО} = f_0(n_v, C), \quad (1.6)$$

де C – безліч параметрів, використовуваних для розрахунку n_v .

2.5. При оцінці МДП для розрахунку показника МДІ зручно використовувати міру інформативності I по Кульбаку з розрахунковою формулою виду:

$$\text{МДІ} = \frac{I_{\max} - I}{I_{\max} - I_{\min}}, \quad (1.7)$$

де I_{\max} и I_{\min} – максимальні і мінімальні значення інформативності використовуваного простору ознак.

3. З огляду на різну природу показників, використовуваних для оцінки МДВ і МДП, а так само їх різний внесок при вирішенні різних типів завдань, доцільно для агрегації приватних показників використовувати вирази виду:

$$\text{МДВ} = \frac{\alpha_1 \text{МДВ} + \alpha_2 \text{МДО} + \alpha_3 \text{МДЭВ}}{\alpha_1 + \alpha_2 + \alpha_3}, \quad (1.8)$$

$$\text{МДП} = \frac{\beta_1 \text{МДИ} + \beta_2 \text{МДЭП} + \beta_3 \text{МДК}}{\beta_1 + \beta_2 + \beta_3}, \quad (1.9)$$

де α_i , β_j – вагові коефіцієнти визначають внесок приватних показників в розрахунок МДВ і МДП відповідно.

4. З огляду на, що складові МДВ і МДП доповнюють один одного в оцінці класифікаційних можливостей використовуваних медичних даних,

загальну міру класифікаційного довіри до даних МДД будемо визначати виразом:

$$\text{МДД} = \text{МДВ} + \text{МДП} - \text{МДП} \cdot \text{МДВ}. \quad (1.10)$$

Міра довіри МДД може бути використана як для навчальних (МДД₀), так і для контрольних (МДД_k) вибірок.

Отримані значення заходів МДД дозволяють уточнювати ступінь довіри до синтезуються вирішальним правилами, тому що враховують не тільки роботу самих класифікаційних правил, а й особливості тих даних, які залучаються для процесів навчання і контролю роботи автоматизованої системи класифікації.

1.5. Постановка задачі

Задана сукупність пар $X^L = (x_i, y_i)_{i=1}^L$, які отримані в результаті проведених досліджень. Необхідно на підставі критерію МДД (міра довіри до даних) оцінити якість вибірки і в разі необхідності, що визначається значенням помилки узагальнення нейронної мережі навченої згідно даної вибірки, вибрати метод додавання даних в навчальну множину і використовувати його для розширення навчальної вибірки.

До даної проблеми має також пряме відношення проблема незбалансованості навчальної вибірки в задачах навчання алгоритмів класифікації. Ця проблема може виникнути в ситуаціях нерівномірного представництва різних класів, коли деякі класи образів представлені істотно меншими за обсягом даними, ніж інші. Звідси виникає потреба в розробці ефективних методів і алгоритмів, що забезпечують підвищення якості машинного навчання і, перш за все, класифікації образів в умовах недостатнього обсягу навчальних даних.

Одним з перспективних напрямків для дослідження є алгоритми штучного розмноження даних (ІРД) при формуванні навчальної вибірки. У ряді джерел використовуються схожі, по суті, терміни: штучне розширення,

аугментація (від англ. *Augmentation* - примноження, додавання), морфінг - розмноження даних [5-7].

В даний час алгоритми розмноження даних застосовуються переважно в задачах обробки зображень. Проте, представляється можливим створення алгоритмів, які були б застосовні і до числових даних на основі аналізу статистичних властивостей цих даних по вихідній навчальній вибірці обмеженого обсягу. Важливо тут домовитися про те, що використання технологій ІРД не може повністю замінити процес отримання представницьких реальних навчальних даних, але дозволяє, при цьому, скоротити їх обсяг, а також забезпечити попередній аналіз якості застосовуваних алгоритмів аналізу даних в умовах, коли збір реальних даних важко або взагалі вимагає значного часу.

В роботі [8] розроблено та досліджено метод розмноження даних в навчальній вибірці на основі відновлення багатовимірної щільності розподілу ймовірностей вектора ознак, що описує об'єкти в задачі розпізнавання, із застосуванням ядерних оцінок (оцінок Парзена-Розенблатта) і генерації нових векторів з використанням методу виключень фон Неймана. Представлений алгоритм має і самостійне значення: він може використовуватися як алгоритм генерації випадкових векторів на основі відновленої багатовимірної щільності розподілу ймовірностей за експериментальними даними.

1.6. Способи генерації навчальної множини

Програмна генерація. Генерація синтетичних навчальних даних по деякому алгоритму. В даному випадку алгоритм генерації і його параметри визначають отриманий на виході розподіл в просторі об'єктів. Доцільно в процесі генерації варіювати якомога більше параметрів. Однак якщо реально можлива лише мала частина об'єктів з простору параметрів генерації або найбільш типові об'єкти мають складну конфігурацію, метод виявляється неефективним, так як згенерує багато безглузких і мало дійсно важливих даних.

Семпсування. Покликане подолати недоліки попереднього методу. Здається деякий апріорний розподіл в просторі об'єктів, і алгоритм намагається згенерувати вибірку з даного розподілу. Застосовуються методи вибірки з відхиленням, семплірування по Гіббсу, схема Метрополіс-Гастингса, *Markov Chain Monte Carlo*. Дані методи застосовуються для того, щоб досліджувати не весь простір об'єктів, а тільки його найбільш осмислені частини (зазвичай осмисленими є лише дуже невелика частина потенційно можливих об'єктів, коли простий перебір об'єктів з перевіркою їх правильності може зайняти неоправданно багато часу).

Закономірна модифікація базового об'єкта. Є набір базових об'єктів, навчальна множина виходить шляхом безперервної модифікації їх параметрів (як правило, зовнішніх змінних). Приклади - вибірка кадрів з відеопослідовності, вибірка поз людини з *posar-a*. Даний метод застосовується в тих випадках, коли немає можливості аналітично задати розподіл можливих об'єктів в просторі об'єктів, або не підходить використання синтетичних даних.

При використанні даного методу генерації навчальної множини стійко зберігається велика кількість фонових закономірностей. Крім того, виникають проблеми, якщо отримане таким чином безліч розбити в деякому відношенні на навчальне і тестове і використовувати отримане тестове безліч для контролю за перенавчанням. Оскільки об'єкти вихідного мають сильну взаємозалежність, низький рівень помилки алгоритму на тестовій вибірці не гарантує відсутності «заучування» навчальних даних. При використанні метричних алгоритмів класифікації найвигідніша стратегія в даному випадку - завжди відносити об'єкт до того ж класу, до якого належить найближчий навчальний об'єкт. Приклад помилки подібного роду при використанні алгоритму *k-NN* наведено в роботі [9] (при налаштуванні гіперпараметров алгоритму методом ковзного контролю за багатьма даних, отриманого вищеописаним методом, завжди вибирався параметр $k = 1$). Тому при

використанні даного методу формування безлічі даних не слід застосовувати метод ковзаючого контролю в чистому вигляді.

Вибірка з бази об'єктів. Процеси формування безлічі зображень з произвольними пейзажами або безлічі зображень облич людини погано описуються наведеними вище моделями. Ми не можемо отримати об'єкт з будь-якими заданими внутрішніми параметрами, навпаки, є фіксований набір об'єктів із заздалегідь визначеними параметрами. У разі фотографій місцевості розташування цих об'єктів підпорядковується суворим обмеженням. Всі об'єкти можна розбити на групи, причому об'єкти всередині групи будуть дуже схожими, а об'єкти з різних груп - відрізнятися (можна порівняти зображення осіб злочинців і зображення облич політиків). Об'єкти, географічно розташовані близько один до одного, зазвичай більш схожі, ніж об'єкти, що знаходяться далеко один від одного (наприклад, вдома в одному і в різних містах). Якщо ми маємо справу з колекцією фотографій, слід також брати до уваги переваги оператора - люди набагато частіше знімають красиві і незвичайні об'єкти (наприклад, пам'ятки), ніж одноманітний ліс або пустелю, різні люди роблять знімки різних типів.

Проблема, що виникає при генерації навчальної множини даними способом - важко гарантувати наявність усіх принципово важливих типів об'єктів в безлічі даних. Допомогти в цьому випадку може перехід до автоматичної генерації або автоматичному збиранню даних. Також слід робити вибір об'єктів з якомога більш широкого набору груп об'єктів або місць зйомки (детектор особи, навчений на базі осіб злочинців, можливо, буде не дуже добре працювати в загальному випадку).

Приклад використання даної моделі: якщо при вирішенні завдання *hard samples mining* складним негативним прикладом виявилось вікно деякого турецького будинку, то логічно шукати схожі зображення на знімках будинків, зроблених в Туреччині (якщо є подібна інформація про зображення). Проста імовірнісна модель формування навчальної множини не дозволяє проводити подібні міркування.

1.7. Можливі проблеми при формуванні навчальної множини

Для оцінки якості навчальної множини зазвичай використовується його обсяг (кількість навчальних прикладів). Однак дана метрика не особливо інформативна. По-перше, даних може бути дуже багато, але всі вони - однакові, по-друге, навіть якщо всі об'єкти - різні, деякі області простору ознак можуть залишитися незаповненими, і, по-третє, в самій процедурі формування навчальної множини можуть бути закладені помилки. Розглянемо деякі можливі проблеми і помилки при формуванні навчальної множини.

Фонові закономірності. У завданнях машинного навчання об'єкт може бути заданий набором значень ознак f_1, \dots, f_q і значеннями цільових змінних z_1, \dots, z_L . Завдання машинного навчання - знайти закономірності між значеннями спостережуваних ознак і цільових змінних. При цьому на основі кожного конкретного навчального об'єкта, не беручи до уваги інші об'єкти, будь-яку залежність $y_k(x_i)$, характерну для даного об'єкта, можна порахувати завжди істинною. При розгляді великої кількості різноманітних об'єктів з усіх можливих закономірностей характерними залишаться лише невелика кількість дійсно значимих закономірностей. Зауважимо, що на підставі малого числа даних немає ніякого способу відрізнити правильну закономірність від помилкової. Будемо називати подібні помилкові закономірності, що виникають в результаті браку даних, фоновими закономірностями. По суті, деякі види перенавчання полягають в заучуванні фонових закономірностей. Приклад фонові закономірності - залежність між класом зображення і кольором одного конкретного пікселя.

Відсутність навчальних об'єктів певного виду. Найпростіший приклад помилки при формуванні навчальної множини - якщо в ньому відсутні дані певного виду (не покритих деяка область простору об'єктів, в кейсові моделі даних - відсутні об'єкти деякого кейсу), алгоритм не зможе правильно навчитися їх класифікувати. При цьому маються на увазі об'єкти в просторі генеруючих змінних z_1, \dots, z_L , а не в просторі ознак f_1, \dots, f_q .

Логічно було б додати сюди і недостатня кількість навчальних об'єктів певного виду, однак в різних випадках достатнім є різне число об'єктів, при різних алгоритмах навчання дана проблема буде проявлятися зовсім по-різному, тому будемо вважати, що ця проблема входить в дві наступні проблеми.

Відсутність даних певного виду щодо ознакової системи. Ознакова система f_1, \dots, f_q породжує деякий розбиття множини даних на кейси, кожному кейсу відповідає деякий вузький набір значень ознак, при цьому кейсів тим більше, чим більш різноманітні і складні ознаки. Якщо деякий з даних кейсів НЕ буде покритий об'єктами з навчальної множини або імовірнісний розподіл всередині кейса буде невірно відображати властивості генеральної сукупності, навчання може виявитися некоректним. Зауважимо, що при ускладненні признакової системи підвищуються вимоги до навчального безлічі. Приклад виникнення даної проблеми: в задачі пошуку руки людини на зображенні в якості ознак використовується гістограма орієнтованих градієнтів з розподілом простору поворотів на 64 комірки, при цьому в деякі з вічок не потрапляє жодна рука з навчальної множини, і алгоритм може вивчити, що рук з таким кутом повороту не існує. При розподілі простору поворотів на 32 комірки проблема, можливо, зникне, а при виборі іншого признакової системи (наприклад, інтегральні зображення) даний недолік навчальної множини взагалі не буде проявлятися при будь-яких параметрах признакової системи.

Деякі з генеруючих змінних не варіюються. Важливий окремий випадок проблеми відсутності даних певного виду. Дуже часто при формуванні навчальної множини частина генеруючих змінних має завжди одні й ті ж значення або дуже вузький діапазон значень.

Розбалансування. Нерозумне з семантичної точки зору порушення співвідношень кількості даних різного виду в аналізованому безлічі даних, що приводить до необґрунтованого завищення впливу на результат одних і заниження впливу або повного ігнорування інших даних, і, як наслідок, до

прийняття неоптимальних рішень. Найпростіший приклад - в навчальній множині через особливості його формування прикладів одного класу або типу набагато більше, ніж прикладів іншого класу або типу. Різні алгоритми навчання мають різну стійкість до подібних проблем. Розбалансування особливо критична при використанні дерев рішень. Зауважимо, що розбалансування - досить загальний клас явищ, які можуть виникати не тільки в процесі формування навчальної множини. Розбалансування також можлива в задачах чисельної оптимізації і при тестуванні алгоритмів розпізнавання (наприклад, коли в задачі розпізнавання пози людини штраф за відхилення стегна від справжнього стану перевищує поліпшення в точності визначення інших суглобів, взяті разом, хоча з практичної точки зору дана помилка не має вирішального значення). Часто проблема розбалансування вирішується за допомогою різних видів нормалізації.

Зовнішні закономірності. У навчальній множині $X^L = (x_i, y_i)_{i=1}^L$ можуть бути присутніми залежності між зовнішніми і цільовими змінними, які алгоритм навчання може вивчити як справжні через розбалансіровок, особливостей навчання або ознакової системи. Наприклад, в навчальній множині $X^L = (x_i, y_i)_{i=1}^L$ всі чоловічі особи сфотографовані в світлий час доби, а всі жіночі - в темне. В таких умовах коректне навчання можливо лише в тому випадку, якщо в ознаковій системі $f_b \dots f_q$ немає ознак, що залежать від освітленості зображення, як тільки такі ознаки будуть додані, алгоритм буде працювати неправильно на вибірці з іншого джерела. Наявність подібних закономірностей - приклад помилки при формуванні навчальної множини. Ознака наявності подібної проблеми - якщо після додавання більш якісних ознак або ознак на основі інформації іншого виду алгоритм перестає коректно працювати.

1.8. Способи додавання даних в навчальну множину

Додавання (штучне розмноження даних (ІРД)) даних є одним з найпростіших і ефективних способів поліпшити якість навчальної множини. У ряді джерел використовуються схожі, по суті, терміни: штучне розширення,

аугментація (від англ. Augmentation - примноження, додавання), морфінг - розмноження даних. В даний час алгоритми розмноження даних застосовуються переважно в задачах обробки зображень. Проте, представляється можливим створення алгоритмів, які були б застосовні і до числових даних на основі аналізу статистичних властивостей цих даних по вихідній навчальній вибірці обмеженого обсягу. Важливо тут домовитися про те, що використання технологій ІРД не може повністю замінити процес отримання представницьких реальних навчальних даних, але дозволяє, при цьому, скоротити їх обсяг, а також забезпечити попередній аналіз якості застосовуваних алгоритмів аналізу даних в умовах, коли збір реальних даних важко або взагалі вимагає значного часу. При цьому просте додавання даних довільного виду не завжди ефективно, часто потрібно додати дані певного різновиду для підвищення якості розпізнавання. Розглянемо деякі способи додавання даних.

Програмна генерація. У разі використання синтетичних навчальних даних найзручніше згенерувати відсутні навчальні приклади. Однак не у всіх завданнях допустимо використання програмно згенерованих даних. У таких випадках доводиться застосовувати більш складні методи додавання даних.

Data augmentation. Модифікація наявних зображень з метою розширити навчальну множину. Активно застосовується при навчанні глибоких нейронних мереж, а також в умовах дефіциту розмічених даних. Застосовуються стиснення / розтягування, горизонтальне відображення, поворот, випадковий зсув в колірному просторі, випадкове або закономірна зміна деяких пікселів. Вважається, що додавання повністю випадкового шуму неефективно, слід додавати шум, обумовлений даними (тільки потенційно можливі в реальних даних спотворення). Істотний недолік даного методу - більшість фонових закономірностей зберігається.

Hard samples mining [10]. Класична проблема в задачах пошуку об'єктів на зображенні - потреба в підтримці достатнього числа hard negative samples

(навчальних прикладів, які схожі на об'єкт інтересу, але таким не є) в навчальній множині. Складність виникає через те, що в природних умовах такі об'єкти зустрічаються рідко, тому застосовуються спеціальні методи для їх пошуку і додавання в навчальну множину (hard samples mining). Ключове припущення в даних методах - цікавлять нас об'єкти сильно схожі між собою. Зазвичай застосовуються data augmentation, адаптивний пошук, пошук по шаблонах, методи на основі машинного навчання. Цікавим є застосування методів тематичного моделювання для пошуку складних негативних прикладів.

Імітація додавання даних. При навчанні глибоких нейронних мереж обов'язковим вважається застосування методу dropout [11]: випадкове обнулення активацій деяких нейронів в мережі при подачі їй на вхід чергового тренувального зображення (зазвичай в кожному шарі випадково вибирається 20-50% нейронів). Без застосування даної техніки нейронна мережа «заучиває» велика кількість фонових закономірностей через те, що складність моделі перевищує обсяг доступних даних. По суті, dropout - це імітація додавання даних в навчальну множину. У середині алгоритму навчання ми імітуємо мінливість даних - на вхід глибших рівнів мережі надходить випадковим чином змінена версія реального зображення (мається на увазі зображення, що не викликало б активацію обнулення нейронів), хоча таких даних насправді немає в навчальній множині. Недолік даного методу - може бути імітований додавання таких даних, яких в принципі не може бути в реальності, через що може страждати точність розпізнавання. Цікавим є створення модифікацій даного методу, що враховують природу даних.

Краудсорсинг. Оскільки для навчання глибоких нейронних мереж потрібні величезні обсяги вручну розмічених навчальних даних, для формування навчальної множини активно використовуються сервіси краудсорсингу - користувачі сервісу за невелику плату виробляють розмітку «сирих» даних (наприклад, вказують, які об'єкти є на даному зображенні і де вони розташовані) . Найпопулярніший з таких сервісів - Amazon Mechanical

Turk [12]. Проблема даного методу - велика кількість помилок в розмітці, так як користувачі не завжди роблять свою роботу сумлінно, а іноді просто помиляються, тому потрібні спеціальні методи контролю помилок в розмітці. Зазвичай одну і ту ж картинку дають розмітити кільком користувачам, а потім вибирають той варіант розмітки, який вибрало найбільше число користувачів. Також вводяться спеціальні метрики «сумлінності» користувача.

Застосування генеративно-змагальної нейронної мережі (GAN)[13]

GAN є архітектурою, яка складається з генератора і дискримінатора. Архітектура даної мережі складається з двох різних мереж. Одна нейронна мережа - генератор, створює випадкові нові екземпляри даних, а інша - дискримінатор, оцінює їх на справжність. Тобто дискримінатор приймає рішення, чи стосується екземпляр даних до набору тренувальних даних чи ні. Так само існує різновид даної архітектури, званої DCGAN (Deep Convolutional Generative Adversarial Networks) - згорткові генеративно-змагальні мережі. Ця модель замінює сверточное шарами повнозв'язні шари генеративної змагальної мережі. Дану мережу можна ефективно застосувати при вже наявному наборі даних.

Генератор створює нові зображення, які він передає на оцінку дискримінатору. Мета генератора полягає в тому, щоб генерувати такі дані, які будуть прийняті дискримінатором. Мета дискримінатора - визначити, чи є зображення справжнім (рис. 1.2).

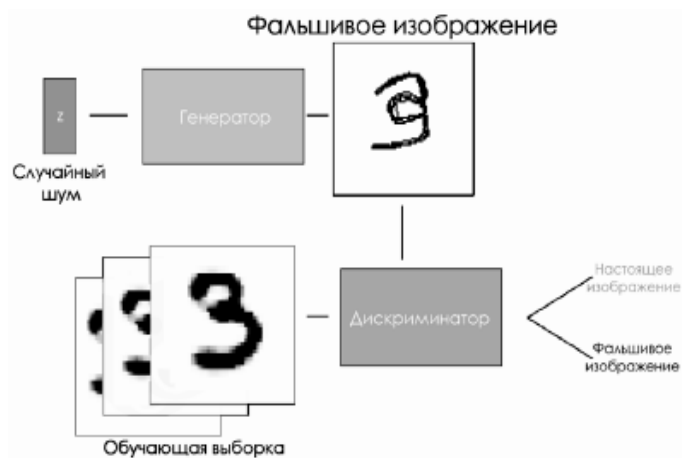


Рис. 1.2. Принцип роботи GAN мережі

При цьому генератор не має уявлення про те, що являють собою вихідні дані і навчається на основі відповідей дискримінатора, з кожної ітерацією змінюючи результати своєї роботи. За основу генератор бере вектор випадкового шуму і на підставі його генерує дані.

При використанні мережі GAN для генерації зображень існує певна складність з навчанням даної мережі. Існує ряд правил, яких варто дотримуватися, наприклад, при навчанні дискримінатора необхідно утримувати значення генератора постійними і навпаки. Тобто кожна мережа повинна тренуватися проти статичного «противника».

1.9. Алгоритми генерації синтетичних навчальних прикладів

Як відомо, у випадку використання синтетичних навчальних даних зручніше всього згенерувати відсутні навчальні приклади. Модифікація зображень з метою збільшити навчальну вибірку активно використовується при навчанні нейронних мереж, а також в умовах дефіциту розмічених даних. Використовується стягування(розтягування), поворот, випадкове зрушення в контрастування зображень, сегментація зображення та його апроксимація.

Розглянемо метод повороту зображення. Пропонований метод повороту опишемо на конкретному прикладі. Спочатку розглянемо випадок, коли зображення представляється в растровому вигляді і необхідно змінити його орієнтацію. Для цього потрібно зробити для кожної точки зображення геометричну операцію повороту щодо заданої базової точки. Тобто для кожного пікселя екрану кольору X з координатами (x, y) передбачається здійснити поворот на кут j за допомогою синусно-косинусного перетворення [100] і визначити, в який новий піксель екрану (x', y') цей піксель перемістився, і після цього присвоїти пікселю (x', y') колір X .

Недоліком цього методу є те, що при такому методі в новій матриці символу залишаються пікселі, значення яким не присвоєно.

Для вирішення цього завдання пропонується рішення від зворотного, тобто для кожної точки з нового розбиття екрану з координатами (x'', y'') треба

здійснити поворот на $-j$ і визначити, в піксель якого кольору потрапила ця точка, і привласнити точці з координатами (x'', y'') цей колір.

Синусно-косинусне перетворення повороту символу в матричній формі описується формулою

$$|X^*| = |X| \cdot |T|, \quad (1.12)$$

де $X^* = (x^*, y^*, I)$ – результуючий вектор координат точки;

$X = (x, y, I)$ – вектор вихідних координат точки;

$$|T| = \begin{vmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ \Delta X & \Delta Y & 1 \end{vmatrix} - \text{матриця перетворення};$$

Формула (1.13) може бути представлена у вигляді системи рівнянь (1.13):

$$\begin{cases} x^* = x \cos \varphi - y \sin \varphi + \Delta X; \\ y^* = x \sin \varphi + y \cos \varphi + \Delta Y. \end{cases} \quad (1.13)$$

Аналіз даної системи рівнянь вказує на те, що для зміни координат однієї точки зображення необхідно виконати чотири операції множення і чотири операції арифметичного додавання для кожної точки растра.

Альтернативний метод перетворення зображень розглядався в роботі [101]. Його основний зміст полягає в тому, що в відеопам'яті зберігаються заздалегідь підготовлені зображення, повернені з кроком 22.50 градуси. Цей метод дозволяє здійснювати повороти тільки за рахунок змін напрямків зчитувань матриці растрового зображення символу, проте вимагає додаткових ресурсів пам'яті для зберігання заздалегідь підготовлених зображень.

Інші описані в літературі [100],[101] методи часто використовують "в лоб" перетворення за формулами (1.13), яке дозволяє виконати поворот за $4S$ операцій множення і $4S$ операцій додавання, або більш просте перетворення (1.12) в полярній системі координат, що скорочує час виконання до $2S$ операцій множення і $3S$ операцій додавання, за умови, що S - кількість пікселів символу, а значення синуса і косинуса кута повороту вже попередньо обчислено. Для певних типів простих зображень, які представляють собою композицію деяких ліній (векторів), кількість арифметичних операцій можна

скоротити до $4S$ ($2S$ операцій множення і $2S$ операцій додавання) [102, 103], що для деяких додатків є цілком прийнятним.

У пропонованому методі операція повороту, заданого прямокутною матрицею $S[i, j]$, наприклад, зображення ділянки карти на екрані, здійснюється за 2 арифметичні операції додавання для кожного пікселя. Цей метод є модифікацією методу базових символів [101] і годиться як для довільного реєстрового, так і довільного векторного зображення символу.

Назвемо абсолютною системою координат координатну сітку, поодинокі клітини якої збігаються з пікселями екрану. Відносної системою координат назвемо координатну сітку, поодинокі клітини якої збігаються з пікселями матриці символу $S[i, j]$.

Виберемо точку початку відносної системи координат (x_1, y_1) . Для зручності обчислень можна взяти одну з вершин прямокутної матриці, в якій зберігається растрове зображення символу. Для цієї точки обчислюються її координати (x', y') після повороту за допомогою синусно-косинусного перетворення (1.14):

$$\begin{aligned}x' &= x_1 + (x_0 - x_1) \cdot \cos(\varphi) + (y_1 - y_0) \cdot \sin(\varphi); \\y' &= y_1 + (y_0 - y_1) \cdot \cos(\varphi) + (x_0 - x_1) \cdot \sin(\varphi).\end{aligned}\quad (1.14)$$

де φ - кут повороту;

x_0, y_0 - координати точки, щодо якої здійснюється поворот O .

Потім обчислюються проєкції кордонів одиничної клітини відносної системи координат на абсолютну:

$$\begin{aligned}d \times 1 &= h_1 \cdot \sin(\varphi); \quad dy_1 = h_1 \cdot \cos(\varphi); \\d \times 2 &= -h_2 \cdot \cos(\varphi); \quad dy_2 = -h_2 \cdot \sin(\varphi).\end{aligned}\quad (1.15)$$

де h_1, h_2 - довжина сторін пікселя.

Далі від обчислених значень x' і y' нові значення обчислюються за допомогою циклу по клітинам відносної системи координат. Так визначається новий колір пікселів при повороті.

Нижче представлена реалізація цього циклу на мові програмування C++ (див. рис 1.3).

```

void rotate(){
    int x1:=x'-dx2;
    int y1:=y'-dy2;
    for(int x=0; x<n;x++){
        j:=0;
        x1:=x1+dx2;
        y1:=y1+dy2;
        x2:=x1;
        y2:=y1;
        S[i,j]:=S[x1,y1];
        for(int j=1; j<m;j++){
            x2:=x2+dx1; y2:=y2+dy2;
            S'[x2,y2]:=S[i,j];
        }
    }
}

```

Рис 1.3. Реалізація алгоритму повороту зображення

Результати роботи алгоритму(див.рис.1.4):

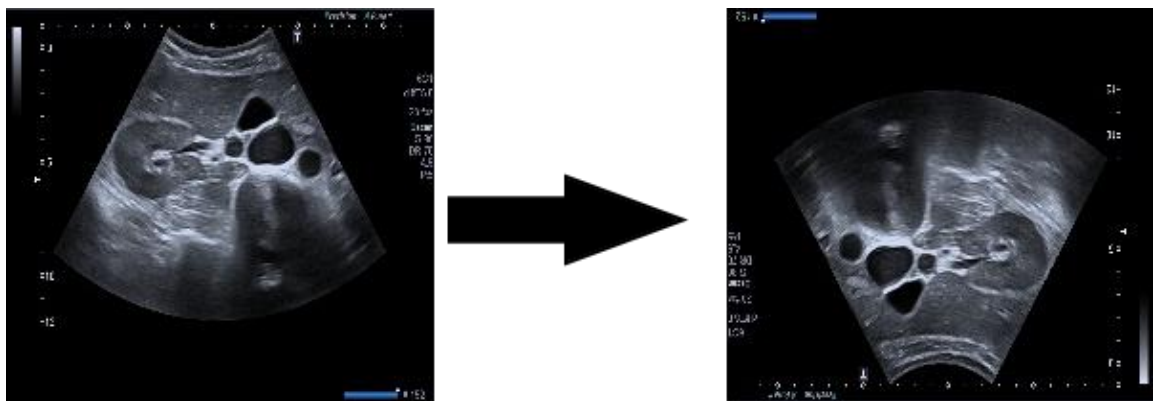


Рис 1.4. Результати роботи алгоритму повороту зображення

Запропонований метод показує, як зробити поворот зображення, представленого матрицею $n*m$ пікселів за $2n(m+1)+18$ операцій додавання, в той час як алгоритми повороту, описані в літературі [100-103], здійснюють поворот не менше, ніж за $2nm$ операцій множення і $2nm$ операцій додавання.

Отже, запропонований алгоритм дозволяє заощадити $2nm$ операцій множення. З урахуванням того, що операція множення вимагає більше витрат ресурсів, ніж операція додавання (для цілих чисел можна покласти, що витрачається в $a=4$), то економиться значна кількість часу

$$\frac{a}{a+1} * 100 = 80\% \quad (1.16)$$

Розглянемо метод контрастування деяких пікселів. Якщо з переліку графічних форматів виключити .jpg, який використовує YCbCr координати, то

переважна більшість графічних файлів зберігає саме RGB координати. Тому за операційну колірну координатну систему для розробки була прийнята RGB. Тепер будь-який з проєктованих перетворень кольору буде відбуватися не виходячи за рамки RGB координат, в яких існує початкове зображення.

Для розуміння алгоритму потрібно розглянути наступні характеристики кольоровості:

- відтінок

$$H = \begin{cases} \text{MinMin}(RGB) \\ \text{MaxMax}(RGB) \times [\text{Med}(RGB) - \text{Min}(RGB)] / \\ \quad \quad \quad / [\text{Max}(RGB) - \text{Min}(RGB)] \\ \text{MaxMax}(RGB), \end{cases} \quad (1.17)$$

де $\text{Min}(RGB)$, $\text{Med}(RGB)$, $\text{Max}(RGB)$ - найменша, проміжна і найбільша колірні координати відтінку;

$\text{MinMin}(RGB) \equiv 0$ і $\text{MaxMax}(RGB)$ - найменше та найбільше можливі значення колірних координат, що відповідають обраному колірному вирішенню: при 24bitRGB $\text{MaxMax}(RGB) = 255$, а при 48bitRGB $\text{MaxMax}(RGB) = 65535$ [104];

- насиченість

$$S = [\text{Max}(RGB) - \text{Min}(RGB)] / \text{Max}(RGB). \quad (1.18)$$

Доцільність визначення колірної насиченості була обґрунтована в [105].

Ми знаємо, що для однозначного зміни візуального контрасту будь якої характеристики кольору необхідно зберігати вихідними значення двох інших характеристик. Тобто, - для однозначної зміни візуального контрасту яркостей необхідно зберігати вихідними значення відтінків і насичення всіх контрастуючих пікселів.

Тобто, проміжна колірна координата відтінку залишається постійною

$$\begin{aligned} & \text{MaxMax}(RGB) \times [\text{Med}(RGB) - \text{Min}(RGB)] / \\ & \quad \quad \quad / [\text{Max}(RGB) - \text{Min}(RGB)] \equiv \\ & \equiv \text{MaxMax}(RGB) \times [\text{Med}'(RGB) - \text{Min}'(RGB)] / \\ & \quad \quad \quad / [\text{Max}'(RGB) - \text{Min}'(RGB)] \end{aligned} \quad (1.19)$$

лише при однаковій зміні всіх трьох координат кольору:

$$\begin{aligned} \text{Min}'(RGB) &= \text{const} \cdot \text{Min}(RGB) \\ \text{Med}'(RGB) &= \text{const} \cdot \text{Med}(RGB) \\ \text{Max}'(RGB) &= \text{const} \cdot \text{Max}(RGB). \end{aligned} \quad (1.20)$$

А це може бути справедливо тільки в разі пропорційності(1.21):

$$R' / R = G' / G = B' / B. \quad (1.21)$$

Штрихом позначені результуючі значення. Неважко показати, що і координати кольоровості при пропорційній зміні RGB координат виявляться постійними:

$$\begin{aligned} R' / (R'+G'+B') &= R / (R+G+B) \\ G' / (R'+G'+B') &= G / (R+G+B) \\ B' / (R'+G'+B') &= B / (R+G+B). \end{aligned} \quad (1.22)$$

Інакше кажучи, альтернативна форма для вираження (1.23) матиме вигляд

$$R' / G' = R / G; \quad G' / B' = G / B; \quad B' / R' = B / R. \quad (1.23)$$

Аналогічним чином, використовуючи вираз (1.19), можна довести, що збереження колірної насиченості можливо тільки при пропорційній зміні найменшою і найбільшою колірних координат. Цей важливий висновок є методичною основою для системного контрастування яскравостей.

Слід підкреслити, що принцип пропорційної зміни всіх колірних координат має важливе значення і в структурі інших алгоритмів контрастного перетворення. Так, крім яскравості контрастування, він, наприклад, затребуваний і при контрастированні колірної насиченості. Питання, однак, виходить за рамки цієї роботи.

Атрибути пропорційного контрастування - опорна точка і коефіцієнт контрастування. Перша визначається як точка, що не міняє своєї яскравості Y_0 в процесі контрастування. Другий - M - показує у скільки разів повинен змінитися контраст яскравостей між усіма парами пікселів в результаті контрастування. У сучасному ПО значенням Y_0 прийнято керувати за допомогою інструменту <Brightness>, значенням M - за допомогою інструменту <Contrast>.

Структура алгоритму формується функціональними процедурами, які деталізують їх операціями і умовними переходами. Всі елементи структурної і ряд - принципової схеми алгоритму суміщені нижче на рис. 1.5. Передбачається, що зображення вже розкладено на матриці R -, G -, B компонент.

Словесно алгоритм описується наступним чином:

1. Зчитування з командного рядка або зі спливаючого вікна значень яскравості опорної точки Y_0 і коефіцієнта контрастування M , а з файлу *rgb.txt* - яскравості коефіцієнтів K , Z і C .

2. Зчитування з трьох *R.txt*, *G.txt* і *B.txt* файлів значень кольірних координат всіх пікселів зображення.

3. Обчислення вихідної яскравості першого пікселя по формулі

$$Y = K \cdot R + Z \cdot G + C \cdot B. \quad (1.24)$$

4. Якщо $Y > Y_0$ при $M > 1$, то перехід до п. 5.

при $M < 1$, то перехід до п. 10.

при $M = 1$, то перехід до п. 15.

якщо $Y < Y_0$ при $M > 1$, то перехід до п. 10.

при $M < 1$, то перехід до п. 5.

при $M = 1$, то перехід до п. 15.

якщо $Y = Y_0$, то перехід до п. 17.

5. Оцінювання гранично можливого значення яскравості першого пікселя, досяжного в процесі яркостного контрастування при збереженні його кольоровості. Яскравість першого пікселя буде тільки збільшуватися, коли вона спочатку більше опорної при підвищенні контрасту (при $M > 1$) і коли вона спочатку менше опорної при зниженні контрасту (при $M < 1$).

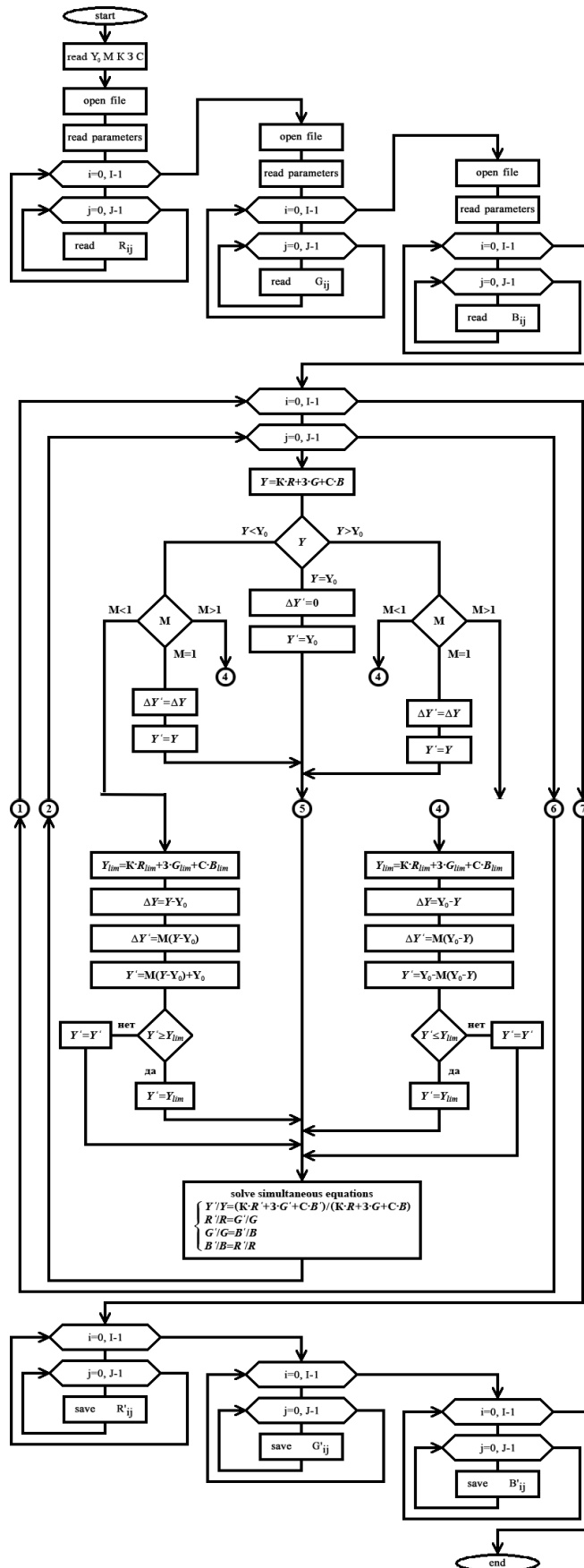


Рис.1.5. Схема алгоритму яркісного контрастування.

Зростання яскравості обмежує лише верхня межа діапазону колірних координат $\text{MaxMax}(RGB)$. Найбільше значення яскравості встановлюється при послідовному виконанні операцій, наведених нижче:

- виявити найбільшу (з трьох) колірну координату $\text{Max}(RGB)$;
- привласнити $\text{Max}(RGB)_{lim}$ значення $\text{MaxMax}(RGB)$;
- знайти значення двох, що залишилися граничних колірних координат з умови збереження кольоровості при:

$$\begin{aligned} R_{lim} = \text{MaxMax}(RGB) &\Rightarrow G_{lim} = R_{lim} \cdot G/R, B_{lim} = R_{lim} \cdot B/R, \\ G_{lim} = \text{MaxMax}(RGB) &\Rightarrow B_{lim} = G_{lim} \cdot B/G, R_{lim} = G_{lim} \cdot R/G, \\ B_{lim} = \text{MaxMax}(RGB) &\Rightarrow R_{lim} = B_{lim} \cdot R/B, G_{lim} = B_{lim} \cdot G/B; \end{aligned} \quad (1.25)$$

- обчислити граничне значення яскравості

6. Обчислення вихідної дев'ятиці яскравості першого пікселя

7. Обчислення нової дев'ятиці яскравості першого пікселя

8. Обчислення нової яскравості першого пікселя

9. Якщо $Y' \geq Y_{lim}$, то $Y' = Y_{lim}$;

якщо $Y' < Y_{lim}$, то $Y' = Y'$. Перехід до п. 19.

10. Оцінювання гранично можливого значення яскравості першого пікселя, досяжного в процесі яскравісного контрастування при збереженні його кольоровості. Яскравість першого пікселя буде тільки зменшуватися, коли вона спочатку менше опорної при підвищенні контрасту (при $M > 1$); і коли вона спочатку більше опорної при зниженні контрасту (при $M < 1$). Падіння яскравості обмежує лише нижня межа діапазону колірних координат $\text{MinMin}(RGB)$. Найменше значення яскравості встановлюється при послідовному виконанні операцій, наведених нижче:

- виявити найменшу (з трьох) колірну координату $\text{Min}(RGB)$;
- привласнити $\text{Min}(RGB)_{lim}$ значення 1;
- якщо $\text{Min}(RGB) \neq 0$, то знайти значення двох, що залишилися граничних колірних координат з умови збереження кольоровості;

- якщо $\text{Min}(\text{RGB}) = 0$, то виявити (з трьох) проміжну колірну координату $\text{Med}(\text{RGB})$;
- привласнити $\text{Med}(\text{RGB})_{\text{lim}}$ значення 1;
- якщо $\text{Med}(\text{RGB}) \neq 0$, то знайти значення однієї залишилася граничної колірної координати з умови збереження кольоровості;
- якщо $\text{Med}(\text{RGB}) = 0$, то виявити найбільшу (з трьох) колірну координату $\text{Max}(\text{RGB})$;
- привласнити $\text{Max}(\text{RGB})_{\text{lim}}$ значення 1;
- тепер все граничні колірні координати виявляться безумовно визначені, а саме $R_{\text{lim}} = G_{\text{lim}} = B_{\text{lim}} = 1$;

- обчислити граничне значення яскравості

11. Обчислення вихідної девіації яскравості першого пікселя

12. Обчислення нової девіації яскравості першого пікселя по формулі

13. Обчислення нової яскравості першого пікселя по формулі

14. Якщо $Y' \leq Y_{\text{lim}}$, то $Y' = Y_{\text{lim}}$;

якщо $Y' > Y_{\text{lim}}$, то $Y' = Y'$. Перехід до п. 19.

15. Оцінювання гранично можливого значення яскравості першого пікселя, досяжного в процесі яркостного контрастування при збереженні його кольоровості в разі збереження вихідної яскравості не затребуване.

Коли коефіцієнт $M = 1$, яскравість першого пікселя не змінюватиметься в процесі контрастування ні при якому значенні Y_0 . Отже, нова девіації яскравості першого пікселя і вихідна будуть однаковими.

16. Обчислення нової яскравості першого пікселя. Перехід до п. 19.

17. Оцінювання гранично можливого значення яскравості першого пікселя, досяжного в процесі яркостного контрастування при збереженні його кольоровості в разі обнулення девіації яскравості не затребуване.

Коли спочатку яскравість першого пікселя дорівнює опорної, вона не буде змінюватися при контрастируванні ні при якому значенні M . Отже, і вихідна, і нова девіації яскравості першого пікселя будуть нульовими:

- привласнити $\Delta Y' = 0$.

18. Обчислення нової яскравості першого пікселя

19. Обчислення нових значень колірних координат першого пікселя R' , G' і B' по відомим Y' , Y , R , G і B шляхом вирішення відповідної системи рівнянь. Результуючі колірні координати відшуковують послідовним виконанням однієї з чотирьох груп операцій:

при $R \neq 0$

$$R' = Y' / (K + 3 \cdot G / R + C \cdot B / R)$$

$$G' = G \cdot R' / R$$

$$B' = B \cdot R' / R,$$

при $G \neq 0$

$$G' = Y' / (K \cdot R / G + 3 + C \cdot B / G)$$

$$B' = B \cdot G' / G$$

$$R' = R \cdot G' / G,$$

при $B \neq 0$

$$B' = Y' / (K \cdot R / B + 3 \cdot G / B + C)$$

$$R' = R \cdot B' / B$$

$$G' = G \cdot B' / B,$$

при $R=G=B=0$

$$R' = Y'$$

$$G' = Y'$$

$$B' = Y'.$$

20. Перехід до другого пікселя і виконання циклу, починаючи з п. 3.

Перевагою запропонованого алгоритму контрастування є те, що він виконується за один прохід, і це головна передумова швидкодії для реалізованої на його основі програми. Технічні можливості програми. За результатами випробувань розробки: розмір оброблюваного файлу може досягати $25000 \times 25000 = 625 M_{pxl}$; допустима глибина кольору - 48bitRGB.

Приклад роботи алгоритму(див.рис 1.6-1.8):



Рис.1.6. Результат роботи алгоритму контрастування

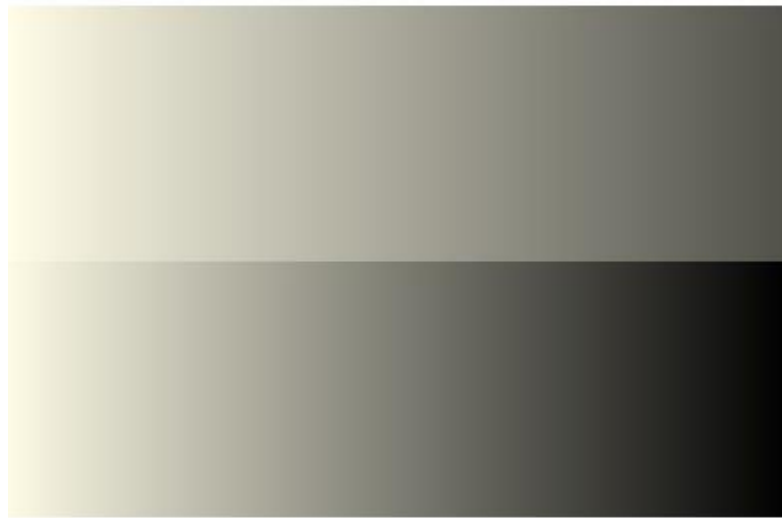


Рис.1.7. Результат роботи алгоритму контрастування



Рис.1.8. Результат роботи алгоритму контрастування

Перейдемо до огляду алгоритму сегментації. Пропонований алгоритм колірної сегментації зображення полягає в кластеризації кольорової палітри зображення шляхом розбиття на M тривимірних підпросторів (вокселів) колірного простору HSV (Hue - колірний тон; Saturation - насиченість; Value - значення кольору), обчисленні ваг w_i підпросторів (кількість пікселів зображення, колір яких належить цим підпросторам) і відрізняється скануванням колірного простору за допомогою тривимірного оператора пошуку локальних максимумів ваги шляхом аналізу околиці кожного підпростору (воксель).

Якщо два таких воксела розташовані ближче заданого мінімального відстані між собою, то підпростір з меншою вагою виключається зі списку кластеростворюваних. Далі до кластеростворюючого рекурсивно приєднуються сусідні підпростори, поки вага сусіднього простору менше ваги поточного. Результуюче зображення замальовується усередненими кольорами кластерів.

Простір XYZ можливих значень ознак з рівномірним кроком по кожній з осей розбитий на безліч підпросторів $W = \{w_{ijk}\}$, $i = 1, \dots, M$, $j = 1, \dots, N$; $k = 1, \dots, K$ підпросторів (вокселів) і обчислена вага w_{ijk} кожного з підпросторів - кількість пікселів зображення, у яких значення ознак потрапили в відповідне підпростір. Потрібно знайти безліч E підпросторів, в яких значення ваги є локальним максимумом.

Для кожного підпростору w_{ijk} сформуємо допоміжну безліч ваг W' сусідніх підпросторів

$$W'_{ijk} = \{w_{i-ljk}, w_{i+ljk}, w_{ij-lk}, w_{ij+lk}, w_{ijk-l}, w_{ijk+l}\}. \quad (1.26)$$

Тоді підпростір, в якому функція ваги набуває значення локального максимуму, визначається за допомогою простого правила:

$$\forall_{l=1}^{\sigma} (W_{ijk} > W'_l) \Rightarrow W_{ijk} \in E. \quad (1.27)$$

Оператор пошуку локальних максимумів кольорової палітри зображення аналогічний широко відомому оператору Собеля для виділення контурів, але працює в 3D просторі.

Щоб здійснити кластеризацію на задану кількість кольорів палітри, в яку увійшли б усі воксели колірного простору, а значить і всі пікселі зображення, необхідно вокселі, приєднати до найближчого кластеру.

Якщо між двома "важкими" кластерами є "легкий" який не увійшов до задану палітру, то він повинен бути поглинений, тобто розділений між ними. Це можна зробити двома способами:

1. по відстані від піку кластера, тоді горизонтальну ділянку притягнеться до кластеру, який містить більше вокселів;

2. по відстані від краю кластера, тоді горизонтальну ділянку вододілу буде рівномірно ділитися між сусідніми кластерами.

У першому випадку кожен з решти вокселів приписується до найближчого по відстані від піку кластера. У другому випадку для кожного з решти вокселів відшукується найближчий по відстані воксел з числа записаних в кластери на першому етапі. В обох випадках середні кольору кластерів, отримані на першому етапі, не коригуються.

Далі може бути виконана рекурсивна обробка отриманих колірних областей (для кожної отриманої області будується своє колірне простір). Для зручності зафарбовування зображення квітами отриманої палітри воксели містять адресні посилання на відповідні їм пікселі. Перехід на наступний, більш високий рівень кластеризації здійснюється з метою більшої деталізації об'єктів зображення. Наприклад, при аналізі рослинності на космічних знімках колір змінюється незначно, а насиченість і інтенсивність плавають майже по всьому діапазону. Більш точна колірна сегментація в областях, що представляють інтерес, дозволяє підвищити релевантність результатів аналізу зображення в цілому.

Виділення меж колірних областей здійснюється автоматично в ході рекурсивної зафарбовування зображення квітами, отриманими в результаті кластеризації кольорової палітри.

Піксель належить кордоні колірної області в тих випадках, коли хоча б один із сусідніх пікселів відноситься до іншого кластеру кольорової палітри.

Тепер слід розглянути алгоритм апроксимації. Апроксимація кордонів колірних областей зображення здійснюється для скорочення надлишкової інформації з метою спрощення і прискорення наступних процесів аналізу форми об'єктів і пошуку зображень за змістом.

Суть алгоритму полягає в рекурсивном нарощуванні дуг, аппроксимируючих ланцюжка кордонів. З цією метою ми спочатку виконуємо апроксимацію кривих B-сплайнами.

Точність апроксимації регулюється кількістю керуючих точок сплайна, яке ставиться в залежність від кількості вхідних точок.

Наступний крок - апроксимація сплайнів дугами кіл і відрізками прямих. Алгоритм приймає на вхід ланцюжок $O = (o_1, \dots, o_n)$ відрізків, що мають приблизно рівну довжину, тобто $\forall i | o_i | \approx const$.

Чим менше крок дискретизації, тим точніше може бути проведена апроксимація дугами кіл і відрізками прямих, проте потрібно більше обчислювальних ресурсів (в ході експериментів оптимальним виявився крок, рівний 2-4 пікселя, в залежності від розмірів об'єктів).

Алгоритм повертає ланцюжок дуг і відрізків $E = (e_1, \dots, e_m)$, що володіє наступними властивостями:

- 1) $m \leq n$;
- 2) елементи e_i відхиляються від початкового ланцюжка не більше ніж на $d \in \mathfrak{R}$;
- 3) кінці елементів e_i лежать на вихідній ланцюжку, тобто $\forall i \exists j: e_i \text{End} P o_{int} = o_j \text{End} P o_{int}$;
- 4) кути в E не сильно відрізняються від відповідних кутів в O , тобто $|\angle(T(e_i), T(e_{i+1})) - \angle(o_j, o_{j+1})| \leq 5^\circ$,

де $T(e_i)$ - дотичний вектор до елементу e_i в його кінцевий точці.

Процес апроксимації має рекурсивний характер. Спочатку на всьому ланцюжку визначається місце екстремуму (максимальний кут), потім з цієї точки відбувається побудова оптимальної апроксимуючої дуги / відрізка. Процес рекурсивно повторюється для підцепочек зліва і праворуч від створеної екстремальної дуги:

```
procedure CreateExtremeArc( $l, r$ )
   $v \leftarrow 0; w \leftarrow 0;$ 
  Find  $s \in [l, r - 1]$  such that  $|\angle(o_s, o_{s+1})| \rightarrow \max;$ 
  for  $i := s$  downto  $l$  do
    for  $j := s + 1$  to  $r$  do
       $e \rightarrow$  ApproximateSubchainByOneArcOrLine ( $i, j$ );
      if  $e$  satisfies the deviation and junction angle
      constraints then
         $v \leftarrow i; w \leftarrow j;$ 
  if ( $v \neq 0$ ) & ( $w \neq 0$ ) then
     $e \leftarrow$  ApproximateSubchainByOneArcOrLine ( $v, w$ );
    Add  $e$  to the result  $E$ ;
  else
    Add  $o_s$  to the result  $E$ ;
  if  $s > l$  then
    CreateExtremeArc( $l, s$ );
  if  $s + 1 < r$  then
    CreateExtremeArc( $s + 1, r$ );
```

Параметри l і r - кордони даної підцепочки, спочатку задаються рівними відповідно 1 і n .

В даному алгоритмі апроксимації додатково реалізовано три евристичних правила відсікання для скорочення переборівши:

1. Перервати цикл по j , коли розглянута підцепочка (i, j) по довжині перевищує поріг λ_1 .
2. Припинити цикл по j , коли розглянута підцепочка (i, j) по довжині перевищує поріг λ_2 і при цьому жодна задовільна апроксимуюча дуга від i до j ще не була побудована.
3. Припинити цикл по j , якщо $(j - i) - m > 20$, де m - кількість відрізків, що покриваються найбільшою на даний момент апроксимуючої дугою.

Точність апроксимації регулюється параметром d - порогом допустимого відхилення.

Ми задали d рівним 0.4 пікселя. При цьому коефіцієнт стиснення інформації про межі в середньому дорівнює 3

Приклади роботи алгоритму (див.рис. 1.9-1.12):

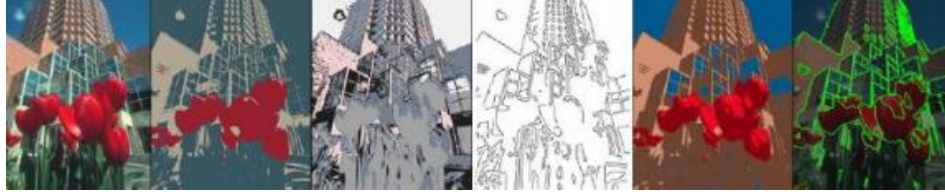


Рис.1.9. Приклад роботи алгоритму апроксимації та сегментації

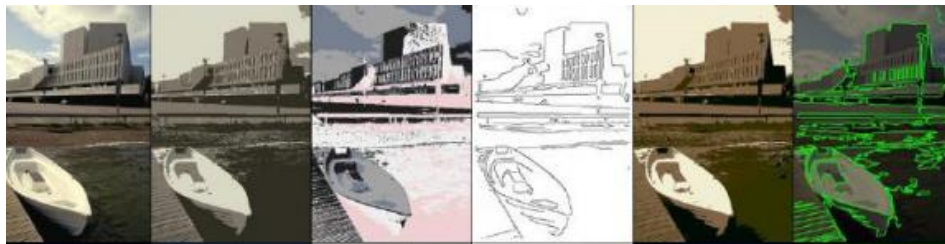


Рис.1.10. Приклад роботи алгоритму апроксимації та сегментації

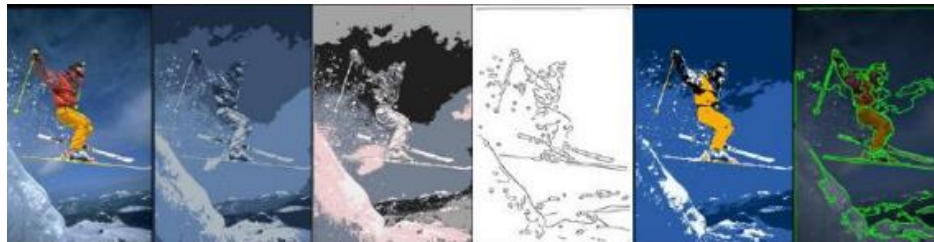


Рис.1.11. Приклад роботи алгоритму апроксимації та сегментації



Рис.1.12. Приклад роботи алгоритму апроксимації та сегментації

РОЗДІЛ 2. ОГЛЯД МЕТОДІВ ТРАНСФЕРНОГО НАВЧАННЯ

2.1. Постановка задачі трансферного навчання

Технології обробки даних та машинного навчання вже досягли значних успіхів у багатьох областях інженерії, включаючи класифікацію, регресію та кластеризацію (наприклад, [16], [17]). Однак багато методів машинного навчання добре працюють лише за єдиного припущення: дані навчання та тестування беруться з одного простору функцій та однакового розподілу.

Коли розподіл змінюється, більшість статистичних моделей потрібно будувати з нуля, використовуючи нещодавно зібрані навчальні дані. У багатьох реальних програмах повторно збирати необхідні навчальні дані та перебудувати моделі дорого або взагалі неможливо. Тому непогано зменшити потребу та зусилля щодо повторного збору даних для навчання. У таких випадках бажано передати знання або передати навчання між доменами завдань.

Можна знайти багато прикладів, де навчання за допомогою трансферу може бути справді корисним. Одним із прикладів є класифікація веб-документів [18], [19], [20], де метою є класифікація веб-документа на кілька визначених категорій. Як приклад у галузі класифікації веб-документів [21], позначеними прикладами можуть бути університетські веб-сторінки, які пов'язані з інформацією про категорії, отриманою завдяки попереднім зусиллям щодо маркування вручну.

Для завдання класифікації на новоствореному веб-сайті, де функції даних або розподіл даних можуть бути різними, може не вистачати розмічених навчальних даних. Як результат, ми можемо не мати можливості безпосередньо застосувати класифікатори веб-сторінок, вивчені на веб-сайті університету, до нового веб-сайту. У таких випадках було б корисно, перенести знання про класифікацію в новий домен.

КАФЕДРА АКІК

НАУ 20 19 32 000 ПЗ

<i>Розроб.</i>	<i>Хоцянівський В.П.</i>			<i>Інтелектуальна медична система на основі нейронних мереж RESNET</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Керівник</i>	<i>Синєглазов В.М.</i>						
<i>Консульт.</i>							
<i>Н.контр.</i>	<i>Тупіцин М. Ф.</i>						
<i>Зав.кафедри</i>	<i>Синєглазов В.М.</i>						
					<i>151 204М</i>		

Потреба в передачі навчання може виникнути, коли дані швидко старіють. У такому випадку розмічені дані, отримані за один проміжок часу, можуть не мати такого самого розподілу в інший проміжок часу.

Наприклад, у проблемах локалізації WiFi у приміщенні, яка спрямована на виявлення поточного місцезнаходження користувача на основі раніше зібраних даних WiFi, дуже дорого калібрувати дані WiFi для побудови моделей локалізації у великому масштабі, оскільки користувачеві потрібно розмітити велику вибірку даних про сигнали WiFi у кожному місці. Однак значення потужності сигналу WiFi може бути залежним від часу, пристрою та інших факторів. Модель, навчена протягом одного періоду часу або на одному пристрої, може давати некоректну оцінку місцезнаходження в інший період часу або на іншому пристрої. Щоб зменшити зусилля з повторного калібрування, потрібно адаптувати модель локалізації, навчену за один період часу (вихідний домен), для нового періоду часу (цільовий домен), або адаптувати модель локалізації, навчену на мобільному пристрої (вихідний домен) для нового мобільного пристрою (цільовий домен) [22].

Як третій приклад, розглянемо проблему класифікації відгуків, де завдання полягає в тому, щоб автоматично класифікувати відгуки про товар, наприклад, бренд камери, на позитивні та негативні відгуки. Для цього завдання класифікації нам потрібно спочатку зібрати безліч відгуків про товар та розмітити їх. Потім потрібно підготувати класифікатор з відповідними налаштуваннями. Оскільки розподіл даних оглядів між різними типами товарів може бути різним, для підтримки належної класифікації потрібно зібрати велику кількість розмічених даних, щоб навчити моделі класифікації оглядів для кожного товару. Однак процес розмічення даних може бути дуже дорогим. Щоб зменшити зусилля для коментування відгуків для різних товарів, ми можемо захотіти адаптувати модель класифікації, яка навчена для деяких товарів, щоб допомогти вивчити моделі класифікації для інших продуктів. У таких випадках навчання за допомогою трансферу може заощадити значну кількість зусиль на маркуванні [23].

У літературі про машинне навчання (наприклад, [24], [25]) було проведено велику кількість робіт з трансфертного навчання для навчання з підкріпленням. У цій роботі ми зосереджуємось на вивченні передачі для проблем класифікації, регресії та кластеризації, які більш тісно пов'язані із завданнями інтелектуального аналізу даних.

2.2 Огляд трансферного навчання

Традиційні алгоритми видобутку даних та машинного навчання роблять прогнози, використовуючи статистичні моделі, які тренуються на раніше зібраних розмічених або нерозмічених навчальних даних [26], [27], [28]. Класифікація [29], [30], [31], [32] вирішує проблему, що мічених даних може бути замало для створення хорошого класифікатора, використовуючи велику кількість немаркованих даних і невелику кількість маркованих даних. Вивчені варіації контрольованого та напівконтрольованого навчання для недосконалих наборів даних; вивчали, як боротися з проблемами міток класу. Оскільки це розглядається як економічне навчання [33], коли додаткові тести можна зробити для майбутніх зразків. Тим не менше, більшість із них припускають, що розподіл маркованих та немаркованих даних однаковий. Трансферне навчання, навпаки, дозволяє відрізнити домени, завдання та розподіли, що використовуються при навчанні та тестуванні. У реальному світі ми спостерігаємо багато прикладів трансферного навчання. Наприклад, ми можемо виявити, що система навчена розпізнавати яблука може допомогти розпізнати груші. Подібним чином навчання гри на електронних музичних інструментах може допомогти полегшити навчання гри на фортепіано. Вивчення трансферного навчання мотивоване тим, що люди можуть розумно застосовувати знання, отримані раніше, для швидшого або кращого вирішення нових проблем.

Дослідження з питань трансферного навчання привертають все більше уваги з 1995 р. Під різними назвами: навчання вчитися, навчання протягом усього життя, передача знань, індуктивна передача, багатозадачне навчання,

консолідація знань, контекстне навчання, індуктивне упередження на основі знань, мета навчання, та поступове/кумулятивне навчання [34]. Серед них, тісно пов'язаною технікою навчання для передачі навчання є багатозадачна система навчання [35], яка намагається вивчити кілька завдань одночасно, навіть коли вони різні. Типовим підходом до багатозадачного навчання є розкриття загальних (прихованих) особливостей, які можуть бути корисними для кожного окремого завдання.

У 2005 році агентство оборонних дослідницьких проєктів (DARPA) дало нову місію трансфертному навчанню: здатність системи розпізнавати та застосовувати знання та навички, отримані в попередніх завданнях, для нових завдань. У цьому визначенні трансфертне навчання має на меті витягти знання з одного або декількох вихідних завдань і застосувати знання для цільового завдання. На відміну від багатозадачного навчання, а не вивчення всіх вихідних та цільових завдань одночасно, трансферне навчання найбільше дбає про цільове завдання. Ролі вихідних та цільових завдань перестали бути симетричними в процесі навчання.

На рис. 2.1 показано різницю між навчальними процесами традиційних та трансферних методів навчання. Тобто традиційні техніки машинного навчання намагаються вивчити кожне завдання з нуля, тоді як методи передачі навчання намагаються перенести знання з деяких попередніх завдань до цільового завдання, коли останні мають менше високоякісних навчальних даних.

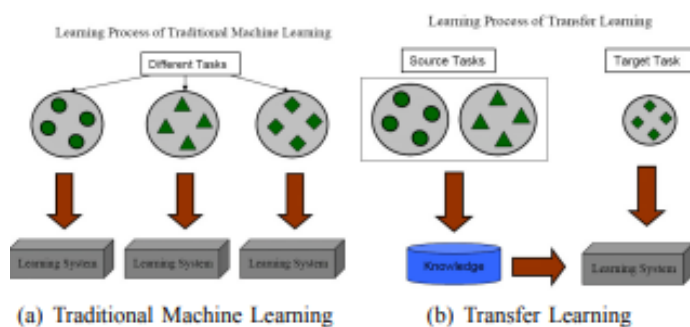


Рис. 2.1. Різні навчальні процеси між традиційним машинним навчанням та трансферним навчанням

Сьогодні методи трансферного навчання з'являються у провідних розробках, особливо в інтелектуальному аналізі даних, машинному навчанні та застосуванні машинного навчання в інтелектуальному аналізі даних.

Перш за все, потрібно дати визначення поняття «домен» та «завдання» відповідно. Нехай область D складається з двох компонентів: простору ознак X та граничного розподілу ймовірностей $P(X)$, де $X = \{x_1, \dots, x_n\} \in X$. Наприклад, якщо наше навчання - класифікація документів, і кожен термін приймається як двійкова ознака, тоді X - простір усіх термінових векторів, x_i - i -й вектор терміна, що відповідає деяким документам, а X - конкретна навчальна вибірка.

Загалом, якщо два домени різні, то вони можуть мати різні простори ознак або різні граничні розподіли ймовірностей. Враховуючи конкретний домен, $D = \{X, P(X)\}$, завдання складається з двох компонентів: простору міток Y та об'єктивної передбачувальної функції $f(\bullet)$ (позначається $T = \{Y, f(\bullet)\}$), що не спостерігається, але можна дізнатись із навчальних даних, які складаються з пар $\{x_i, y_i\}$, де $x_i \in X$ та $y_i \in Y$. Функцію $f(\bullet)$ можна використовувати для прогнозування відповідної мітки $f(x)$, нового екземпляра x . З імовірнісної точки зору $f(x)$ можна записати як $P(y/x)$. У прикладі класифікації документів Y - це набір усіх міток, тобто *True*, *False* для двійкового завдання класифікації, а y_i - "*True*" або "*False*".

Для простоти в цьому опитуванні розглядаємо випадок, коли існує один вихідний домен D_S та один цільовий домен D_T , оскільки це, найпопулярніша з дослідницьких робіт у літературі. Більш конкретно, ми позначаємо вихідні дані домену як $D_S = \{(x_{S1}, y_{S1}), \dots, (x_{Sn_S}, y_{Sn_S})\}$, де $X_{Si} \in X_S$ - екземпляр даних, а $Y_{Si} \in Y_S$ - відповідна мітка класу. У нашому прикладі класифікації документів D_S може бути набором векторів термінів разом із пов'язаними з ними мітками істинних чи помилкових класів. Подібним чином ми позначаємо дані цільового домену як $D_T = \{(X_{T1}, Y_{T1}), \dots, (X_{Tn_T}, Y_{Tn_T})\}$, де вхід X_{Ti} знаходиться в X_T , а $Y_{Ti} \in Y_T$ - відповідний вихід. У більшості випадків $0 \leq n_T \leq n_S$.

Тобто з огляду на вихідний домен D_S та навчальне завдання T_S , цільовий домен D_T та навчальне завдання T_T , трансферне навчання має на меті покращити засвоєння цільової передбачувальної функції $f_T(\bullet)$ в D_T , використовуючи знання в D_S та T_S , де $D_S \neq D_T$, або $T_S \neq T_T$.

У наведеному вище визначенні домен - це пара $D = \{X, P(X)\}$. Таким чином, умова $D_S \neq D_T$ передбачає, що $X_S \neq X_T$, або $P_S(X) \neq P_T(X)$. Наприклад, у нашому прикладі класифікації документів це означає, що між набором вихідних документів і цільовим набором документів, ознаки терміна відрізняються між двома наборами (наприклад, вони використовують різні мови), або їх граничний розподіл різних.

Подібним чином завдання визначається як пара $T = \{Y, P(Y | X)\}$. Таким чином, умова $T_S \neq T_T$ передбачає, що або $Y_S \neq Y_T$, або $P(Y_S/X_S) \neq P(Y_T/X_T)$. Коли цільовий і вихідний домени однакові, тобто $D_S = D_T$, а їх навчальні завдання однакові, тобто $T_S = T_T$, проблема навчання стає традиційною проблемою машинного навчання. Коли домени різні, тоді простори об'єктів між доменами різні, тобто $X_S \neq X_T$, або простори об'єктів між доменами однакові, але граничні розподіли ймовірностей між даними домену різні; тобто $P(X_{Si}) \neq P(X_{Ti})$, де $X_{Si} \in X_S$ та $X_{Ti} \in X_T$. Наприклад, у прикладі класифікації документів випадок відповідає тому, що два набори документів описані різними мовами, а випадок може відповідати тому, коли вихідні документи домену та документи цільового домену зосереджені на різних темах.

Враховуючи конкретні домени D_S і D_T , коли навчальні завдання T_S і T_T відрізняються, простори міток між доменами різні, тобто $Y_S \neq Y_T$, або (2) умовні розподіли ймовірностей між доменами різні; тобто $P(Y_S/X_S) \neq P(Y_T/X_T)$, де $Y_{Si} \in Y_S$ та $Y_{Ti} \in Y_T$. У прикладі класифікації документів випадок відповідає ситуації, коли вихідний домен має бінарні класи документів, тоді як цільовий домен має десять класів, до яких можна класифікувати документи. Випадок відповідає ситуації, коли вихідні та цільові документи дуже незбалансовані з точки зору визначених користувачем класів. Крім того, коли існує певний

зв'язок, явний чи неявний, між просторами ознак двох доменів, говорять, що вихідний та цільовий домени пов'язані.

У процесі трансферного навчання ми маємо такі три основні наукові проблеми:

- Що передати;
- Як передати;
- Коли передавати.

“Що передати” запитує, яку частину знань можна передати між доменами або завданнями. Деякі знання специфічні для окремих доменів або завдань, а деякі знання можуть бути спільними між різними доменами, таким чином, що вони можуть допомогти поліпшити продуктивність цільового домену або завдання. Після виявлення того, які знання можна передати, необхідно розробити алгоритми навчання для передачі знань, що відповідає питанням «як передати».

“Коли передавати” запитує, в яких ситуаціях слід робити навички передачі. Так само нам цікаво знати, в яких ситуаціях знання не слід передавати. У деяких ситуаціях, коли вихідний і цільовий домени не пов'язані між собою, груба сила може бути невдалою. У найгіршому випадку це може навіть зашкодити результатам навчання у цільовій області, ситуація, яку називають негативною передачею. Більшість сучасних робіт з навчання трансферу зосереджуються на питаннях "Що передати" та "Як передати", неявно припускаючи, що вихідний та цільовий домени пов'язані між собою. Однак, як уникнути негативних передач - важливе питання, яке привертає все більше уваги в майбутньому.

На основі визначення трансферного навчання ми підсумовуємо взаємозв'язок між традиційним машинним навчанням та різними налаштуваннями трансферного навчання в таблиці 2.1, де класифікуємо трансферне навчання за трьома підналаштуваннями, індуктивне трансферне навчання, трансдуктивне трансферне навчання та безконтрольне трансферне

навчання, засноване на різні ситуації між вихідним та цільовим доменами та завданнями.

1) В налаштуваннях індуктивного навчання передачі цільове завдання відрізняється від вихідного, незалежно від того, коли вихідний та цільовий домени однакові чи ні. У такому випадку деякі розмічені дані в цільовому домені необхідні для наведення об'єктивної прогностичної моделі $f_T(\bullet)$ для використання в цільовому домені. Крім того, відповідно до різних ситуацій мічених та немаркованих даних у вихідному домені, ми можемо додатково класифікувати індуктивне налаштування навчання на два випадки:

Таблиця 2.1. Зв'язок між традиційним машинним навчанням та різними налаштуваннями трансферного навчання

Learning Settings		Source and Target Domains	Source and Target Tasks
Traditional Machine Learning		the same	the same
Transfer Learning	<i>Inductive Transfer Learning /</i>	the same	different but related
	<i>Unsupervised Transfer Learning</i>	different but related	different but related
	<i>Transductive Transfer Learning</i>	different but related	the same

(1.1) Доступно багато мічених даних у вихідному домені. У цьому випадку налаштування індуктивного перекладу аналогічні налаштуванням багатозадачного навчання. Однак установка індуктивного перекладу має на меті лише досягнення високої продуктивності цільового завдання шляхом передачі знань із вихідного завдання, тоді як багатозадачне навчання намагається вивчити цільове та вихідне завдання одночасно.

(1.2) Немає розмічених даних у вихідному домені. У такому випадку установка індуктивного перекладу є подібною до обстановки самонавчання[36]. В умовах самонавчання простори міток між вихідним та цільовим доменами можуть бути різними, що означає, що побічна інформація вихідного домену не може використовуватися безпосередньо. Таким чином, це схоже на налаштування індуктивного навчання передачі, де розмічені дані у вихідному домені недоступні.

2) В умовах трансдуктивного навчання передачі джерело та цільові завдання однакові, тоді як джерела та цільові домени різні. У цій ситуації недоступні мічені дані в цільовому домені, тоді як доступно багато мічених даних у вихідному домені. Крім того, відповідно до різних ситуацій між

вихідним та цільовим доменами, ми можемо додатково класифікувати параметри трансдуктивного навчання на два випадки.

(2.1) Простіри функцій між вихідним та цільовим доменами різні, $X_S \neq X_T$.

(2.2) Простіри характеристик між доменами однакові, $X_S = X_T$, але граничні розподіли ймовірностей вхідних даних різні, $P(X_S) \neq P(X_T)$.

Останній випадок налаштування навчання трансдуктивного переносу пов'язаний з адаптацією домену для передачі знань в області класифікації тексту[37] та упередженням вибірки вибірки [38] або зміщенням ко-змінних [39], припущення яких подібні.

3) В умовах неконтрольованого навчання передачі, подібно до налаштувань індуктивного навчання передачі, цільове завдання відрізняється, але пов'язане з вихідним завданням. Однак безконтрольне трансферне навчання фокусується на вирішенні неконтрольованих навчальних завдань у цільовій області, таких як кластеризація, зменшення розмірності та оцінка щільності [40], [41]. У такому випадку під час навчання відсутні розмічені дані як у вихідному, так і в цільовому доменах. Взаємозв'язок між різними установками трансферного навчання та суміжними сферами узагальнено див. в таблиці 2.2 та рис.2.1.

Таблиця 2.2. Різні налаштування навчання трансферу

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
<i>Inductive Transfer Learning</i>	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
<i>Transductive Transfer Learning</i>	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
<i>Unsupervised Transfer Learning</i>		Unavailable	Unavailable	Clustering, Dimensionality Reduction

Підходи до передачі навчання у вищезазначених умовах можна узагальнити до чотирьох випадків на основі “Що передати”. У таблиці 2.3 наведені ці чотири випадки та короткий опис.

Таблиця 2.3. Різні підходи до трансферного навчання

Transfer Learning Approaches	Brief Description
<i>Instance-transfer</i>	To re-weight some labeled data in the source domain for use in the target domain [6], [28], [29], [30], [31], [24], [32], [33], [34], [35].
<i>Feature-representation-transfer</i>	Find a “good” feature representation that reduces difference between the source and the target domains and the error of classification and regression models [22], [36], [37], [38], [39], [8], [40], [41], [42], [43], [44].
<i>Parameter-transfer</i>	Discover shared parameters or priors between the source domain and target domain models, which can benefit for transfer learning [45], [46], [47], [48], [49].
<i>Relational-knowledge-transfer</i>	Build mapping of relational knowledge between the source domain and the target domains. Both domains are relational domains and i.i.d assumption is relaxed in each domain [50], [51], [52].

Перший контекст можна назвати підходом до вивчення (або екземпляра) на основі примірника [42], [43], [44], [45], [46], [47], [48], [49] який передбачає, що певні частини даних у вихідному домені можуть бути використані повторно для навчання в цільовому домені шляхом повторного зважування. Переважування екземплярів та вибірка важливості є двома основними методами в цьому контексті.

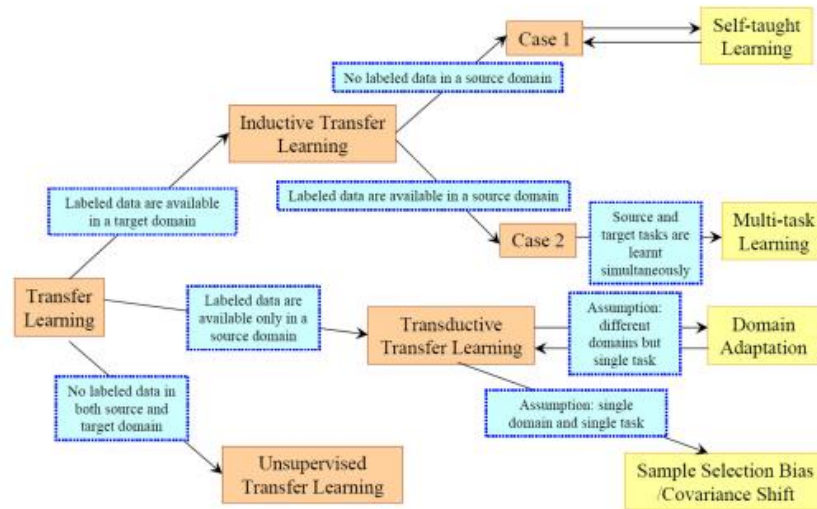


Рис. 2.2. Огляд різних налаштувань передачі

Другий випадок можна назвати підходом передачі ознак [50], [51], [52], [53], [54], [55], [56], [57], [58]. Ідея, полягає в тому, щоб навчитися “хорошому” представленню функцій цільового домену. У такому випадку знання, що використовуються для передачі між доменами, кодуються у представлення вивченого об’єкта. З представленням нової функції, очікується, що цільове завдання значно покращиться.

Третій випадок можна назвати підходом передачі параметрів [59], [60], [61], [62], [63], який передбачає, що вихідні завдання та цільові завдання мають спільні параметри або попередні розподіли гіперпараметри моделей. Передані знання кодуються у загальні параметри або пріоритети. Таким чином, виявляючи спільні параметри або попередні знання, можна передавати знання між завданнями.

Нарешті, останній випадок можна назвати проблемою передачі реляційних знань [64], який стосується навчання передачі для реляційних

доменів. Основним припущенням цього контексту є те, що деякі відносини між даними у вихідному та цільовому доменах подібні. Таким чином, знання, що підлягають передачі, - це співвідношення між даними. Останнім часом статистичні методи реляційного навчання домінують у цьому контексті [65], [66].

У таблиці 2.4 наведені випадки, коли різні підходи використовуються для кожної обстановки навчання переказу. Ми можемо бачити, що індуктивний режим навчання передачі вивчався у багатьох дослідницьких роботах, тоді як безконтрольний режим навчання трансферу є відносно новою темою дослідження і вивчається лише в контексті випадку передачі ознак. Крім того, для всіх трьох параметрів навчання передачі пропонується проблема представлення функцій-передачі. Однак підхід до передачі параметрів та реляційного передачі знань вивчається лише в умовах індуктивного навчання передачі.

Таблиця 2.4. Підходи, що використовуються в різних налаштуваннях

	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
<i>Instance-transfer</i>	✓	✓	
<i>Feature-representation-transfer</i>	✓	✓	✓
<i>Parameter-transfer</i>	✓		
<i>Relational-knowledge-transfer</i>	✓		

2.3 Індуктивна передача навчання

Враховуючи вихідний домен D_S та навчальне завдання T_S , цільовий домен D_T та навчальне завдання T_T , індуктивне навчання з передачі має на меті допомогти покращити засвоєння цільової передбачувальної функції $f_T(\bullet)$ у D_T за допомогою знання в D_S та T_S , де $T_S \neq T_T$.

Виходячи з вище наведеного визначення індуктивного налаштування навчання, в якості навчальних даних для наведення цільової передбачувальної функції потрібно кілька позначених даних у цільовій області. Як згадано у вище, цей параметр має два випадки:

- Доступні розмічені дані у вихідному домені;

- Розмічені дані у вихідному домені недоступні, тоді як доступні немарковані дані у вихідному домені. Більшість підходів до трансферного навчання в цій ситуації зосереджені на першому випадку.

Підхід передачі екземпляра для налаштування індуктивного навчання передачі є привабливим: хоча вихідні дані домену не можна використовувати повторно безпосередньо, є певні частини даних, які все ще можуть бути використані разом із кількома позначеними даними в цільовому домені.

Тому було запропоновано алгоритм посилення[21], *TrAdaBoost*, який є розширенням алгоритму *AdaBoost*, для вирішення проблем індуктивного навчання при передачі. *TrAdaBoost* припускає, що вихідні та цільові дані домену використовують абсолютно однаковий набір функцій та міток, але розподіл даних у двох доменах різний. Крім того, *TrAdaBoost* припускає, що через різницю в розподілах між вихідним та цільовим доменами деякі дані вихідного домену можуть бути корисними для вивчення цільового домену, але деякі з них можуть не бути і навіть можуть бути шкідливими. Він намагається повторно зважити дані вихідного домену, щоб зменшити вплив "поганих" вихідних даних, заодно заохочуючи "хороші" вихідні дані щоб сприяти цільовому домену. Для кожного раунду ітерації *TrAdaBoost* тренує базовий класифікатор на зважених вихідних та цільових даних. Помилка обчислюється лише на цільових даних. Крім того, *TrAdaBoost* використовує ту саму стратегію, що і *AdaBoost*, для оновлення неправильно класифікованих прикладів у цільовому домені, використовуючи іншу стратегію, ніж *AdaBoost*, для оновлення неправильно класифікованих прикладів джерел у вихідному домені[21].

Виходячи з цього було запропоновано евристичний метод для видалення "оманливих" прикладів навчання з вихідної області на основі різниці між умовними ймовірностями $P(y_T/x_T)$ і $P(y_S/x_S)$ [44]. Тому Ляо та його команда[45] запропонували новий метод активного навчання для вибору нерозмічених

даних у цільовому домені, які будуть мічені за допомогою вихідних даних домену. Згодом механізм[67] інтегрували вихідні (допоміжні) дані в структуру SVM для поліпшення класифікаційних характеристик.

Підхід репрезентації ознак до проблеми індуктивного навчання передачі спрямований на пошук “хороших” репрезентацій ознак, щоб мінімізувати розбіжність доменів та помилку моделі класифікації або регресії. Стратегії пошуку “хороших” подань функцій різні для різних типів вихідних даних домену. Якщо в вихідному домені доступно багато розмічених даних, для побудови представлення об’єкта можна використовувати контрольовані методи навчання. Це схоже на загальне вивчення особливостей у галузі багатозадачного навчання [54]. Якщо в вихідному домені відсутні мічені дані, для побудови представлення ознак пропонуються некеровані методи навчання.

Контрольовані методи побудови ознак для налаштування індуктивного перекладу подібні до тих, що використовуються при багатозадачному навчанні. Основна ідея полягає в тому, щоб навчитися низьковимірному поданню, яке поділяється між відповідними завданнями. Крім того, вивчене нове подання може також зменшити помилку класифікації або моделі регресії кожного завдання. Тому було запропоновано рідкісний метод навчання для багатозадачного навчання[54]. В умовах індуктивного навчання передачі загальні риси можна вивчити, вирішивши задачу оптимізації, подану наступним чином.

$$\arg \min_{A,U} \sum_{t \in \{T,S\}} \sum_{i=1}^{n_t} L(y_{t_i}, \langle a_t, U^T x_{t_i} \rangle) + \gamma \|A\|_2^2. \quad (2.1)$$

У цьому рівнянні S та T позначають завдання у вихідному та цільовому доменах відповідно. $A = [a_S, a_T] \in R^{d \times 2}$ - матриця параметрів. U - ортогональна матриця $d \times d$ (функція відображення) для відображення вихідних даних великого розміру в представлення низьких розмірів. (r, p) - норма A визначається як $\|A\|_{r,p} := (\sum_{i=1}^d \|a^i\|_r^p)^{\frac{1}{p}}$. (2.2).

Задача оптимізації (1) одночасно оцінює низьковимірні подання $U^T X_T$, $U^T X_S$ та параметри A моделі.

Задачу оптимізації (1) можна додатково перетворити на еквівалентну опуклу формулювання оптимізації та ефективно її вирішити. В подальшому було [41] запропоновано структуру спектральної регуляризації на матрицях для вивчення багатозадачної структури. Що призвело [56] до пропозиції створити опуклий алгоритм оптимізації для одночасного вивчення мета-пріоритетів та ваг функцій із ансамблю відповідних завдань прогнозування. Мета-пріоритети можна передавати між різними завданнями. Джебара [57] запропонувала вибрати функції для багатозадачного навчання за допомогою SVM. Рукерт та ін. [58] розробив підхід до індуктивного перенесення, який спрямований на пошук відповідного ядра для цільових даних.

У роботі [36] автор запропонував застосовувати розріджене кодування [68], яке є некерованим методом побудови ознак, для вивчення особливостей вищого рівня для навчання передачі. Основна ідея цього підходу складається з двох етапів. На першому кроці базові вектори вищого рівня $b = \{b_1, b_2, \dots, b_s\}$ вивчаються на вихідних даних домену шляхом вирішення задачі оптимізації:

$$\begin{aligned} \min_{a,b} \sum_i \|x_{S_i} - \sum_j a_{S_i}^j b_j\|_2^2 + \beta \|a_{S_i}\|_1 \\ \text{s.t.} \quad \|b_j\|_2 \leq 1, \forall j \in 1, \dots, s \end{aligned} \quad (2.3)$$

У цьому рівнянні $a_{S_i}^j$ є новим поданням базису b_j для вхідного x_{S_i} , а β - коефіцієнтом, що врівноважує член побудови ознаки та термін регуляризації. Після вивчення базових векторів b , на другому кроці, алгоритм оптимізації (3) застосовується до даних цільового домену для вивчення функцій вищого рівня на основі базових векторів b .

$$a_{T_i}^* = \arg \min_{a_{T_i}} \|x_{T_i} - \sum_j a_{T_i}^j b_j\|_2^2 + \beta \|a_{T_i}\|_1 \quad (2.4)$$

Нарешті, дискримінаційні алгоритми можуть бути застосовані до $\{a_{T_i}^*\}$ з відповідними мітками для підготовки моделей класифікації або регресії для

використання в цільовій області. Одним недоліком цього методу є те, що так звані базові вектори вищого рівня, вивчені на вихідному домені в задачі оптимізації (2), можуть бути непридатними для використання в цільовому домені. Нещодавно для трансферного навчання були адаптовані різноманітні методи навчання. У роботі [58] було запропоновано підхід на основі аналізу Прокруста до вирівнювання колекторів без відповідностей, який можна використовувати для передачі знань між доменами через вирівняні колектори.

Більшість підходів до передачі параметрів до налаштування навчання індуктивного переносу припускають, що окремі моделі для відповідних завдань повинні містити деякі параметри або попередній розподіл гіперпараметрів. Більшість підходів, включаючи систему регуляризації та ієрархічну байєсівську структуру, призначені для роботи в рамках багатозадачного навчання. Однак їх можна легко модифікувати для трансферного навчання. Як вже згадувалося, багатозадачне навчання намагається вивчити як вихідне, так і цільове завдання одночасно і досконало, тоді як навчання за допомогою трансферу має на меті лише підвищення ефективності цільового домену за допомогою даних вихідного домену. Таким чином, під час багатозадачного навчання ваги функцій втрат для вихідних та цільових даних однакові. І навпаки, при навчанні передачі ваги функцій втрат для різних доменів можуть бути різними. Інтуїтивно ми можемо призначити більшу вагу функції втрати цільовому домену, щоб переконатися, що ми можемо досягти кращої продуктивності в цільовому домені.

Лоуренс Платт [59] запропонував ефективний алгоритм, відомий як MT-IVM, який базується на Гаусових процесах (GP) для обробки багатозадачного навчального випадку. MT-IVM намагається вивчити параметри гауссового процесу за кількома завданнями, використовуючи одного і того ж лікаря загальної практики. Бонілла та ін. [60] також досліджував багатозадачне навчання в контексті загальної практики. Автори запропонували використовувати коваріаційну матрицю вільної форми над завданнями для моделювання міжзадачних залежностей, де GP-пріор використовується для

індукції кореляції між завданнями. Швайггофер[61] запропонував використовувати ієрархічну байєсівську структуру (НБ) разом із загальним лікарем для багатозадачного навчання. Окрім передачі пріоритетів моделей GP, деякі дослідники також пропонували передавати параметри SVM в рамках регуляризації.

Також в деяких роботах автори [62] запозичили ідею байєсівської структури у SVM для навчання з багатьма завданнями. Запропонований метод передбачав, що параметр w в SVM для кожного завдання можна розділити на два терміни. Один - це загальний термін над завданнями, а інший - це термін, що стосується конкретного завдання.

В процесі індуктивного навчання трансферу,

$$w_S = w_0 + v_S \text{ and } w_T = w_0 + v_T, \quad (2.5)$$

де, w_S та w_T - параметри SVM для вихідного завдання та цільового навчального завдання, відповідно. w_0 - загальний параметр, тоді як v_S та v_T - конкретні параметри для вихідного завдання та цільового завдання відповідно. Припускаючи, що $f_t = w_t \cdot x$ є гіперплощиною для завдання t , розширення SVM до багатозадачного навчального випадку можна записати так:

$$\begin{aligned} \min_{w_0, v_t, \xi_{t_i}} \quad & J(w_0, v_t, \xi_{t_i}) \quad (4) \\ = \quad & \sum_{t \in \{S, T\}} \sum_{i=1}^{n_t} \xi_{t_i} + \frac{\lambda_1}{2} \sum_{t \in \{S, T\}} \|v_t\|^2 + \lambda_2 \|w_0\|^2 \\ \text{s.t.} \quad & y_{t_i} (w_0 + v_t) \cdot x_{t_i} \geq 1 - \xi_{t_i}, \\ & \xi_{t_i} \geq 0, \quad i \in \{1, 2, \dots, n_t\} \text{ and } t \in \{S, T\}. \quad (2.6) \end{aligned}$$

Вирішивши вищезазначену задачу оптимізації, ми можемо визначити параметри w_0 , v_S та v_T одночасно. Кілька дослідників продовжували застосовувати підхід до передачі параметрів. Гао та ін. [63] запропонував локально зважену ансамблевую систему навчання для поєднання кількох моделей для навчання на трансфері, де ваги динамічно призначаються відповідно до прогнозованої потужності моделі на кожному тестовому прикладі в цільовій області.

На відміну від інших трьох контекстів, реляційний підхід до передачі знань має справу з проблемами навчання передачі в реляційних доменах і може бути представлений множинними відносинами, такими як мережеві дані та дані соціальних мереж. Цей підхід не передбачає, що дані, що отримуються з кожного домену, повинні бути незалежними та однаково розподіленими. Він намагається перенести зв'язок між даними з вихідного домену в цільовий домен. У цьому контексті для вирішення цих проблем пропонуються статистичні методи реляційного навчання.

Для вирішення цього завдання[64] було запропоновано алгоритм TAMAR, який передає реляційні знання з логічними мережами Маркова (MLN) через реляційні домени. MLN [68] - це потужний формалізм, який поєднує компактну виразність логіки першого порядку з гнучкістю ймовірності для статистичного реляційного навчання.

У MLN сутності в реляційному домені представлені предикатами, а їх відносини представлені в логіці першого порядку. TAMAR мотивовано тим, що якщо два домени пов'язані між собою, можуть існувати зіставлення для з'єднання сутностей та їх зв'язків із вихідного до цільового домену. Наприклад, можна вважати, що професор відіграє подібну роль в академічній галузі як менеджер у галузі управління промисловістю. Крім того, стосунки між професором та його студентами схожі на стосунки між керівником та його працівниками.

Таким чином, може існувати відображення від професора до менеджера та відображення відносин професора-студента до відносин менеджер-працівник. У цьому ключі TAMAR намагається використовувати MLN, навчений для вихідного домену, щоб допомогти у вивченні MLN для цільового домену. В основному, TAMAR - це двоступеневий алгоритм. На першому кроці створюється зіставлення від вихідного MLN до цільового домену на основі зваженої міри псевдо-лог-правдоподібності (WPLL). На другому етапі здійснюється перегляд відображеної структури в цільовій області за допомогою алгоритму FORTE [69], який є алгоритмом індуктивного

логічного програмування (ILP) для перегляду теорій першого порядку. Перероблений MLN може бути використаний як реляційна модель для висновків або міркувань у цільовому домені. Основна ідея алгоритму полягає у виявленні структурних закономірностей у вихідному домені у вигляді формул логіки Маркова з предикатними змінними, шляхом створення цих формул з предикатами з цільового домену.

2.4 Навчання трансдуктивного трансферу

Термін трансдуктивне навчання передачі вперше був запропонований Ендрю О. Арнольдом, Рамешом Наллапаті, Вільямом В. Коеном[70], де вони вимагали, щоб джерело та цільові завдання були однаковиими, хоча домени можуть бути різними. На додаток до цих умов, вони також вимагали, щоб усі немарковані дані у цільовому домені були доступні під час навчання, але цю умову можна послабити; натомість, у нашому визначенні налаштування трансдуктивного трансферу, ми вимагаємо лише, щоб частина немічених цільових даних бачилась під час навчання, щоб отримати граничну ймовірність цільових даних.

Слід звернути увагу, що слово „трансдуктивний” вживається з декількома значеннями. У традиційній обстановці машинного навчання трансдуктивне навчання [71] відноситься до ситуації, коли всі дані тесту потрібно бачити під час навчання, і що вивчену модель не можна використовувати повторно для майбутніх даних. Таким чином, коли надходять нові тестові дані, вони повинні бути класифіковані разом із усіма існуючими даними. Натомість у нашій категоризації трансферного навчання ми використовуємо термін трансдуктивний, щоб підкреслити концепцію, що в цьому типі трансферного навчання завдання повинні бути однаковиими, і в цільовій області повинні бути наявні немарковані дані.

За умови використання вихідного домену D_S та відповідного навчального завдання T_S , цільового домену D_T та відповідного навчального завдання T_T , трансдуктивне трансферне навчання має на меті покращити засвоєння цільової

функції прогнозування $fT(\bullet)$ у D_T з використанням знання в D_S і T_S , де $D_S \neq D_T$ і $T_S \neq T_T$. Крім того, на час навчання повинні бути доступні деякі не марковані дані цільового домену.

Це визначення охоплює роботу Ендрю О. Арнольда, Рамеша Наллапаті, Вільяма В. Коена [71], оскільки вони розглядали адаптацію доменів, де різниця полягає у граничному розподілі ймовірностей вихідних та цільових даних; тобто завдання однакові, але домени різні. Подібно до традиційної установки трансдуктивного навчання, яка має на меті якнайкраще використовувати немарковані тестові дані для навчання, у нашій класифікаційній схемі за трансдуктивним передаванням, ми припускаємо, що буде надано деякі немічені дані цільового домену.

У наведеному вище визначенні трансдуктивного навчання передачі вихідне та цільове завдання однакові, що означає, що можна адаптувати передбачувальну функцію, засвоєну у вихідному домені, для використання в цільовому домені за допомогою деяких немаркованих даних цільового домену. Як згадувалося в вище, цей параметр можна розділити на два випадки:

- Простори об'єктів між вихідним та цільовим доменами різні, $X_S \neq X_T$,
- простори об'єктів між доменами однакові, $X_S = X_T$, але граничні розподіли ймовірностей вхідних даних різні, $P(X_S) \neq P(X_T)$. Це схоже на вимоги щодо адаптації домену та упередженості вибору вибірки. Більшість підходів, описаних у наступних розділах, стосуються цього випадку.

Більшість підходів до передачі функціональних репрезентацій до навчальної програми трансдуктивного трансферу перебувають під невідконтрольною системою навчання. Бліцер та ін. [52] запропонував алгоритм структурного навчання кореспонденції (SCL), який розширюється [51], щоб використовувати нерозмічені дані з цільового домену для вилучення деяких благодатних особливостей, які можуть зменшити різницю між

доменами. Перший крок алгоритму полягає у визначенні набору опорних елементів для немаркованих даних з обох доменів. Потім SCL видаляє ці основні елементи з даних і розглядає кожну основну функцію як новий вектор мітки.

Припускаючи, що кожна проблема може бути вирішена за допомогою лінійного класифікатора, який показано наступним чином:

$$f_l(x) = \text{sgn}(w_l^T \cdot x), \quad l = 1, \dots, m \quad (2.7)$$

SCL може навчити матрицю $W = [w_1 w_2 \dots w_m]$ параметрів. На третьому етапі до матриці $W = [w_1 w_2 \dots w_m]$ застосовується декомпозиція особливого значення (SVD).

Нехай $W = UDV^T$, тоді $\theta = U_{T[h, :]}$, де h - кількість спільних функцій. Стандартні дискримінаційні алгоритми можна застосувати до розширеного вектора ознак для побудови моделей. Доповнений векторний об'єкт містить усі вихідні об'єкти x_i , додані до нових спільних об'єктів θ_{x_i} . Як згадувалося в [52], якщо опорні елементи добре розроблені, тоді вивчене відображення θ кодує відповідність між ознаками з різних доменів.

Хоча Бен-Девід і Шуллер [72] експериментально показали, що SCL може зменшити різницю між доменами, як вибрати основні функції важко і залежить від домену. У роботі [52] використано евристичний метод для вибору основних функцій для обробки природних мов (NLP), такий як тегування речень. У своїй подальшій роботі дослідники запропонували використовувати взаємну інформацію (MI) для вибору основних ознак замість використання більш евристичних критеріїв [22]. MI-SCL намагається знайти основні функції, які сильно залежать від міток у вихідному домені.

Навчання передачі в домені NLP іноді називають адаптацією домену. У цій області Дауме [53] запропонував функцію відображення ядра для проблем NLP, яка відображає дані як з вихідного, так і цільового доменів у просторовий простір ознак, де для навчання класифікаторів використовуються стандартні дискримінаційні методи навчання.

Однак побудована функція відображення ядра керується знаннями домену. Непросто узагальнити відображення ядра для інших областей або програм. Бліцер та ін. [73] проаналізував єдині межі збіжності для алгоритмів, які мінімізували опуклу комбінацію емпіричних ризиків джерела та цілі. У роботі 36 автори запропонували алгоритм, заснований на кластеризації для розповсюдження інформації мітки в різних доменах. У роботі [74] Дікан Сін запропонував новий алгоритм, відомий як мостове вдосконалення, щоб виправити мітки, передбачені класифікатором *shiftnaware*, до цільового розподілу та прийняти сумісний розподіл навчальних та тестових даних як міст для кращого переходу від навчальних даних до даних тесту. У роботі [75] автор запропонував структуру спектральної класифікації для проблеми навчання міждоменного передавання, де введена цільова функція для пошуку узгодженості між наглядом за індоменом внутрішньої структурою поза доменом.

Гуй-Ронг Сюе, Веньюань Дай, Цян Ян, Йонг Ю, запропонували міждоменний алгоритм класифікації тексту[76], який розширив традиційний імовірнісний алгоритм прихованого семантичного аналізу (PLSA) для інтеграції мічених та немічені дані з різних, але споріднених областей, в єдину імовірнісну модель. Нова модель називається *Topicbridged PLSA*, або *TPLSA*. Пан та співавт. Нещодавно запропонували трансферне навчання за допомогою зменшення розмірності[77]. У цій роботі Пан та ін. використав метод максимального середнього невідповідності (MMDE), спочатку розроблений для зменшення розмірності, щоб вивчити низький розмірний простір, щоб зменшити різницю розподілів між різними доменами для навчання трансдуктивного передавання. Однак MMDE може постраждати від свого обчислювального тягара. Так, у роботі [78] автор запропонував ефективний алгоритм вилучення ознак, відомий як аналіз компонентних передач (TCA), щоб подолати недолік MMDE.

2.5 Неконтрольованна передача навчання

Отримавши D_S вихідного домену із T_S навчального завдання, D_T цільового домену та відповідним T_T навчального завдання, безконтрольне навчання передачі має на меті покращити засвоєння цільової передбачувальної функції $f_T(\bullet)$ у D_T використовуючи знання з D_S та T_S , де $T_S = T_T$, а Y_S та Y_T не спостерігаються. Виходячи з визначення некерованого навчального режиму передачі, під час навчання в вихідних та цільових областях не спостерігаються розмічені дані. Поки що дослідницьких робіт з цього питання мало. Нещодавно запропоновано алгоритми кластеризації самоучок (STC) [40] та дискримінаційного аналізу переданих даних (TDA) [41] для передачі кластеризації та передачі проблем зменшення розмірності відповідно.

Згодом під час вивчення нових випадків проблем кластеризації, було запропоновано новий алгоритм, відомий як самоучка [40]. Кластеризація, що навчається самостійно, - це приклад безконтрольного навчання передачі, яке спрямоване на кластеризацію невеликої колекції немаркованих даних у цільовому домені за допомогою великої кількості немаркованих даних у вихідному домені. STC намагається вивчити загальний простір функцій між доменами, що допомагає кластеризації в цільовому домені. Цільова функція STC показана наступним чином:

$$J(\tilde{X}_T, \tilde{X}_S, \tilde{Z}) = I(X_T, Z) - I(\tilde{X}_T, \tilde{Z}) + \lambda [I(X_S, Z) - I(\tilde{X}_S, \tilde{Z})] \quad (2.8)$$

де X_S та X_T - дані джерела та цільового домену відповідно. Z - це спільний простір функцій X_S та X_T , а $I(\bullet, \bullet)$ - взаємна інформація між двома випадковими величинами.

Припустимо, що існують три функції кластеризації $C_{X_T} : X_T \rightarrow \tilde{X}_T$, $C_{X_S} : X_S \rightarrow \tilde{X}_S$ and $C_Z : Z \rightarrow \tilde{Z}$ - це відповідні кластери X_T , X_S і Z відповідно. Метою STC є вивчення \tilde{X}_T , вирішуючи задачу оптимізації

$$\arg \min_{\tilde{X}_T, \tilde{X}_S, \tilde{Z}} J(\tilde{X}_T, \tilde{X}_S, \tilde{Z}) \quad (2.9)$$

Ітераційний алгоритм вирішення функції оптимізації наведено в [40]. Подібним чином [41] було запропоновано алгоритм переданого дискримінаційного аналізу (TDA) для вирішення проблеми зменшення розмірності передачі. TDA спочатку застосовує методи кластеризації для створення міток псевдокласу для цільових немічених даних. Потім він застосовує методи зменшення розмірності до цільових даних та позначених вихідних даних, щоб зменшити розміри. Ці два кроки виконуються ітеративно, щоб знайти найкращий підпростір для цільових даних.

2.6 Обмеження трансферу і негативний трансфер

Важливим завданням є визначення межі сили трансферного навчання. У роботі [79] Хасан Махмуд та Рей проаналізували випадок навчання перекладу, використовуючи складність Колмогорова, де доведено деякі теоретичні межі. Зокрема, автори використовували умовну складність Колмогорова для вимірювання взаємозв'язку між завданнями та передачі "потрібного" обсягу інформації в процесі послідовного перенесення навчального завдання під байєсівські рамки.

Нещодавно Ітон та ін. [80] запропонував новий метод передачі знань, заснований на графіках, де зв'язок між завданнями джерела моделюється шляхом вбудовування набору вивчених моделей джерел у графік із використанням передаваності як метрики. Перехід до нового завдання відбувається шляхом відображення задачі на графіку, а потім вивчення функції на цьому графіку, яка автоматично визначає параметри для перенесення до нового навчального завдання. Негативне перенесення відбувається, коли вихідні дані домену та завдання сприяють зниженню продуктивності навчання в цільовому домені.

Незважаючи на те, що питання як уникнути негативного перенесення є дуже важливим, поки, існує мало дослідницьких робіт з цієї теми. Розенштейн та ін. [81] емпірично показав, що якщо два завдання занадто неподібні, то передача грубої сили може зашкодити виконанню цільового завдання. Деякі

роботи були використані для аналізу взаємозв'язку завдань та методів кластеризації завдань, таких як [82], [83], які можуть допомогти надати вказівки щодо того, як автоматично уникнути негативного перенесення. Баккер і Хескес [83] застосували байєсівський підхід, згідно з яким деякі параметри моделі спільно використовуються для всіх завдань, а інші більш вільно пов'язані за допомогою спільного попереднього розподілу, про який можна дізнатися з даних. Таким чином, дані кластеризовані на основі параметрів завдання, де завдання в одному кластері повинні бути пов'язані між собою. Argürou та ін. [84] розглядаються ситуації, в яких навчальні завдання можна розділити на групи. Завдання в кожній групі пов'язані спільним використанням низьковимірною подання, яке відрізняється між різними групами. Як результат, завданням у межах групи може бути простіше передати корисні знання.

2.7 Застосування трансферного навчання

Останнім часом методики трансферного навчання успішно застосовуються в багатьох реальних додатках. Наприклад, було [85] [50], [42] запропоновано використовувати методи навчання передачі для вивчення текстових даних у різних доменах, відповідно, та використовувати SCL для вирішення проблем НЛП. Розширення SCL було запропоновано для вирішення проблем класифікації настроїв.

Згодом було [67] запропоновано використовувати як неадекватні дані цільового домену, так і велику кількість вихідних даних низької якості для вирішення проблем класифікації зображень.

В процесі дослідження було [72] запропоновано використовувати трансдуктивні методи навчання передачі для вирішення проблем розпізнавання іменних сутностей. У роботах [86], [87], [88], [89], [90] пропонуються технології навчання передачі для вилучення знань з моделей локалізації WiFi через періоди часу, простору та мобільних пристроїв, щоб отримати користь від завдань локалізації WiFi в інших налаштуваннях. Чжуо

та ін. [91] вивчав, як передавати знання про домен для вивчення реляційних моделей дій між доменами в автоматизованому плануванні.

У роботі [92] автор запропонував новий байєсівський алгоритм багатозначного навчання, який може автоматично ідентифікувати відповідну підмножину ознак та використовувати індуктивну передачу для вивчення декількох, концептуально пов'язаних класифікаторів, для автоматизованого проектування (САПР).

У роботі [93] автор запропонував інформаційно-теоретичний підхід до трансферного навчання для вирішення проблеми міжмовної класифікації для перекладу веб-сторінок з англійської на китайську. Підхід вирішив проблему, коли існує безліч позначених англійською мовою текстових даних, тоді як існує лише невелика кількість позначених китайськими текстовими документами.

Навчання передачі через два простори функцій досягається шляхом створення відповідної функції відображення як мосту. Поки що є щонайменше два міжнародні конкурси, засновані на навчанні за допомогою трансферу, які надали деякі вкрай необхідні публічні дані.

Розглянемо приклад виконання завдання персоналізованої фільтрації та узагальнення спаму для відповідних навчальних завдань. Для навчання системі фільтрації спаму нам потрібно зібрати багато електронних листів від групи користувачів із відповідними мітками: спам чи не спам та підготувати класифікатор на основі цих даних. Для нового користувача електронної пошти ми можемо захотіти адаптувати вивчену модель для користувача. Проблема полягає в тому, що розподіл електронних листів для першого набору користувачів та нового користувача відрізняється. Таким чином, цю проблему можна моделювати як індуктивну проблему навчання перекладу, метою якої є адаптація старої моделі фільтрації спаму до нової ситуації із меншою кількістю навчальних даних та меншим часом навчання.

Іншим прикладом є завдання оцінки розташування WiFi у приміщенні, використовуючи дані сигналу WiFi, отримані протягом різних періодів

часу[94]. Оскільки значення потужності сигналу WiFi можуть бути функцією часу, простору та пристроїв, розподіл даних WiFi за різні періоди часу може сильно відрізнятись.

Отже, навчання передачі даних має бути розроблене таким чином, щоб зменшити зусилля щодо повторного маркування даних. До цього часу було опубліковано кілька наборів даних для досліджень трансферного навчання. Ми позначаємо набори даних для видобутку тексту, набір даних про фільтрування спаму електронною поштою, набір даних про локалізацію WiFi за періодами часу та дані класифікації настрою відповідно, Text, Email, WiFi та Sen. Текст Деякі дослідники попередньо обробили три набори даних, 20 груп новин, SRAA та Reuters21578. Дані в цих наборах даних класифікуються за ієрархічною структурою. Дані з різних підкатегорій за однією батьківською категорією вважаються з різних, але пов'язаних доменів.

Завдання - передбачити мітки батьківської категорії.

- Електронна пошта Цей набір даних наданий 2006 ECML / PKDD-виявленням.
- WiFi. Цей набір даних наданий Конкурсом ICDM-2007 10. Дані були зібрані всередині будівлі для локалізації близько $145,5 \times 37,5$ м² протягом двох різних періодів часу.
- Товари. Цей набір даних містить відгуки про товари, завантажені з Amazon.com із 4 типів товарів (доменів): Кухня, Книги, DVD та Електроніка. Кожен домен має кілька тисяч відгуків, але точна кількість залежить від домену. Відгуки містять оцінки зірок (від 1 до 5 зірок).

Для того, щоб показати, наскільки корисними можуть бути методи передачі навчання порівняно з традиційними методами навчання, дослідники використали деякі загальнодоступні масиви даних. У таблиці 5 надано список, взятий з деяких опублікованих статей щодо трансферного навчання. У [20], [95], [53] автори використовували дані 20 груп новин. Через різницю в етапах попередньої обробки алгоритмів різними дослідниками важко порівняти

запропоновані методи безпосередньо. Таким чином, ми позначимо їх відповідно 20-Newsgroup1, 20-Newsgroups2 і 20-Newsgroups3, а результати таблиці покажемо у порівнянні між запропонованими методами навчання з трансферу та методами навчання без передачі.

За даними 20 груп новин[20], було проведено експерименти порівняння між стандартною машиною Vector Vector Machine (SVM) та запропонованим алгоритмом TrAdaBoost. На 20 групах новин, Ши та інші[95] застосували активний алгоритм навчання для вибору важливих екземплярів для трансферного навчання (AcTraK) за допомогою TrAdaBoost та стандартного SVM. Гао та ін. [63] оцінили запропоновані ними локально зважені алгоритми навчання ансамблю, pLWE та LWE, на 20 групах новин, порівняно з SVM та Logistic Regression (LR). Крім того, у таблиці показано результати порівняння набору даних класифікації настроїв, про який повідомляється в [22]. У цьому наборі даних SGD позначає алгоритм стохастичного градієнт-спуску з втратою Губера, SCL являє собою лінійний провісник нових уявлень, вивчених алгоритмом структурного кореспондентського навчання, а SCL-MI є розширенням SCL, застосовуючи Взаємну інформацію для вибору опори особливості алгоритму SCL. Нарешті, на наборі даних локалізації WiFi, ми показуємо результати порівняння, про які повідомляється в [78], де базовою лінією є регуляризована модель найменшої квадратної регресії (RLSR), яка є стандартною моделлю регресії, і KPCA, що представляє застосування RLSR на нові подання даних, засвоєних за допомогою аналізу компонентних принципів ядра. Порівняні методи навчання передачі включають відповідність середнього ядра (КММ) та запропонований алгоритм, аналіз компонентів передачі (ТСА).

З цих результатів порівняння(див. табл.2.16.) ми можемо виявити, що методи навчання трансферу, розроблені належним чином для реальних застосувань, дійсно можуть значно покращити ефективність порівняно з методами навчання без трансферу.

Трансферне навчання знайшло багато застосувань і в послідовному машинному навчанні. Наприклад, [96] було запропоновано графічний метод ідентифікації ігор, що раніше зустрічались, і застосував цей прийом для автоматизації картографування доменів для передачі функції значення та пришвидшення навчання підкріплення на варіантах раніше зіграних ігор.

У такому навчанні пропонується новий підхід до передачі між абсолютно різними просторами об'єктів, що стає можливим завдяки вивченню функції зіставлення для об'єднання об'єктів у двох цілком різних сферах (зображення та текст) [97].

Нарешті, дослідниками [98], [99] було застосовано трансферне навчання до спільних проблем фільтрації для вирішення проблем холодного старту та розрідженості. Також у роботі [98] вченим було вивчено спільну модель суміші рейтингових моделей, відому як генеративна модель рейтингу-матриці (RMGM), з точки зору прихованих змінних користувача та кластера елементів. RMGM з'єднує кілька матриць рейтингу з різних доменів, відображаючи користувачів та елементи в кожній матриці рейтингу на спільних прихованих просторах користувачів та елементів для передачі корисних знань

Таблиця 2.5. Порівняння між методами навчання за допомогою трансферу та методами навчання без трансферу

Data Set (reference)	Source v.s. Target	Baselines		TL Methods	
20 Newsgroups ₁ ([6]) ACC (unit: %)		SVM		TrAdaBoost	
	rec v.s. talk	87.3%		92.0%	
	rec v.s. sci	83.6%		90.3%	
	sci v.s. talk	82.3%		87.5%	
20 Newsgroups ₂ ([84]) ACC (unit: %)		SVM		TrAdaBoost	AcTraK
	rec v.s. talk	60.2%		72.3%	75.4%
	rec v.s. sci	59.1%		67.4%	70.6%
	comp v.s. talk	53.6%		74.4%	80.9%
	comp v.s. sci	52.7%		57.3%	78.0%
	comp v.s. rec	49.1%		77.2%	82.1%
	sci v.s. talk	57.6%		71.3%	75.1%
20 Newsgroups ₃ ([49]) ACC (unit: %)		SVM	LR	pLWE	LWE
	comp v.s. sci	71.18%	73.49%	78.72%	97.44%
	rec v.s. talk	68.24%	72.17%	72.17%	99.23%
	rec v.s. sci	78.16%	78.85%	88.45%	98.23%
	sci v.s. talk	75.77%	79.04%	83.30%	96.92%
	comp v.s. rec	81.56%	83.34%	91.93%	98.16%
	comp v.s. talk	93.89%	91.76%	96.64%	98.90%
Sentiment Classification ([8]) ACC (unit: %)		SGD		SCL	SCL-MI
	DVD v.s. book	72.8%		76.8%	79.7%
	electronics v.s. book	70.7%		75.4%	75.4%
	kitchen v.s. book	70.9%		66.1%	68.6%
	book v.s. DVD	77.2%		74.0%	75.8%
	electronics v.s. DVD	70.6%		74.3%	76.2%
	kitchen v.s. DVD	72.7%		75.4%	76.9%
	book v.s. electronics	70.8%		77.5%	75.9%
	DVD v.s. electronics	73.0%		74.1%	74.1%
	kitchen v.s. electronics	82.7%		83.7%	86.8%
	book v.s. kitchen	74.5%		78.7%	78.9%
	DVD v.s. kitchen	74.0%		79.4%	81.4%
	electronics v.s. kitchen	84.0%		84.4%	85.9%
WiFi Localization ([67]) AED (unit: meter)		RLSR	PCA	KMM	TCA
	Time A v.s. Time B	6.52	3.16	5.51	2.37

У роботі[99] вони застосували алгоритми кластеризації до користувачів та елементів у допоміжній рейтинговій матриці. Потім побудували рейтингову матрицю кластерного рівня, відому як кодова книга. Припускаючи, що цільова рейтингова матриця (для фільмів) пов'язана з допоміжною (для книг), цільовий домен можна реконструювати, розширивши кодову книгу, завершивши процес передачі знань.

ВИСНОВКИ ДО РОЗДІЛУ

У даному розділі розглянуто кілька сучасних тенденцій навчання трансферу. Трансферне навчання класифікується за трьома різними параметрами: індуктивне трансферне навчання, трансдуктивне трансферне навчання та безконтрольне трансферне навчання.

Безконтрольне навчання з трансферу може привертати все більше уваги в майбутньому. Крім того, кожен із підходів до трансферного навчання може бути класифікований на чотири контексти, засновані на "що перенести" у навчанні. Вони включають підхід передачі екземпляра, підхід передачі ознак, підхід передачі параметрів та підхід передачі реляційних знань, відповідно.

Перші три контексти мають припущення щодо даних, тоді як останній контекст стосується навчання передачі реляційних даних. Більшість з підходів передбачають, що вибраний вихідний домен пов'язаний із цільовим доменом. Надалі потрібно вирішити кілька важливих дослідницьких питань. По-перше, як уникнути негативного переказу - це відкрита проблема. Оскільки, багато запропонованих алгоритмів навчання передачі припускають, що вихідний та цільовий домени певним чином пов'язані між собою. Однак, якщо припущення не виконується, може статися негативний переказ навчання, що може призвести до того, що учень працює гірше, ніж до переносу навчання.

Щоб уникнути негативної передачі навчання, слід спочатку вивчити можливість перенесення між вихідними доменами чи завданнями та цільовими доменами чи завданнями. На основі відповідних заходів перенесення, ми можемо вибрати відповідні доменні джерела або завдання, з

яких витягуватимемо знання для вивчення цільових завдань. Щоб визначити передачу між доменами та завданнями, нам потрібно визначити критерії для вимірювання подібності між доменами чи завданнями.

Наприклад, на основі вимірювань відстані ми можемо кластеризувати домени або завдання, що може допомогти виміряти передачу. Проблема полягає в тому, що коли весь домен не може бути використаний для навчання передачі, чи можемо ми все ще передавати частину домену для корисного навчання в цільовому домені. Крім того, більшість існуючих на даний час алгоритмів навчання передачі були зосереджені на вдосконаленні узагальнення в різних розподілах між вихідними та цільовими доменами або завданнями. Роблячи це, вони припустили, що простори об'єктів між вихідним та цільовим доменами однакові. Однак у багатьох додатках ми можемо захотіти передавати знання між доменами або завданнями, що мають різні простори функцій, і переносити з кількох таких вихідних доменів.

Цей тип трансфертного навчання називається гетерогенним трансфертним навчанням. Нарешті, дотепер методи навчання передачі застосовувались переважно до невеликих програм із обмеженим різноманіттям, таких як локалізація на основі сенсорної мережі, проблеми класифікації тексту та класифікації зображень. У майбутньому методики трансферного навчання будуть широко застосовуватися для вирішення інших складних програм, таких як класифікація відео, аналіз соціальних мереж та логічний висновок.

3 СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ RESNET

3.1 Огляд існуючих архітектурних рішень згорткових нейронних мереж для створення інтелектуальної медичної системи

Для створення інтелектуальної медичної системи, було вирішено обрати згорткову нейронну мережу, оскільки її архітектура призначена і використовується для ефективного розпізнавання зображень, де чергуються згорткові шари з нелінійними функціями активації і шари об'єднання або підвибірки.

Згорткові нейронні мережі перевершили людський рівень класифікації зображень в 2015 році. Ці мережі витягають низько-, середньо- і високорівневі ознаки наскрізним багат шаровим способом, а збільшення кількості stacked layers може збагатити «рівні» ознак. Stacked layer має вирішальне значення.

Коли глибша мережа починає згортатися, виникає проблема: зі збільшенням глибини мережі точність спочатку збільшується, а потім швидко погіршується. Зниження точності навчання показує, що не всі мережі легко оптимізувати.

Щоб подолати цю проблему, Microsoft ввела глибоку «залишкову» структуру навчання [106]. Замість того, щоб сподіватися на те, що кожні кілька stacked layers безпосередньо відповідають бажаному основному поданням, вони явно дозволяють цим верствам відповідати «залишковим». Формулювання $F(x) + x$ може бути реалізована за допомогою нейронних мереж з сполуками для швидкого доступу.

З'єднання швидкого доступу (shortcut connections) пропускають один або кілька шарів і виконують зіставлення ідентифікаторів. Їх виходи додаються до виходів stacked layers.

КАФЕДРА АКІК

НАУ 20 19 32 000 ПЗ

<i>Розроб.</i>	<i>Хоцянівський В.П.</i>			<i>Інтелектуальна медична система на основі нейронних мереж RESNET</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Керівник</i>	<i>Синєглазов В.М.</i>						
<i>Консульт.</i>							
<i>Н.контр.</i>	<i>Тупіцин М. Ф.</i>						
<i>Зав.кафедри</i>	<i>Синєглазов В.М.</i>						
					<i>151 204М</i>		

Тому було розглянуто та проаналізовано наступні архітектури загорткових нейронних мереж:

- AlexNet
- ZF Net
- VGGNet
- Inception
- U-net
- ResNet

Для порівняння розглянемо точність архітектур нейронних мереж за результатами точності роботи на тестовому наборі даних ImageNet (див.рис.3.1).

Також для оцінки моделей згорткових нейронних мереж крім виду помилки зазвичай вказують кількість моделей в ансамблі і кількість вирізів зображення, які подавалися на вхід кожної моделі.

Наприклад, 10 вирізів означає, що зроблено чотири вирізи по кутах зображення, один виріз в центрі, і кожен виріз додатково горизонтально перевернули.

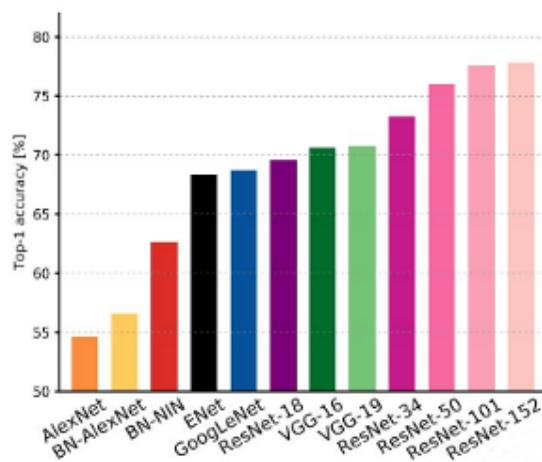


Рис.3.1 – кінцева точність архітектури нейронної на тестовому наборі даних ImageNet

У таблиці 3.2 представлені результати розглянутих нейронних мереж з однією моделлю і з одним вирізом на базі зображень ImageNet (крім ResNet-151, для якої вказано результат для 10 вирізів).

Таблиця 3.2. Результати розглянутих нейронних мереж з однією моделлю і з одним вирізом на базі зображень ImageNet

Нейрона мережа	Топ -1	ТОП-5
AlexNet	39,00%	17,00%
ZF Net	37,50%	16,00%
VGGNet	25,60%	8,10%
Inception	21,20%	9,20%
U-net	20,00%	5,60%
ResNet101	19,90%	4,90%
ResNet151	19,38%	4,49%

У таблиці 3.3 представлені результати використання ансамблів моделей з багатьма вирізами на базі зображень ImageNet. Як видно з даних таблиць, за п'ять років, з 2012 по 2016 роки, топ-5 помилка на ImageNet для одиночних моделей зменшилася майже в чотири рази (з 17% до 4,49%), а для ансамблю - майже в п'ять разів (з 15,40% до 3,10%)

Таблиця 3.3. Результати використання ансамблів моделей з багатьма вирізами на базі зображень ImageNet

Нейронна мережа	Моделі	Вирізи	Топ -1	ТОП-5
AlexNet	7	1	36,70%	15,31%
ZF Net	6	10	36,00%	14,70%
VGGNet	2	150	23,70%	6,80%
Inceptionv3	7	144	17,20%	6,67%
U-net	4	144	-	3,58%
ResNet151	4	144	16,50%	3,10%

Тому зважаючи на данні вище в подальшому для розробки інтелектуальної медичної системи доцільно використовувати архітектуру ResNet.

Оскільки ResNet може мати різні розміри, в залежності від розміру кожного з шарів моделі і кількості шарів, ми будемо слідувати описаному авторами в статті [123] - ResNet34 - щоб пояснити структуру цих мереж.

На рис.3.4 бачимо, що ResNet складається з одного етапу згортки і об'єднання (оранжевого кольору), за яким слідують 4 шари схожої поведінки.

Кожен з цих шарів має однакову структуру. Вони виконують згортку 3×3 з фіксованою розмірністю карти характеристик $(F)[64, 128, 256, 512]$ відповідно, минаючи вхідні дані кожні 2 згортки. Крім того, розміри ширини (W) і висоти (H) залишаються незмінними протягом всього шару.

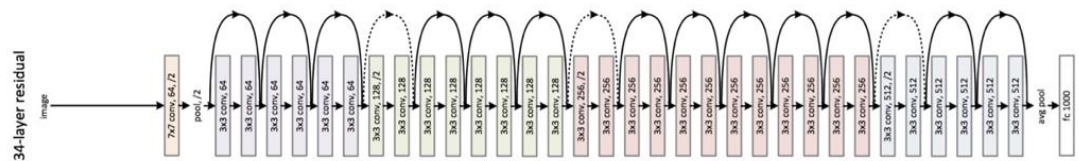


Рис.3.4. Архітектура ResNet34

Пунктирна лінія є саме тому, що відбулася зміна розміру вхідного обсягу (через процес згортки). Скорочення між шарами досягається за рахунок збільшення кроку з 1 до 2 при першій згортці кожного шару, замість операції об'єднання.

На рис.3.5 наведено зведення вихідного розміру на кожному шарі і розмірності згортальних ядер в кожній точці структури.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Рис.3.5. Розміри виходів і згортальних ядер для ResNet34

Розглянемо спрощену структуру ResNet на рис.3.6.

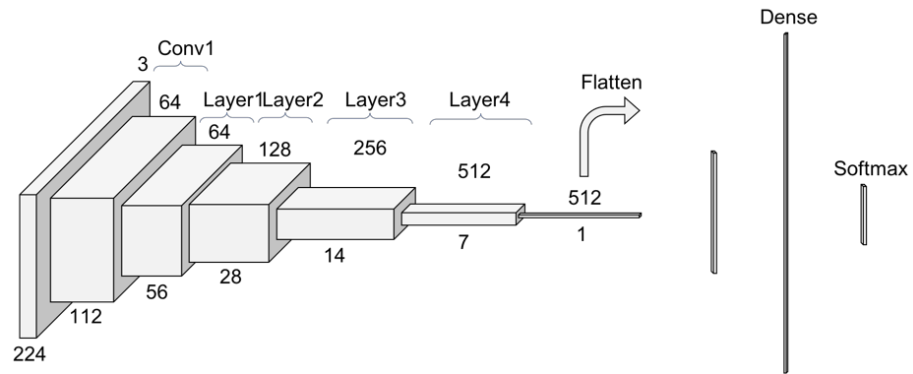


Рис.3.6. Архітектура ResNet34

Першим кроком в ResNet перед входом в поведінку загального рівня є блок, названий Conv1, що складається з згортки + пакетної нормалізації + операції максимального пулу.

Тобто, спочатку йде операція згортки. На рис.3.4 бачимо, що вони використовують розмір ядра 7 і розмір карти характеристик 64. Можесо зробити висновок, що вони доповнюються нулями 3 рази в кожному вимірі. Беручи це до уваги, на рис. 3.7 видно, що вихідний розмір цієї операції буде томом 112x112. Оскільки кожен згортковий фільтр (з 64) надає один канал в вихідному обсязі, ми отримуємо вихідний обсяг (112x112x64).

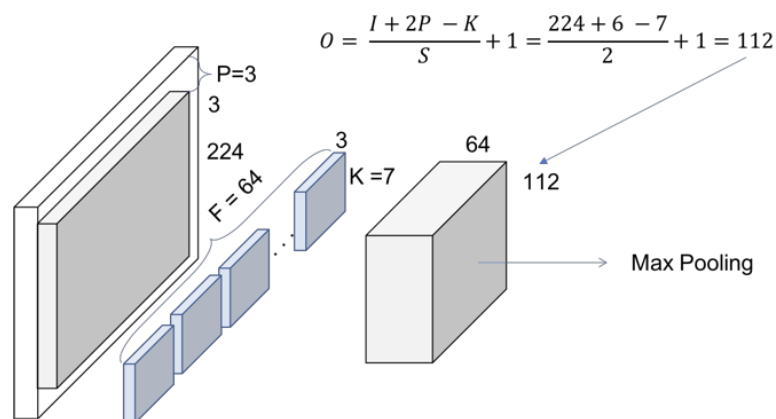


Рис.3.7. Архітектура ResNet34. Conv1 - згортка

Наступним кроком є нормалізація партії, яка поелементарною операцією, тому вона не змінює розмір тома. Нарешті, у нас є операція (3x3) Max Pooling

з кроком 2. Ми також можемо зробити висновок, що спочатку доповнюється вхідний обсяг, тому кінцевий обсяг має бажані розміри (див.рис.3.8).

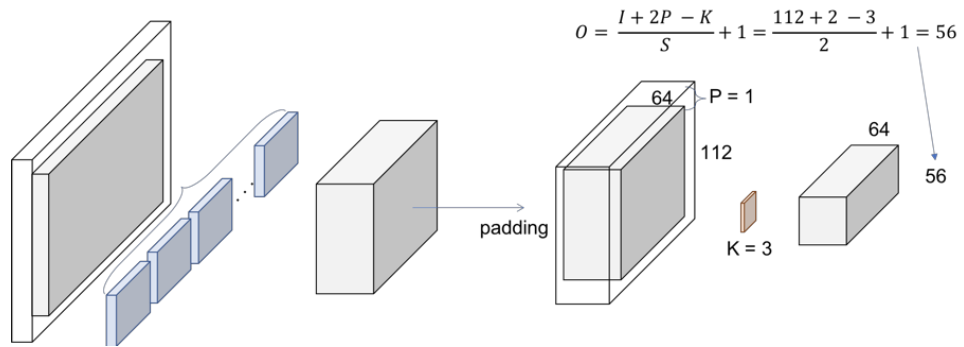


Рис.3.8. Архітектура ResNet34. Conv1 - максимальний пул

Отже, давайте пояснимо поняття - блок. Кожен шар ResNet складається з декількох блоків. Це пов'язано з тим, що коли ResNet-ти ідуть глибше, вони зазвичай збільшують кількість операцій в блоці, але загальна кількість рівнів залишається незмінним - 4. Операція тут відноситься до пакунку, нормалізації партії і активації ReLU на вході, крім останньої операції блоку, який не має ReLU.

Розглянемо кожну з операцій, що показанні на рис.3.4. на прикладі блока 3*3 conv, 64 показано на рис.3.9

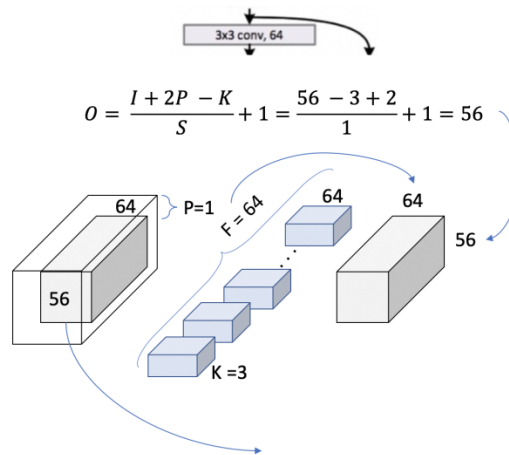


Рис.3.9 Рівень 1, блок 1, операція 1

Тепер ми можемо перевірити це двічі в таблиці 3.5, в якій ми використовуємо ядро [3x3, 64], а розмір виведення дорівнює [56x56]. Як бачимо розмір не змінюється всередині блоку. Це тому, що використовується

відступ 1 і крок 1. Також подивимося, як це поширюється на весь блок, щоб охопити $2 [3 \times 3, 64]$, який з'являється в таблиці.

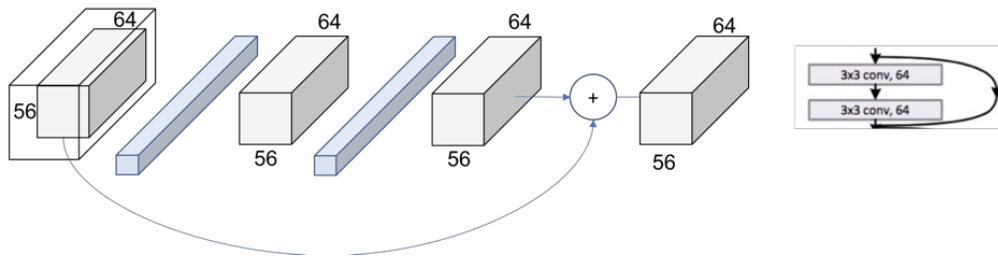


Рис.3.10 шар 1, блок 1

Ця ж процедура може бути розширена до всього шару, як показано на рисунку 3.11. Тепер ми можемо повністю прочитати всю комірку таблиці Памятаємо, що ми в ResNet34 на рівні Conv2_x. Бачимо, як у нас є $[3 \times 3, 64] \times 3$ рази в шарі,

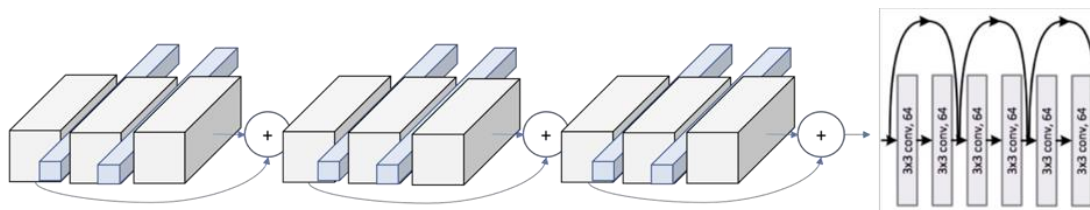


Рис.3.11 шар 1

Наступним кроком є ескалація від всього блоку до всього шару. На рисунку 3.4 ми бачимо, як шари розрізняються по кольорам. Однак, якщо ми подивимося на першу операцію кожного шару, ми побачимо, що крок, використаний в цьому першому, дорівнює 2, а не 1, як для інших.

Це означає, що зменшення обсягу тома, хоча мережа досягається за рахунок збільшення кроку замість операції об'єднання як зазвичай роблять CNN. Фактично, тільки одна операція максимального пулу виконується на нашому рівні Conv1 і один середній рівень пулу в кінці ResNet, прямо перед повністю підключеним щільним шаром на рисунку 3.4.

Також бачимо інший повторюваний патерн над шарами ResNet, точковий шар, що представляє зміну розмірності. Першою операцією кожного шару є зменшення розміру.

Ця різниця в з'єднаннях що пропускається називається ярлик проекції. Ярлик проекції виконує операцію згортки, щоб забезпечити однаковий розмір томів у цій операції додавання.

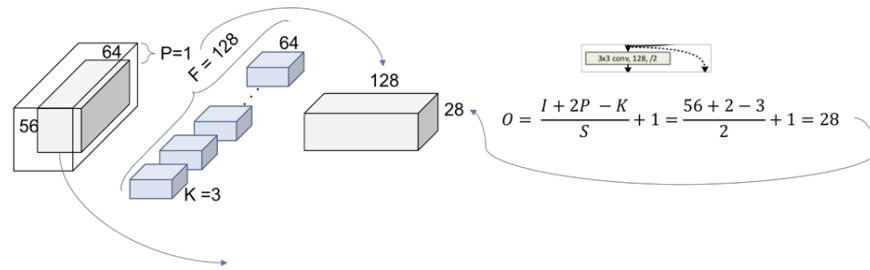


Рис.3.12 Рівень 2, Блок 1, операція 1

На рис.3.12 представлена дискретизація, виконана шляхом збільшення кроку до 2. Кількість фільтрів дублюється в спробі зберегти складність часу для кожної операції ($56 * 64 = 28 * 128$). Тепер операція додавання не може бути виконана, так як том змінився. В найкоротший шлях ми повинні застосувати одну з наших стратегій понижувальної вибірки. Метод згортки 1x1 показаний на рис.3.13.

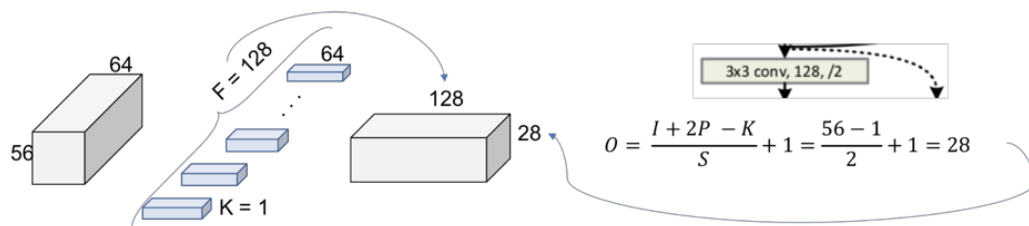


Рис.3.13. Ярлик проекції

Остаточне зображення виглядає так, як на рис.3.14, де тепер 2 вихідних томи кожного потоку мають однаковий розмір і можуть бути додані.

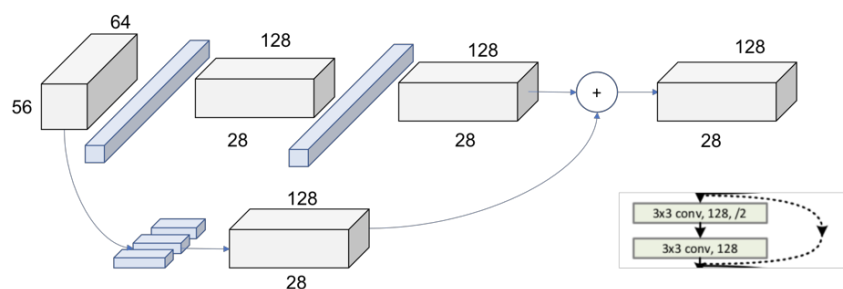


Рис.3.14. Шар 2, Блок 1

На малюнку 3.15 ми бачимо загальну картину всього другого шару. Поведінка точно таке ж для наступних шарів 3 і 4, змінюючи тільки розміри входять томів.

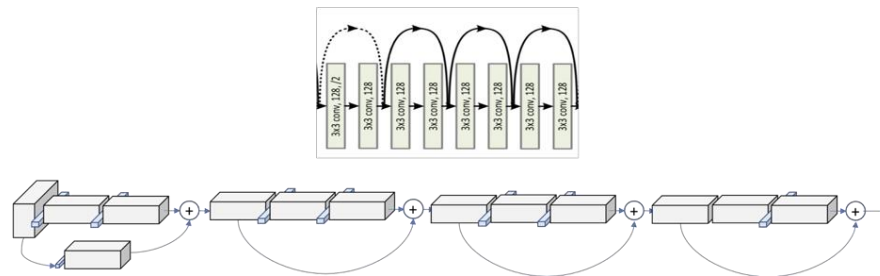


Рис.3.15. Шар 2, Блок 1

Тепер порівняємо графіки навчання мережі VGG та Resnet18 і 34 (див.рис.3.16 та 3.17). Обидві мережі навчались на наборі даних ImageNet. Тонкі криві позначають помилку навчання, а жирні криві позначають помилку валідації. Зліва: прості мережі з 18 і 34 шарів. Справа: ResNet з 18 і 34 шарів. На цьому графіку залишкові мережі не мають додаткових параметрів у порівнянні з їх звичайними аналогами.

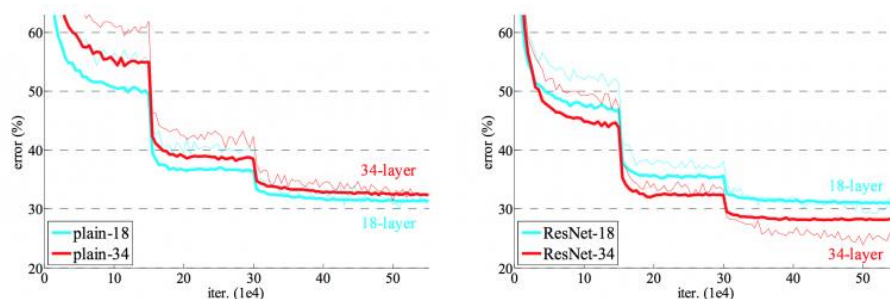


Рис.3.16. Порівняльний графік навчання мереж VGG та ResNet з 18 та 34 шарами

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	25.03

Рис.3.17 Узагальнена таблиця результатів навчання мереж VGG та ResNet з 18 та 34 шарами

Мережа ResNet сходиться швидше, ніж її простий аналог.

Використовуючи ResNet, можна вирішити безліч проблем, таких як:

- ResNet відносно легко оптимізувати: «прості» мережі (які просто складають шари) показують велику помилку навчання, коли глибина збільшується.
- ResNet дозволяє відносно легко збільшити точність завдяки збільшенню глибини, чого з іншими мережами домогтися складніше.

Рисунок 3.18 показує, що більш глибокі ResNet досягають кращих результатів навчання в порівнянні з неглибокою мережею.

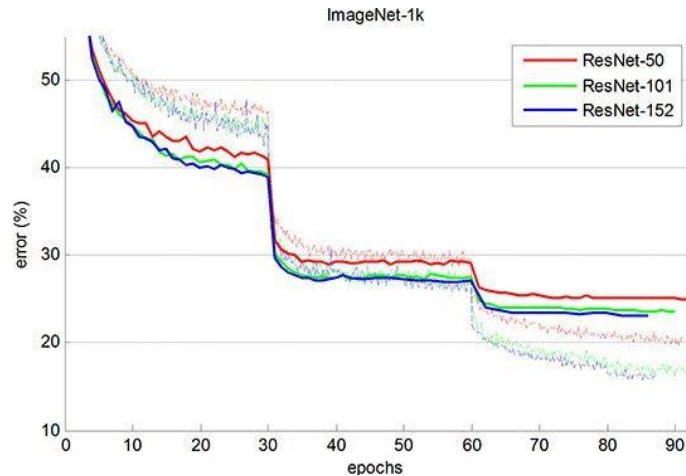


Рис.3.18 Помилка архітектур ResNet

3.2 Формування навчальної множини для задачі обробки зображень при навчанні згорткової нейронної мережі

У початковій вибірці [14] було розглянуто 305 зображень результатів УЗД для аналізу стадій фіброзу печінки. Кількість зображень результатів УЗД для кожної стадії фіброзу:

F0 – 78

F3 – 47

F1 – 38

F4 - 85

F2 – 67

Стадії фіброзу для кожного зображення встановили за допомогою біопсії. У навчальній вибірці знаходилось 229 картинок. У контрольній вибірці було 76 зображень.

Нейронна мережа навчалась на обчислювальному процесорі NVIDIA Tesla K80 з 12 ГБ виділеної відеопам'яті. Для імплементації нейромережі була використана мова програмування Python з використанням бібліотеки Keras (з бекендом TensorFlow) у якості високорівневої бібліотеки нейромереж).

У якості нейронної мережі було використано Resnet101[15].

Під час проведення дослідження, спочатку було навчено мережу на оригінальній навчальній вибірці протягом 50 епох (див.рис 3.19).

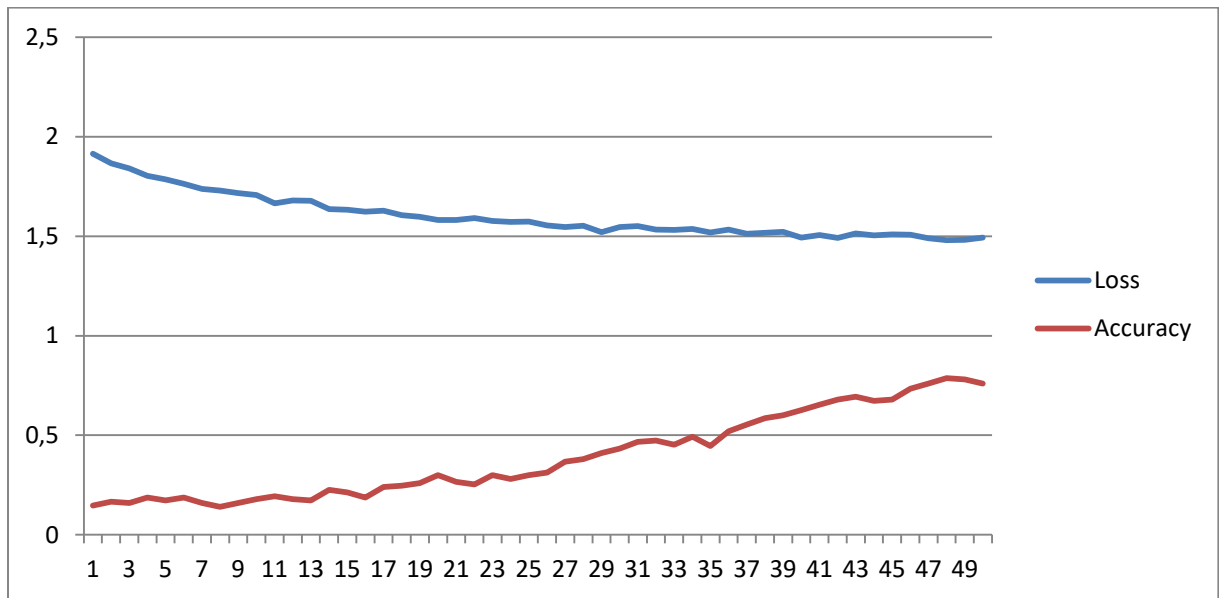


Рис 3.19. Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох на початковому наборі даних

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів(див.рис.3.20).



Рис 3.20. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 76.417 %.

Після навчання оригінального набору даних було перевірено ситуацію, коли один з класів навчання мережі є дефектним (недостатній розмір навчальної вибірки). Для перевірки було обрано клас F0 (кількість зображень

зменшено до 10). Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох (див.рис. 3.21).

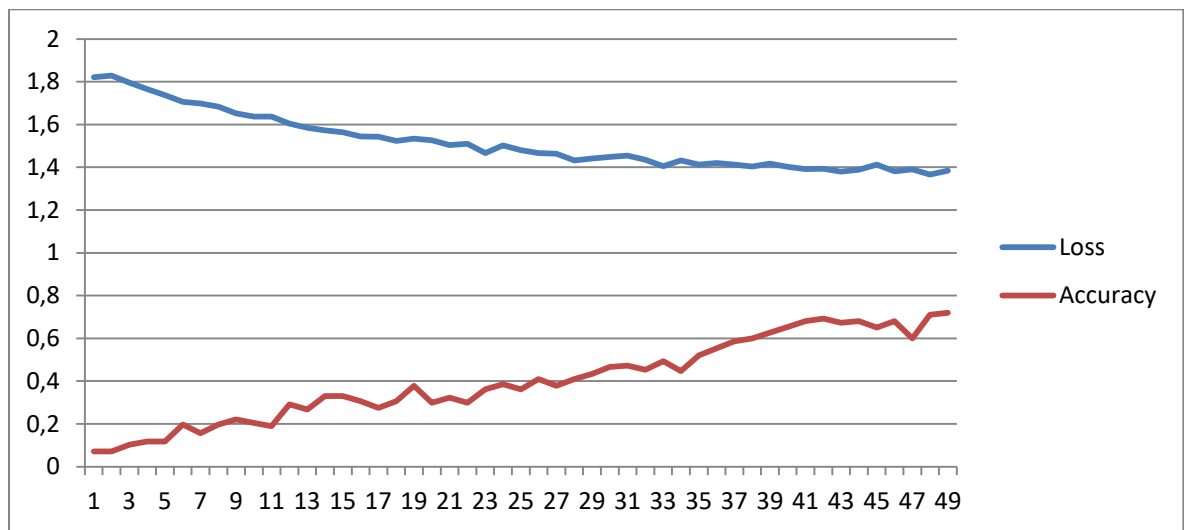


Рис 3.21. Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох на наборі даних з одним дефектним класом (F0)

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів (див.рис. 3.22).



Рис 3.22. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 70.940%

Після навчання і отримання результатів навчання мережі по дефектному класу F0 було повернуто початкову вибірку і в ній штучно створено

дефектний клас F1 (було залишено 10 картинок). Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох (див.рис.3.23).

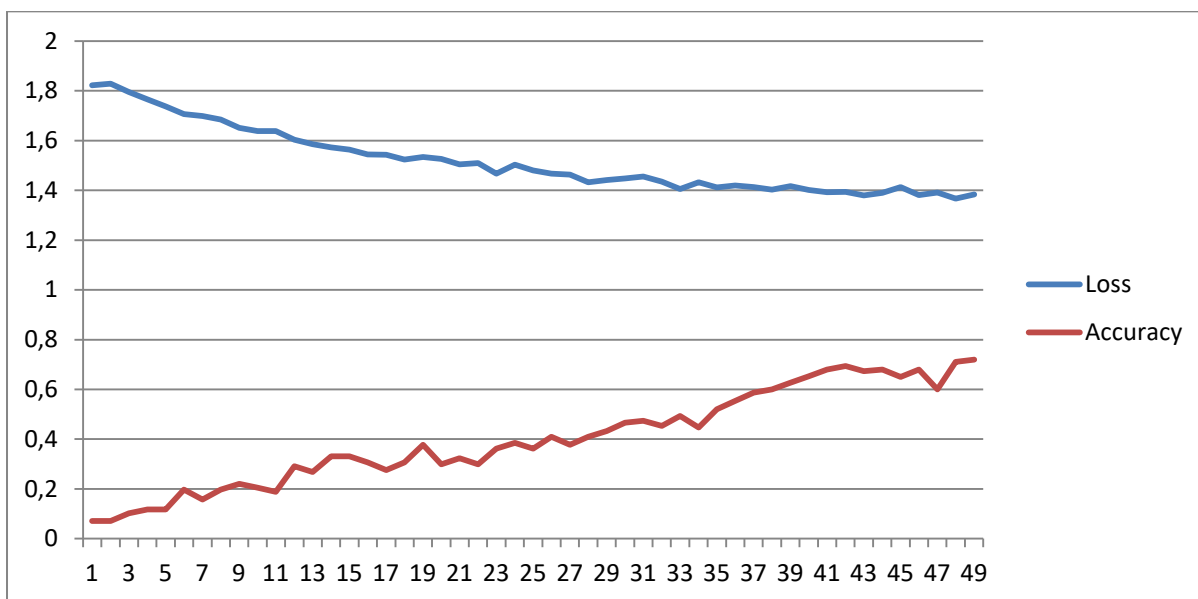


Рис 3.23. Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох на наборі даних з одним дефектним класом (F1)

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів (див.рис.3.24).



Рис 3.24. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 73.6%.

Після навчання і отримання результатів навчання мережі по дефектних класах F0 та F1 було повернуто початкову вибірку і в ній штучно створено

дефектний клас F2 (було залишено 10 картинок). Графік втрат та точності при навчанні мережі Resnet101 протягом 100 епох (див.рис.3.25).

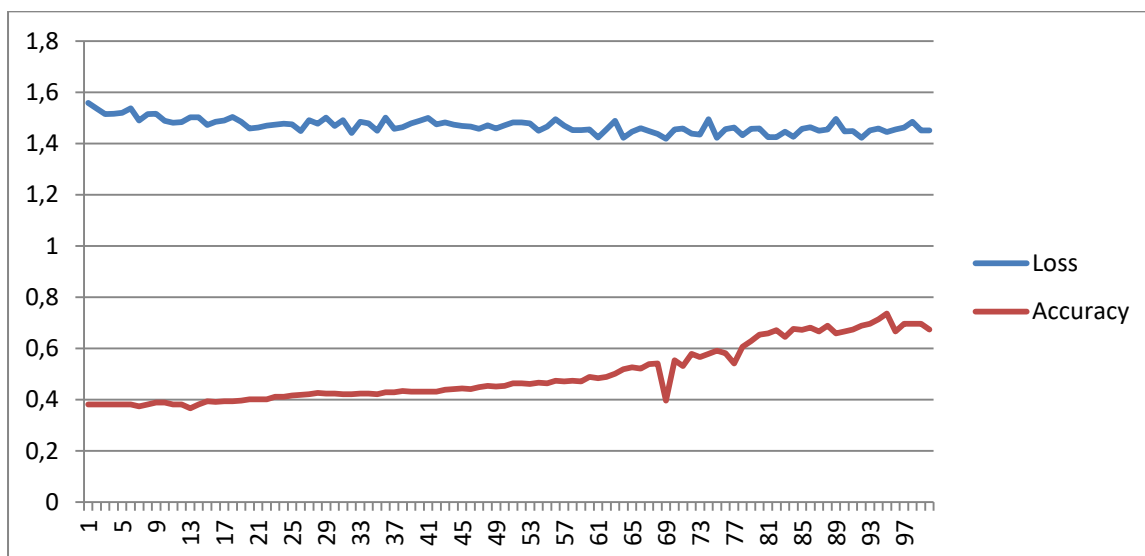


Рис 3.25. Графік втрат та точності при навчанні мережі Resnet101 протягом 100 епох на наборі даних з одним дефектним класом (F2)

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів(див.рис.3.26).



Рис 3.26. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 70.4%.

Після проведення трьох дослідів з дефектними класами (F0,F1,F2) для полегшення аналізу було складено порівняльну гістограму (див.рис 3.27).

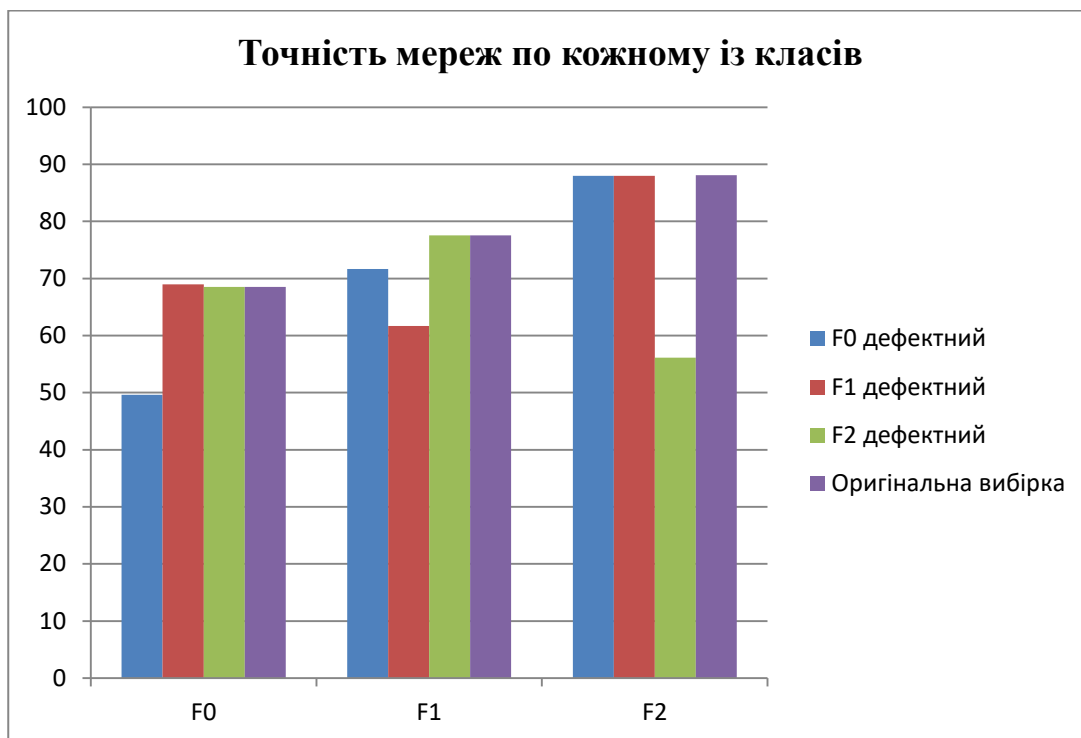


Рис 3.27. Порівняльна гістограма після навчання дефектних класів F0, F1, F2

Виникла необхідність перевірити як впливає на результат навчання, якщо у навчальній вибірці кілька дефектних класів. Для перевірки було створено дефектні навчальні класи F1 та F4 (в кожному класі було залишено по 10 зображень).

Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох на наборі даних з дефектними класами F1 та F4 (див.рис 3.28).

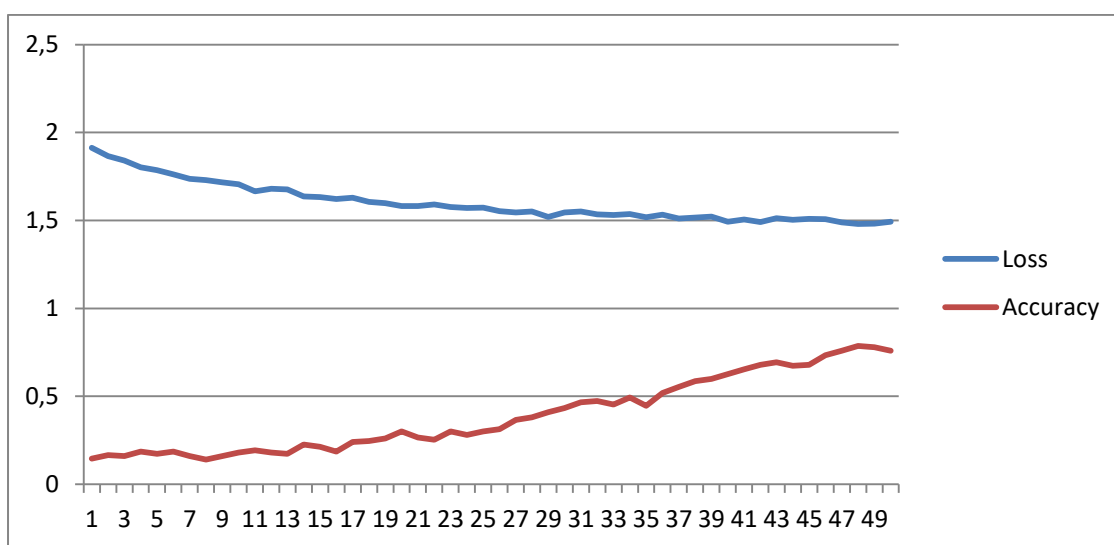


Рис 3.28. Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох на наборі даних з дефектними класами F1 та F4

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів (див.рис.3.29) :



Рис 3.29. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 58.25%.

Після цього було проведено повторний дослід для перевірки як впливає на результат навчання, якщо у навчальній вибірці кілька дефектних класів. Але тепер для перевірки було створено дефектні навчальні класи F2 та F3 (в кожному по 10 зображень).

Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох на наборі даних з дефектними класами F2 та F3 (див.рис. 3.30).

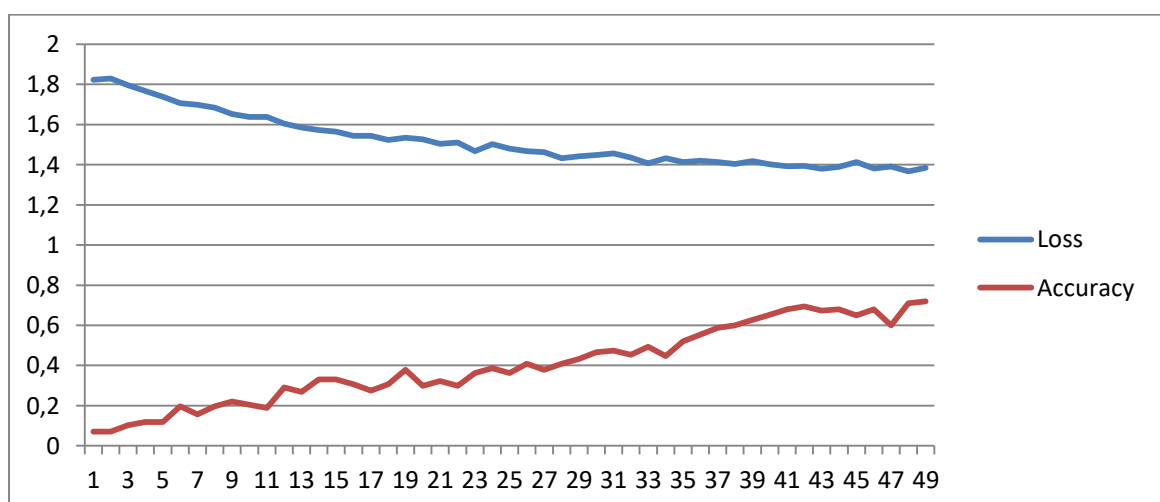


Рис 3.30. Графік втрат та точності при навчанні мережі Resnet101 протягом 50 епох на наборі даних з дефектними класами F2 та F3

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів (див.рис.3.31)



Рис 3.31. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 65.12%.

Як видно із рис 3.27. мала кількість зображень в навчальній вибірці негативно впливає на точність розпізнавання необхідного нам класу зображень.

Як відомо, у випадку використання синтетичних навчальних даних зручніше всього згенерувати відсутні навчальні приклади.

Для покращення результатів розпізнавання мережею дефектних класів було прийнято рішення використати методи описані в розділі 1.9 для збільшення набору даних. Для цього в було взято вибірку з дефектним класом F2 та збільшено кількість зображень в ній методом повороту (алгоритм див.розділ 1.9).

Після чого нейронну мережу було навчено на цій вибірці протягом 100 epoch. Графік втрат та точності при навчанні мережі Resnet101 протягом 100 epoch на наборі даних з штучно згенерованим набором даних для класу F2 методом повороту показано на рис.3.32.

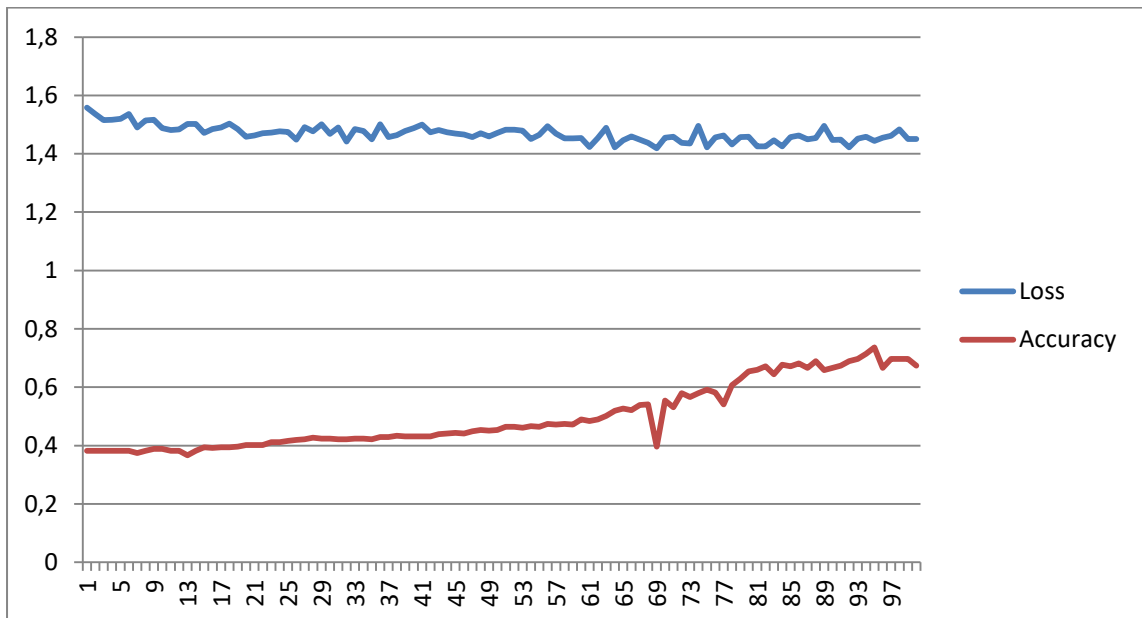


Рис 3.32. Графік втрат та точності при навчанні мережі Resnet101 протягом 100 епох на наборі даних з дефектним класом F2, який був штучно розширений методом повороту

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів показано на рис.3.33.



Рис 3.33. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 69.1%

Порівняємо результати роботи мережі з оригінальною та дефектною вибіркою для класу F2 та збільшеною вибіркою методом повороту (див. рис.3.34).

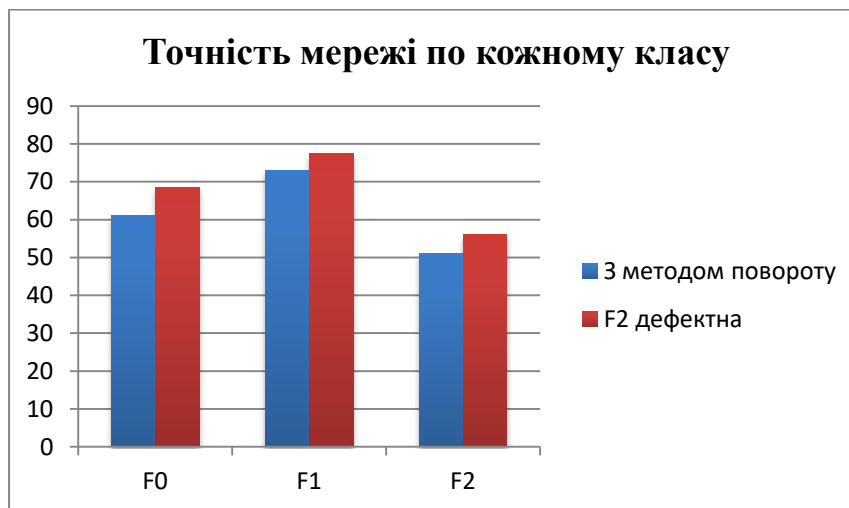


Рис 3.34. Результати роботи мережі з оригінальною та дефектною вибіркою для класу F2 та збільшеною вибіркою методом повороту

Як бачимо з гістограми на рис.3.34, метод повороту в нашому випадку не дозволяє покращити результати точності розпізнавання.

Після перевірки методу повороту було перевірено метод контрастування деяких пікселів. Для цього в досліді було взято вибірку з дефектним класом F2 та збільшено кількість зображень в ній даним методом (огляд методу див. в розділі 1.9).

Після чого нейронну мережу було навчено на цій вибірці протягом 100 епох. Графік втрат та точності при навчанні мережі на наборі даних з дефектним класом F2, який був штучно розширений методом контрастування деяких пікселів показано на рис.3.35).

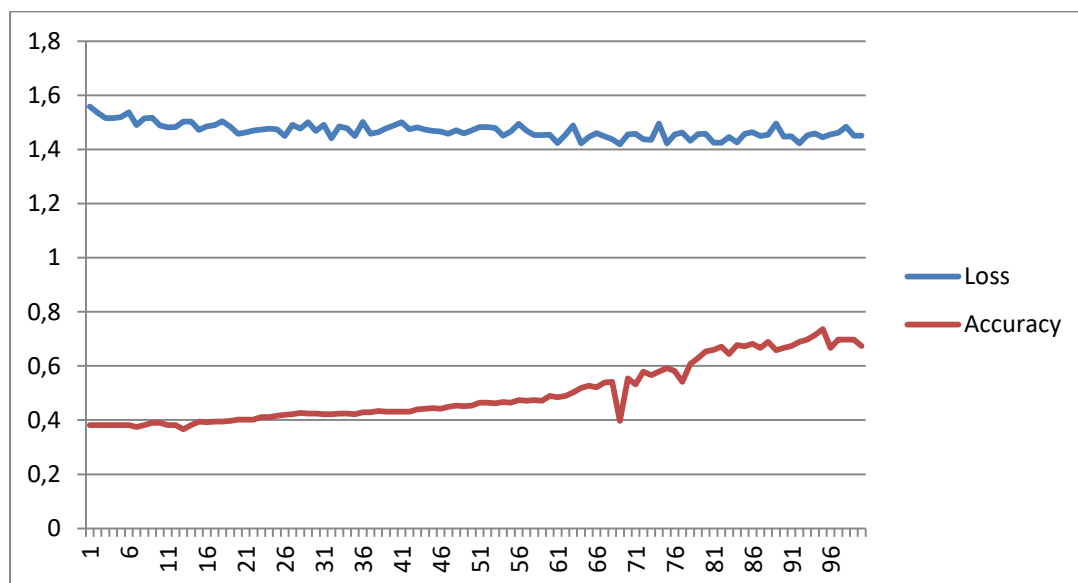


Рис 3.35. Графік втрат та точності при навчанні мережі Resnet101 протягом 100 епох на наборі даних з дефектним класом F2, який був штучно розширений методами масштабування та розмиття деяких пікселів

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів див. на рис.3.36.



Рис.3.36. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 71.4%. Порівняльні результати роботи мережі з оригінальною дефектною вибіркою для класу F2, збільшеною вибіркою методом повороту та методами контрастування деяких пікселів показано на рис. 3.37.

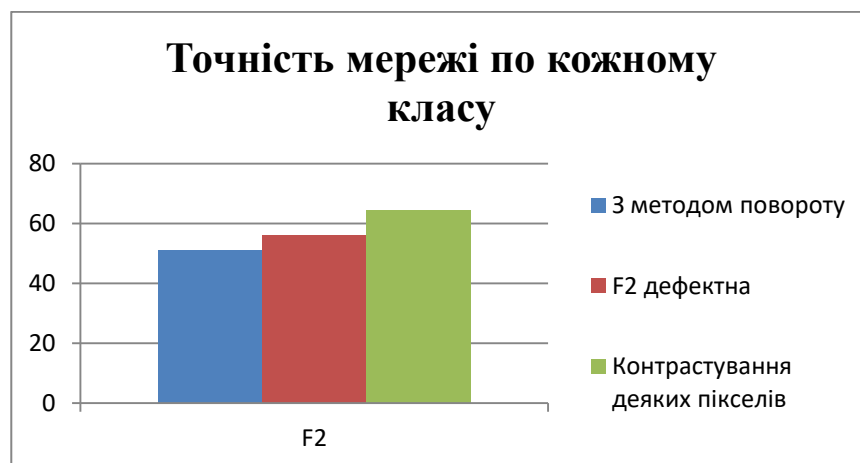


Рис.3.37. Порівняльні результати роботи мережі з оригінальною дефектною вибіркою для класу F2, збільшеною вибіркою методом повороту та методами контрастування деяких пікселів

В подальшому для покращення результатів розпізнавання мережею дефектних класів було прийнято рішення використати методи сегментації та

апроксимації зображень (опис аказаних алгоритмів див. в розділі 1.9). Для цього в досліді було взято початкову вибірку з дефектним класом F2 та збільшено кількість зображень в ній вказаним методом.

Після чого нейронну мережу було навчено на цій вибірці протягом 100 епох. Графік втрат та точності при навчанні мережі на наборі даних з дефектним класом F2, який був штучно розширений методом сегментування та апроксимації показано на рис.3.38.

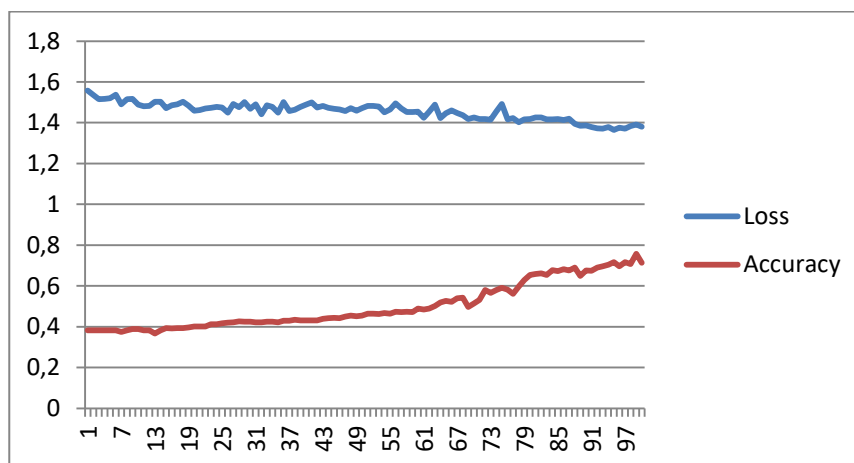


Рис 3.38. Графік втрат та точності при навчанні мережі Resnet101 протягом 100 епох на наборі даних з дефектним класом F2, який був штучно розширений методом зрушення в кольоровому просторі

Після чого нейронну мережу було перевірено на тестовому наборі даних. Результат роботи мережі на тестовому наборі даних для кожного з класів див. на рис.3.39.



Рис.3.39. Результат роботи мережі на тестовому наборі даних для кожного з класів

Загальна точність моделі склала 70.92%

Порівняльні результати роботи мережі з оригінальною дефектною вибіркою для класу F2, збільшеною вибіркою методом повороту, методами контрастування деяких пікселів та методами сегментації на апроксимації показано на рис.3.40.

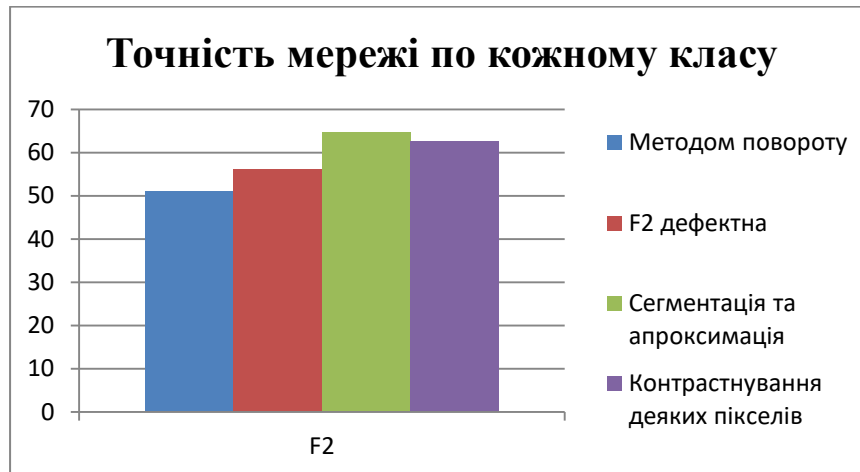


Рис 3.40. Результати роботи мережі з оригінальною дефектною вибіркою для класу F2, збільшеною вибіркою методом повороту, методами контрастування деяких пікселів та методами сегментації на апроксимації

3.3 Донавчання нейронної мережі з використання можливостей трансферного навчання

Оскільки інколи при навчанні мережі може бути непередбачений дефектний клас, а навчання мережі з нуля займає великий проміжок часу, то можливість трансферного навчання, як ніколи стає актуальним, оскільки дозволяє донавчити в мережі дефектний клас або додати новий клас для розпізнавання (див.розділ 2).

Для перевірки можливості донавчання мережі було обрано класи F0-F4 в кожному з яких було зменшено кількість навчальних зображень до 10. Результати цієї навченої моделі показано на рис.3.41.



Рис 3.41. Результати роботи мережі з навченими дефектними класами F0-F4

Основаючись на аналізі результатів роботи мережі показаної на рис.3.41 доцільно зробити висновок про до навчання кожного класу, оскільки точність розпізнавання є низькою. Для цього, кожному з навчених раніше моделей з дефектним класом було донавченно на зображеннях, які були в початковій вибірці, але в процесі формування дефектної вибірки були виключенні, для створення дефектного класу.

Час до навчання по кожному класу склав 100 епох. Графік помилки при донавчанні показанно на рис. 3.42.



Рис 3.42. Графік помилки при донавчанні

Після проведення до навчання кожної моделі для полегшення аналізу складено порівняльну гістограму, в якій показано на рис. 3.43 результати роботи мереж по дефектному класі до та після трансферного навчання.

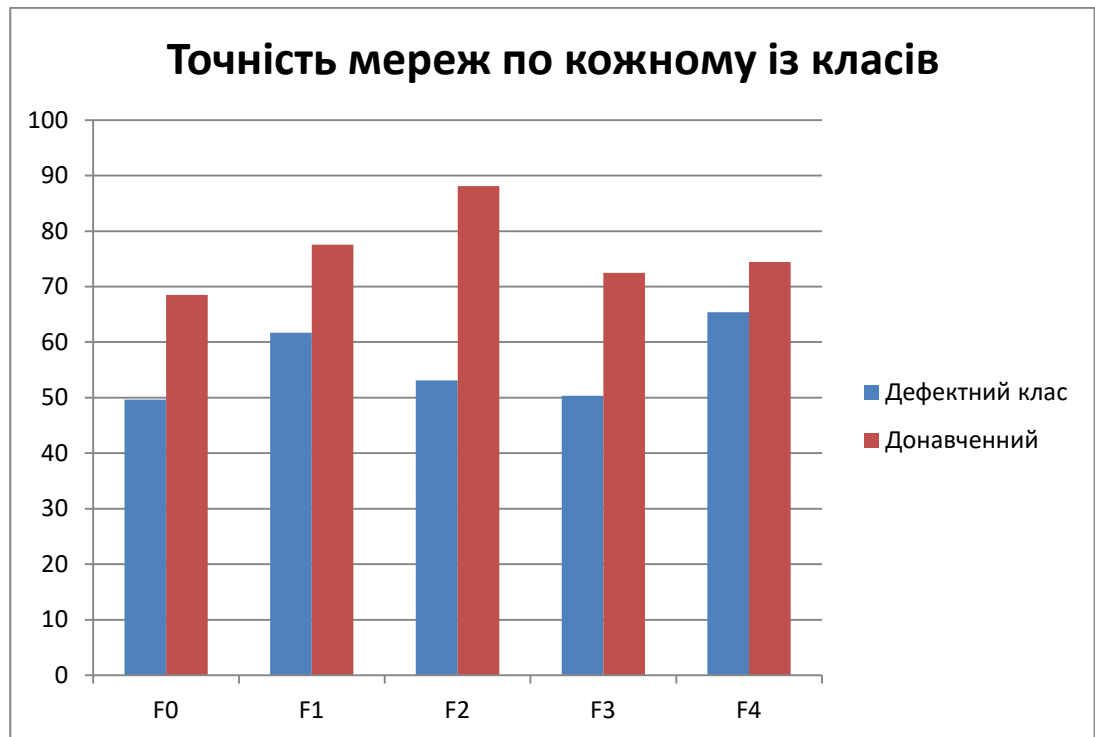


Рис 3.43. Порівняльна гістограма результатів точності мережі здефектним класом і класом донавченим за допомогою трансферного навчання

4. РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ МЕДИЧНОЇ СИСТЕМИ ТА АНАЛІЗ ЇЇ ХАРАКТЕРИСТИК ШЛЯХОМ МОДЕЛЮВАННЯ

4.1 Опис програмної складової інтелектуальної системи медичної діагностики

Розробку програмної частини інтелектуальної системи медичної діагностики було вирішено розробити на мові програмування Python. Це високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис Python мінімалістичний. У той же час його стандартна бібліотека включає великий обсяг різноманітних функцій.

Python підтримує структурне, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється під вільною ліцензією Python Software Foundation License, що дозволяє використовувати його без обмежень в будь-яких додатках, включаючи пропрієтарні[107].

Python - активно розвивається як мова програмування, нові версії з додаванням мовних властивостей виходять приблизно один раз на два з половиною роки.

Python працює на всіх відомих платформах. Існують порти під Microsoft Windows, практично всі варіанти UNIX-систем (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS / 2, Amiga, HaikuOS, AS / 400 і навіть OS / 390, Windows Mobile, Symbian і Android[108].

КАФЕДРА АКІК

НАУ 20 19 32 000 ПЗ

<i>Розроб.</i>	<i>Хоцянівський В.П.</i>			<i>Інтелектуальна медична система на основі нейронних мереж RESNET</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Керівник</i>	<i>Синєглазов В.М.</i>						
<i>Консульт.</i>							
<i>Н.контр.</i>	<i>Тупіцин М. Ф.</i>						
<i>Зав.кафедри</i>	<i>Синєглазов В.М.</i>						
					<i>151 204М</i>		

У міру старіння платформи її підтримка в основній гілці мови припиняється. Наприклад, з серії 2.6 припинена підтримка Windows 95, Windows 98 і Windows ME . Однак на цих платформах можна використовувати попередні версії Python - на даний момент спільнота активно підтримує версії Python починаючи від 2.3 (для них виходять виправлення). Це дозволить інтелектуальній системі працювати навіть на морально застарілому обладнанні[108].

При цьому, на відміну від багатьох портованих систем, для всіх основних платформ Python має підтримку характерних для даної платформи технологій (наприклад, Microsoft COM / DCOM). Більш того, існує спеціальна версія Python для віртуальної машини Java - Jython, що дозволяє інтерпретатору виконуватися на будь-якій системі, що підтримує Java, при цьому класи Java можуть безпосередньо використовуватися з Python і навіть бути написаними на Python. Також кілька проектів забезпечують інтеграцію з платформою Microsoft .NET, основні з яких - IronPython і Python.Net.

Python підтримує динамічну типізацію, тобто, замість «присвоювання значення змінної» краще говорити про «зв'язуванні значення з деяким ім'ям». В Python є вбудовані типи: булевий, рядок, Unicode-рядок, ціле число довільної точності, число з плаваючою комою, комплексне число тощо.

При проектуванні інтелектуальної системи виникає необхідність отримання інформації, що вимагає багаторазового запуску ПЗ при змінених коефіцієнтах і постійного редагування властивостей. При цьому важлива наявність механізму управління коефіцієнтами, який би забезпечував зручний інтерфейс між програмою і користувачем. Ці та інші труднощі можна подолати за допомогою розробки графічного інтерфейсу користувача (GUI - Graphical User Interface).

Для розробки користувацького інтерфейсу було обрано використовувати бібліотеку Tkinter, оскільки вона входить до складу Python та дозволяє створювати графічні додатки.

Додаток GUI може складатися як з одного основного вікна, так і декількох вікон і здійснювати виведення графічної і текстової інформації в основне вікно програми і в окремі вікна.

Ряд функцій Tkinter призначений для створення стандартних діалогових вікон відкриття і збереження файлу, друку, вибору шрифту, вікна для введення даних і ін.

Отже, використання мови програмування Python та графічного інтерфейсу заснованого на відкритій бібліотеці Tkinter дозволить розробити універсальний додаток, який буде мультиплатформенним, та інтуїтивно зрозумілим, а це забезпечить швидкий та зручний спосіб отримання потрібної кінцевим користувачем інформації.

4.2 Лістинг програми

Для створення елементів GUI необхідно створити змінну і викликати метод для створення того чи іншого вікна інтерфейсу. Після додавання елементу інтерфейсу необхідно задати його ім'я, яке буде ідентифікувати даний об'єкт серед інших об'єктів.

Заготовка додатку з елементами інтерфейсу, що добавлені з панелі інструментів має вигляд, зображений на рис. 4.1.

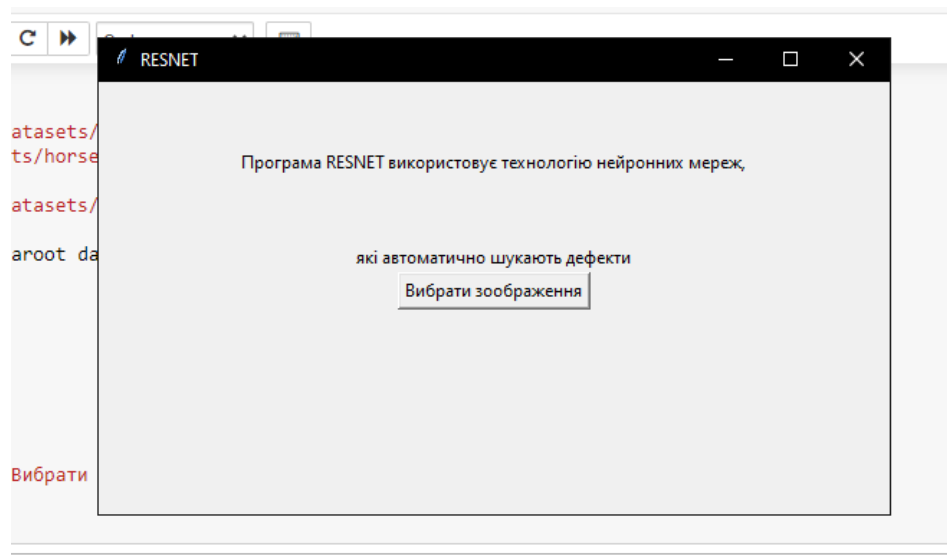


Рис.4.1. Інтерфейс користувача інтелектуальної медичної системи

Для реалізації інтерфесу що зображений на рис.4.1 необхідно підключити бібліотеку Tkinter наступним кодом

```
from tkinter import *
from tkinter import filedialog as fd
from tkinter import messagebox
from numpy import exp, array, random, dot
```

Для зручності роботи з проєктованим інтерфейсом реалізуємо його окремою функцією з назвою *GUIf*. Функція - це іменна частина програми, що може викликатися з будь якої частин програми стільки разів, скільки необхідно для роботи програми.

Реалізація функції *GUIf*:

```
def GUIf:
    root = Tk()
    root.title("RESNET")
    root.geometry("550x300+300+150")
    root.resizable(width=True, height=True)
    f_top = Frame(root)
    l1 = Label(f_top, height=7, text="Програма RESNET
використовує технологію нейронних мереж,")
    l2 = Label(f_top, text="які автоматично шукають дефекти")
```

Оскільки система працює з зображеннями то потрібно надати можливість завантажувати потрібне нам зображення для цього були реалізовані підпрограми *openfn* та *clicked*.

Реалізація функції *openfn*:

```
def openfn():
    filename = filedialog.askopenfilename(title='open')
    return filename
```

Реалізація функції *clicked*:

```
def clicked():
    x = openfn()
    if (os.path.isfile(x)):
        os.remove("x")

    shutil.copyfile(x, "x")

    !python test.py --dataroot datasets /testC --name classes -
-model test --no_dropout
    !python view.py
```

Розроблений графічний інтерфейс, передає картинку на скрипт *test.py* який обробляє зображення, готовий до подальшої уніфікації. Перейдемо до розробки системи обробки зображень. Для цього було використано стандартні математичні бібліотеки Python, але основою системи є ШНМ. Для її впровадження поверх розробленого ПЗ було підключено бібліотеку TensorFlow. Це відкрита бібліотека, написана мовою програмування Python. При її розробці було зосереджено увагу на тому, щоб вона була легкою, модульною та розширюваною. TensorFlow містить численні реалізації широко вживаних блоків нейронних мереж, таких як шари, цільові та передатні функції, оптимізатори, та безліч інструментів для роботи з картинками та текстами. Програмний код бібліотеки розміщено на GitHub.

Реалізація архітектури Resnet:

```
import tensorflow as tf
from config import NUM_CLASSES
from models.residual_block import make_basic_block_layer,
make_bottleneck_layer

class ResNetTypeI(tf.keras.Model):
    def __init__(self, layer_params):
        super(ResNetTypeI, self).__init__()
```

```

        self.conv1 = tf.keras.layers.Conv2D(filters=64,
                                             kernel_size=(7,
7),
                                             strides=2,
                                             padding="same")
        self.bn1 = tf.keras.layers.BatchNormalization()
        self.pool1 = tf.keras.layers.MaxPool2D(pool_size=(3,
3),
                                             strides=2,
                                             padding="same")

        self.layer1 = make_basic_block_layer(filter_num=64,
blocks=layer_params[0])
        self.layer2 = make_basic_block_layer(filter_num=128,
blocks=layer_params[1],
                                             stride=2)
        self.layer3 = make_basic_block_layer(filter_num=256,
blocks=layer_params[2],
                                             stride=2)
        self.layer4 = make_basic_block_layer(filter_num=512,
blocks=layer_params[3],
                                             stride=2)

        self.avgpool =
tf.keras.layers.GlobalAveragePooling2D()
        self.fc = tf.keras.layers.Dense(units=NUM_CLASSES,
activation=tf.keras.activations.softmax)

    def call(self, inputs, training=None, mask=None):
        x = self.conv1(inputs)
        x = self.bn1(x, training=training)
        x = tf.nn.relu(x)

```

```

        x = self.pool1(x)
        x = self.layer1(x, training=training)
        x = self.layer2(x, training=training)
        x = self.layer3(x, training=training)
        x = self.layer4(x, training=training)
        x = self.avgpool(x)
        output = self.fc(x)

    return output

class ResNetTypeII(tf.keras.Model):
    def __init__(self, layer_params):
        super(ResNetTypeII, self).__init__()
        self.conv1 = tf.keras.layers.Conv2D(filters=64,
                                             kernel_size=(7,
7),
                                             strides=2,
                                             padding="same")

        self.bn1 = tf.keras.layers.BatchNormalization()
        self.pool1 = tf.keras.layers.MaxPool2D(pool_size=(3,
3),
                                             strides=2,
                                             padding="same")

        self.layer1 = make_bottleneck_layer(filter_num=64,
blocks=layer_params[0])
        self.layer2 = make_bottleneck_layer(filter_num=128,
blocks=layer_params[1],
                                             stride=2)
        self.layer3 = make_bottleneck_layer(filter_num=256,
blocks=layer_params[2],
                                             stride=2)

```

```

        self.layer4 = make_bottleneck_layer(filter_num=512,
blocks=layer_params[3],
                                                    stride=2)

        self.avgpool =
tf.keras.layers.GlobalAveragePooling2D()
        self.fc = tf.keras.layers.Dense(units=NUM_CLASSES,
activation=tf.keras.activations.softmax)

    def call(self, inputs, training=None, mask=None):
        x = self.conv1(inputs)
        x = self.bn1(x, training=training)
        x = tf.nn.relu(x)
        x = self.pool1(x)
        x = self.layer1(x, training=training)
        x = self.layer2(x, training=training)
        x = self.layer3(x, training=training)
        x = self.layer4(x, training=training)
        x = self.avgpool(x)
        output = self.fc(x)
        return output

def resnet_18():
    return ResNetTypeI(layer_params=[2, 2, 2, 2])
def resnet_34():
    return ResNetTypeI(layer_params=[3, 4, 6, 3])
def resnet_50():
    return ResNetTypeII(layer_params=[3, 4, 6, 3])
def resnet_101():
    return ResNetTypeII(layer_params=[3, 4, 23, 3])
def resnet_152():
    return ResNetTypeII(layer_params=[3, 8, 36, 3])

```

Також слід додати функції легкого вибору потрібної архітектури. Для цього реалізовано наступний скрипт навчання мережі:

```
from __future__ import absolute_import, division,
print_function
import tensorflow as tf
from models.resnet import resnet_18, resnet_34, resnet_50,
resnet_101, resnet_152
import config
from prepare_data import generate_datasets
import math

def get_model():
    model = resnet_50()
    if config.model == "resnet18":
        model = resnet_18()
    if config.model == "resnet34":
        model = resnet_34()
    if config.model == "resnet101":
        model = resnet_101()
    if config.model == "resnet152":
        model = resnet_152()
    model.build(input_shape=(None, config.image_height,
config.image_width, config.channels))
    model.summary()
    return model

if __name__ == '__main__':
    # GPU settings
    gpus = tf.config.experimental.list_physical_devices('GPU')
    if gpus:
        for gpu in gpus:
            tf.config.experimental.set_memory_growth(gpu,
True)
```

```

    # get the original_dataset
    train_dataset, valid_dataset, test_dataset, train_count,
valid_count, test_count = generate_datasets()

    # create model
    model = get_model()

    # define loss and optimizer
    loss_object =
tf.keras.losses.SparseCategoricalCrossentropy()
    optimizer = tf.keras.optimizers.Adadelta()

    train_loss = tf.keras.metrics.Mean(name='train_loss')
    train_accuracy =
tf.keras.metrics.SparseCategoricalAccuracy(name='train_accuracy')

    valid_loss = tf.keras.metrics.Mean(name='valid_loss')
    valid_accuracy =
tf.keras.metrics.SparseCategoricalAccuracy(name='valid_accuracy')

    @tf.function
    def train_step(images, labels):
        with tf.GradientTape() as tape:
            predictions = model(images, training=True)
            loss = loss_object(y_true=labels,
y_pred=predictions)
            gradients = tape.gradient(loss,
model.trainable_variables)

        optimizer.apply_gradients(grads_and_vars=zip(gradients,
model.trainable_variables))

        train_loss(loss)

```

```

        train_accuracy(labels, predictions)

    @tf.function
    def valid_step(images, labels):
        predictions = model(images, training=False)
        v_loss = loss_object(labels, predictions)

        valid_loss(v_loss)
        valid_accuracy(labels, predictions)

    # start training
    for epoch in range(config.EPOCHS):
        train_loss.reset_states()
        train_accuracy.reset_states()
        valid_loss.reset_states()
        valid_accuracy.reset_states()
        step = 0
        for images, labels in train_dataset:
            step += 1
            train_step(images, labels)
            print("Epoch: {}/ {}, step: {}/ {}, loss: {:.5f},
accuracy: {:.5f}".format(epoch + 1,

config.EPOCHS,

step,

math.ceil(train_count / config.BATCH_SIZE),

train_loss.result(),

train_accuracy.result()))

        for valid_images, valid_labels in valid_dataset:
            valid_step(valid_images, valid_labels)

```



```

        print("Epoch:  {}/{}", train_loss: {:.5f}, train
accuracy: {:.5f}, "
            "valid loss: {:.5f}, valid accuracy:
{:.5f}").format(epoch + 1,

config.EPOCHS,

train_loss.result(),

train_accuracy.result(),

valid_loss.result(),

valid_accuracy.result()))

    model.save_weights(filepath=config.save_model_dir,
save_format='tf')

```

Результатом роботи вищеназваного коду є сортування картинок по класам, але для коректної роботи медичної системи потрібне не тільки класифікація зображень, а й виділення шуканих зон, для реалізації цього завдання були використанні карти активації

Кarti активації мережі — це спеціальна техніка, яка використовується для отримання регіонів зображення, які ЗНМ використала для класифікації цього зображення.

Іншими словами, карта активації мережі дозволяє побачити, які регіони зображення релевантні до певного класу, які регіони зображення були найбільш важливими з точки зору ЗНМ під час класифікації.

На рисунку 4.2 зображений приклад карти активації мережі, отриманий в процесі підготовки дипломної роботи. На цьому рисунку видно, як неймережа зреагувала на характерні особливості, характерні для даного класу.

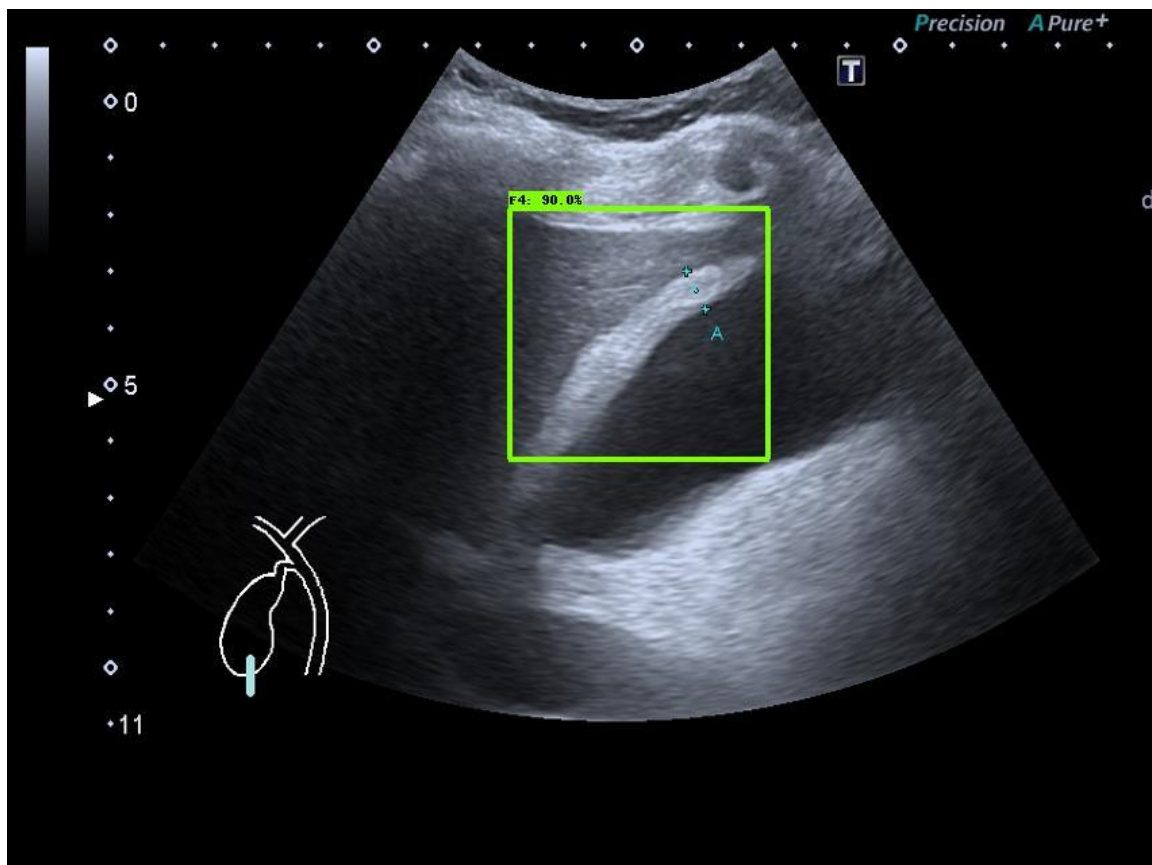


Рис.4.2. Приклад карти активації мережі

Для використання карти активації мережі в архітектурі ЗНМ має бути шар глобальної усередненої підвибірки, також відомий як шар глобального усередненого пулінгу (*global average pooling layer*) після останнього згорткового шару, а потім — звичайний повнозв'язний шар.

Це означає, що для дослідження карт активації мережі необхідно внести деякі модифікації до неї, а саме — додати цей шар, що означає необхідність повторного навчання.

Ці обмеження стосуються лише механізму, описаного в оригінальній статті, вже існує модифікація цього підходу, яка дозволяє проводити дослідження карт активації мережі без обмежень на архітектуру [109].

Такі карти використовуються під час процесу навчання ЗНМ для кращого моніторингу процесу. Вони допомагають детектувати помилки навчання ЗНМ, коли вона “звертає увагу” не на характеристики певного об’єкту, а на щось, що цей об’єкт зазвичай супроводжує. Така проблема може бути викликана неправильним вибором даних, помилками в зборі даних, помилками в попередній обробці даних, тощо. Наприклад, прикладом такої

помилкової роботи може бути реагування на ландшафт гірської місцевості, де зазвичай пасуться вівці, замість реагування на овець при спробі класифікувати тварин.

Таку помилку важко відслідкувати звичайними засобами, адже усі метрики будуть показувати високу точність класифікації (за умови, що в наборі даних мало зображень гірської місцевості з іншими тваринами).

Проте така неправильно навчена нейромережа може погано себе показати при спробі використати її для вирішення реальної задачі.

В цілому, можливість такої проблеми існує для більшості методів машинного навчання, і саме тому правильна підготовка даних при застосуванні алгоритмів машинного навчання до реальних задач є дуже важливою.

Для реалізації можливості детектування до коду були додані наступні бібліотеки:

```
from object_detection.utils import ops as utils_ops
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as
vis_util
```

Для детектування об'єктів була написана функція `plot_detections`.

Реалізація функції:

```
def plot_detections(image_np,
                    boxes,
                    classes,
                    scores,
                    category_index,
                    figsize=(12, 16),
                    image_name=None):

    image_np_with_annotations = image_np.copy()
    viz_utils.visualize_boxes_and_labels_on_image_array(
```

```

image_np_with_annotations,
boxes,
classes,
scores,
category_index,
use_normalized_coordinates=True,
min_score_thresh=0.8)
if image_name:
    plt.imshow(image_np_with_annotations)
else:
    plt.imshow(image_np_with_annotations)

```

4.3 Опис роботи програми

В результаті запуску описаного вище програмного коду відкривається діалогове вікно, що зображене на рис. 4.1. Після натискання на кнопку «Вибрати зображення» відкриється вікно, яке запропонує обрати картинку для аналізу (див. рис. 4.3).

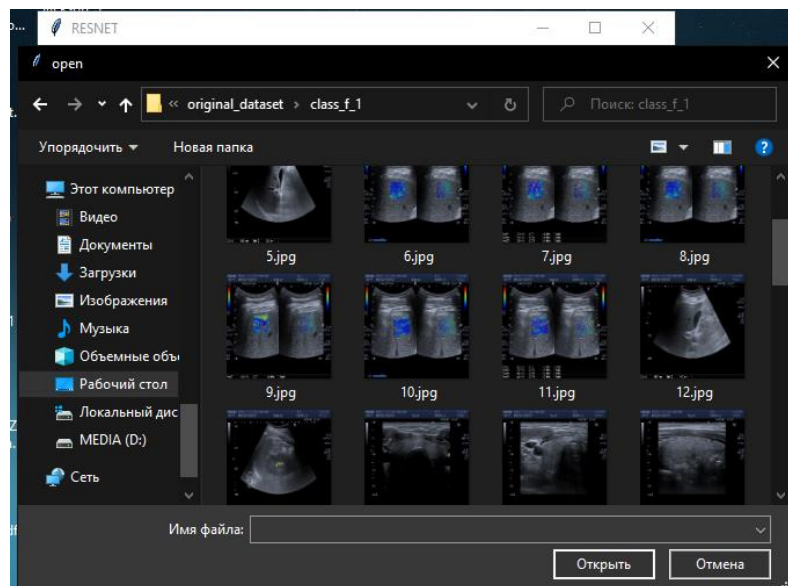


Рис.4.3 Діалогове вікно для вибору потрібного зображення для аналізу

Після вибору потрібного зображення система відкриє діалогове вікно з результатом роботи системи де буде вказано діагноз хворого, підозріла зона та степінь впевненості системи у своєму діагнозі (див рис.4.4-4.7).

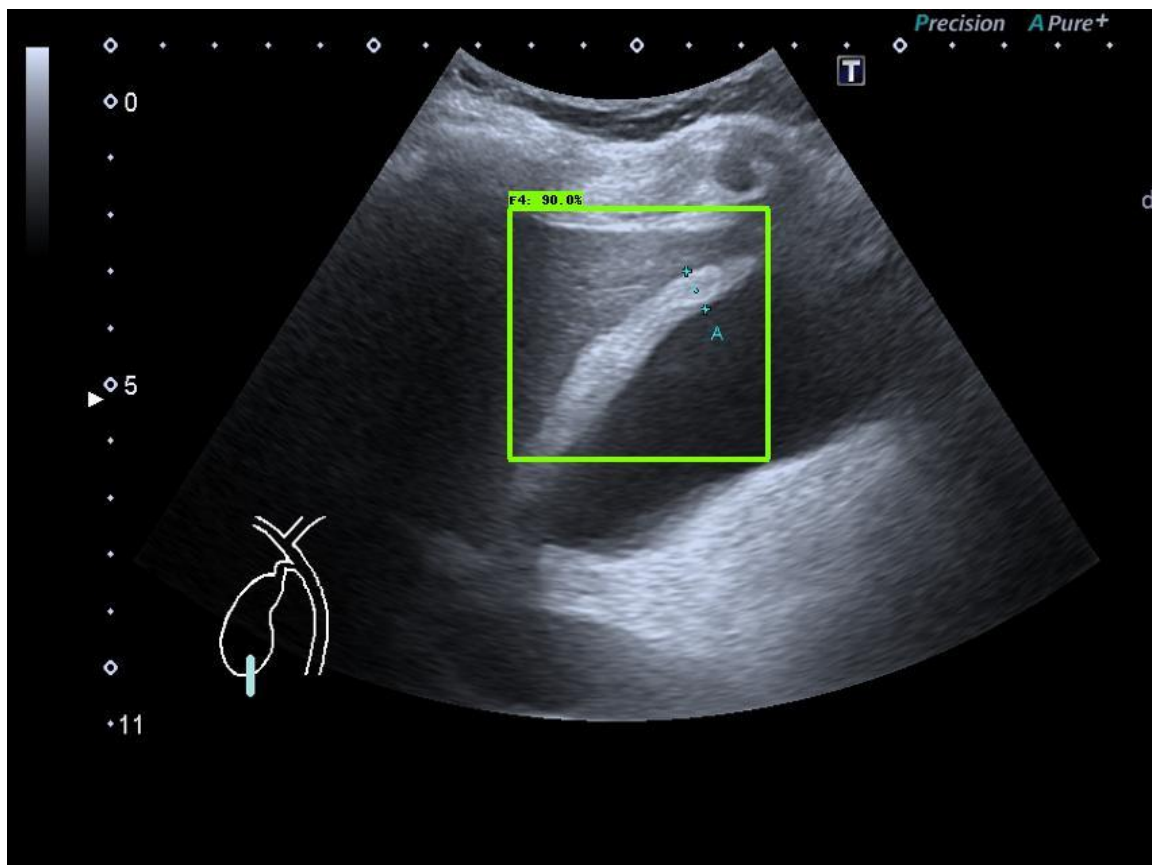


Рис.4.4. Вікно з візуалізацією де система визначила, що на зображенні стадія F4 з ймовірністю 90%

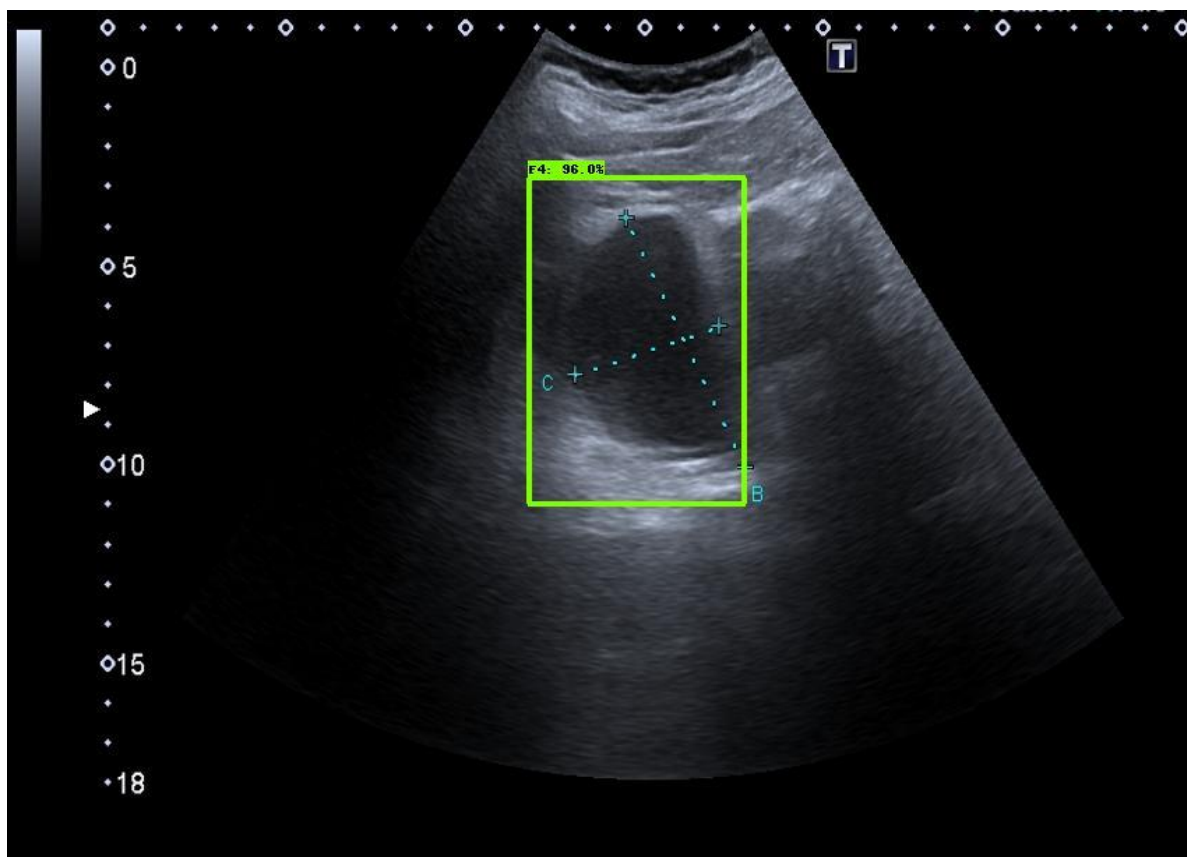


Рис.4.5 Вікно з візуалізацією де система визначила, що на зображенні стадія F4 з ймовірністю 96%

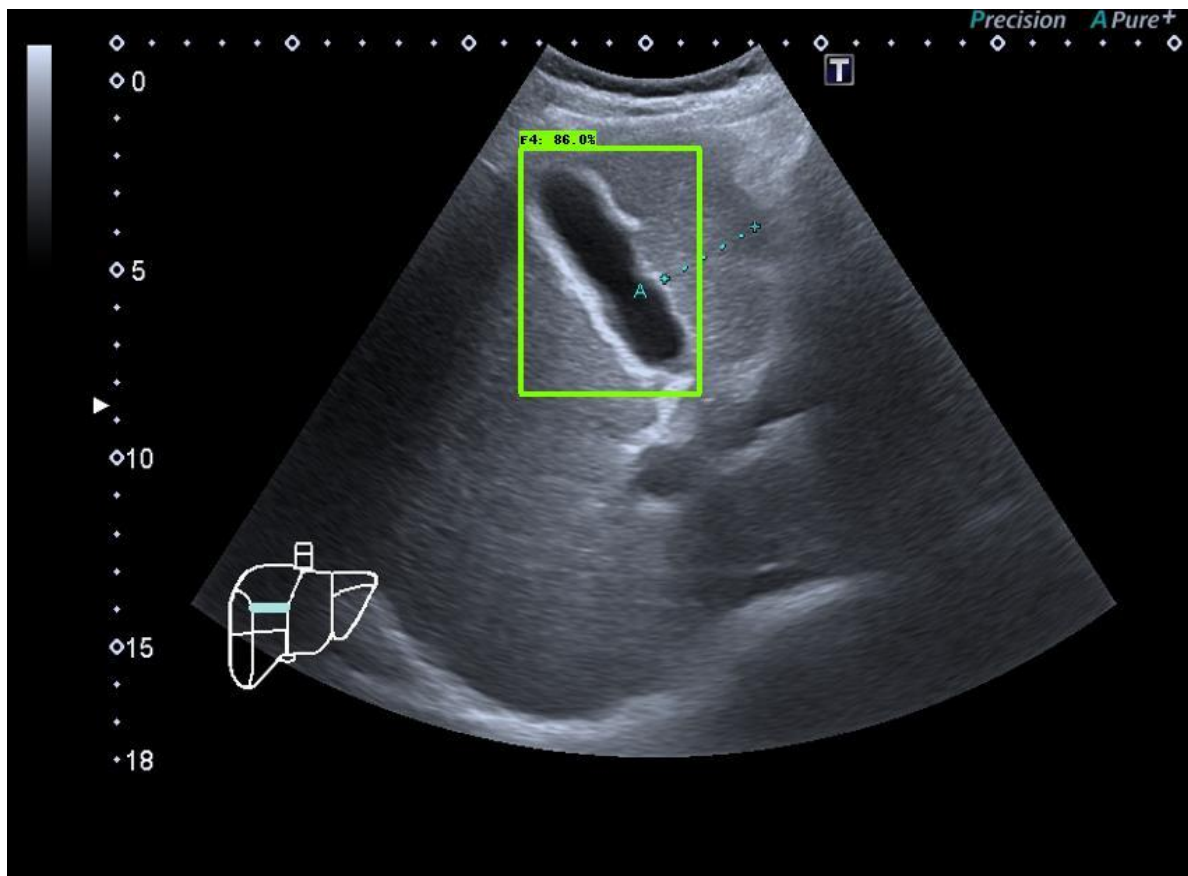


Рис.4.6 Вікно з візуалізацією де система визначила, що на зображенні стадія тадія F4 з ймовірністю 86%

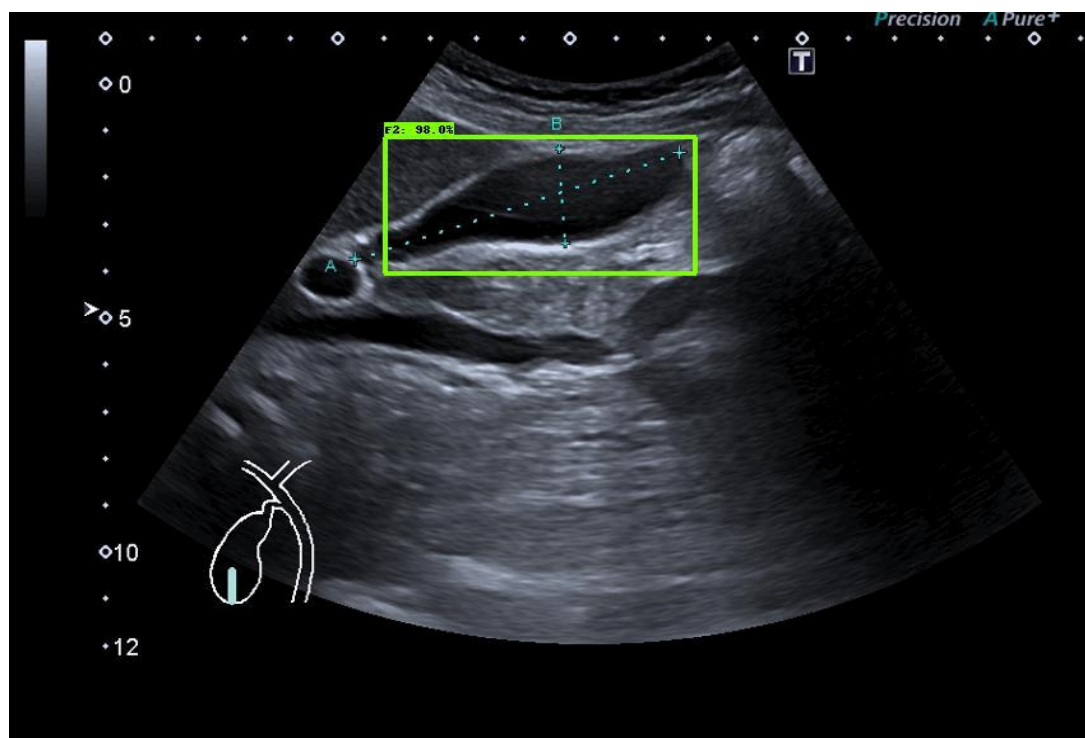


Рис.4.7 Вікно з візуалізацією де система визначила, що на зображенні стадія тадія F2 з ймовірністю 98%

Отже, розроблена програма графічному інтерфейсі дозволяє оперативно отримати інформацію лікарю для підтримки прийняття лікарем.

РОЗДІЛ 5. ОХОРОНА НАВКОЛИШНЬОГО ПРИРОДНОГО СЕРЕДОВИЩА

Охорона навколишнього середовища – це регулювання відносин у галузі охорони, використання і відтворення природних ресурсів, забезпечення екологічної безпеки, запобігання і ліквідації негативного впливу господарської та іншої діяльності на навколишнє природне середовище, збереження природних ресурсів, генетичного фонду живої природи, ландшафтів та інших природних комплексів, унікальних територій та природних об'єктів, пов'язаних з історико-культурною спадщиною.

Забруднення навколишнього середовища це дії, які привнесли в екологічну систему не властивих їй живих або неживих компонентів, фізичних або структурних змін, в результаті яких порушуються процеси круговороту і обміну речовин, а також відтоки енергії, унаслідок чого знижується продуктивність або руйнується дана екосистема.

Забруднюючі речовини зазвичай групуються по їх природі:

– фізичні забруднення, до них відносять: шумове забруднення і низькочастотна вібрація, електромагнітне забруднення, радіоактивні елементи;

– хімічні та біологічні забруднювачі, до них відносять: синтетичні органічні речовини, важкі метали, фтористі з'єднання;

– механічні, до них відносять: пил та тверді частки.

Для вирішення питань в галузі охорони навколишнього середовища займається наука – екологія. Екологія – це економіка природи й одночасне вивчення усіх взаємин живого з органічними і неорганічними компонентами

КАФЕДРА АКІК

НАУ 20 19 32 000 ПЗ

<i>Розроб.</i>	<i>Хоцянівський В.П.</i>			<i>Охорона навколишнього природного середовища</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Керівник</i>	<i>Синєглазов В.М.</i>						
<i>Консульт.</i>	<i>Мадж С.М.</i>				<i>151 204М</i>		
<i>Н.контр.</i>	<i>Тупіцин М.Ф.</i>						
<i>Зав.кафедри</i>	<i>Синєглазов В.М.</i>						

середовища. Основними задачами екології є всебічна діагностика стану природи, розробка прогнозів по зміні стану природного середовища, формування профілактичних заходів щодо охорони навколишнього середовища.

Темою моєї дипломної роботи є «Інтелектуальна медична система на основі нейронних мереж RESNET».

В цьому розділі будуть розглянуті питання щодо забруднення навколишнього середовища персональними комп'ютерами, та будуть запропоновані методи та заходи стосовно раціональної розробки використання та утилізації обладнання не завдаючи шкоди навколишньому середовищу.

5.1 Вплив об'єкту дослідження на навколишнє середовище

За деякими даними дослідників ООН, щоб створити один середньостатистичний персональний комп'ютер, потрібно в 10 разів більше хімічних речовин і палива, ніж вага кінцевого продукту[110]. Багато сировини, яка використовується в збірці комп'ютерів є токсичною. Викопне паливо лише посилює невирішену проблему глобального потепління. Відходи виробництва також не зникають, перетворюючись на звалища, або переробляються, надаючи поганий вплив на екологію. Багато користувачів і виробників помиляються, вважаючи, що зі зменшенням і удосконаленням комп'ютерів, їх негативний вплив на екологію зменшується. Тому вчені підкреслюють важливість вторинного використання обладнання. Перш ніж викинути «гаджет» в сміття, краще остаточно переконатися, що він не підлягає відновленню і тільки потім звертатися в конфігуратор комп'ютера. Можливо, він буде частково корисний в іншій збірці.

На даний момент найбільш суворим з існуючих світових стандартів екологічності для комп'ютерної техніки є стандарт TCO99. У порівнянні з попередніми він містить додаткові обмеження по частині екології, ергономіки, енергоспоживання і емісії пристроїв[111].

Найбільш значимі ярлики, такі як «Блакитний ангел», що видається Німецькою сертифікаційною організацією як знак відповідності екологічним стандартам, є великою рідкістю в сфері електроніки. Навпаки, широке поширення отримав логотип «Energy Star», якого удостоюються енергозберігаючі пристрої. Однак у випадку з ним проблема полягає в тому, що кожен виробник має право самостійно маркувати свою продукцію, не проходячи при цьому перевірок[112].

З огляду на те, що дана емблема не несе ніяких відомостей про дійсне енергоспоживання пристроїв, її цілком можна ігнорувати.

Організація по захисту навколишнього середовища Greenpeace з 2006 року оцінює виробників електроніки за кількістю важких металів і отруйних речовин, наприклад інгібіторів горіння, використовуваних ними при виробництві (інгібітор - речовина, присутність якого в невеликих кількостях призводить до запобігання або уповільнення процесів горіння або корозії; інгібітори знижують швидкість хімічних реакцій або пригнічують їх). Однак навіть оцінки такої організації, як Greenpeace, які не можуть претендувати на об'єктивність. Адже в одних випадках вона використовує перевірену інформацію, що стосується, наприклад, заходів щодо утилізації відходів, а в інших спирається тільки на дані виробника. А якщо компанія не повідомляє ніяких відомостей, то автоматично опиняється на нижніх рядках рейтингу. Крім того, енергетичні витрати на виробництво і перевезення продукції також необхідно враховувати при оцінці екологічної ефективності. Адже часи, коли техніка виготовлялася тільки на одному заводі, давно пройшли. Сьогодні окремі комплектуючі закупаються на різних підприємствах по всьому світу, після чого здійснюється складання пристроїв[113]. Тому найчастіше навіть самі компанії не можуть знати, які шкідливі речовини потрапляють в атмосферу при виготовленні їх продукції і які саме метали або токсини в ній містяться. ЖК-екрани - один з джерел парникових газів, які набагато шкідливіші діоксиду вуглецю. Рідкокристалічні монітори швидко знайшли популярність, прийшовши на зміну громіздким ЕПТ-моделями. І це не дивно,

адже вони мають тонкі корпуси і споживають значно менше електроенергії. За іншим аспектам екологічної безпеки дисплеї на основі рідких кристалів також вважалися проривом, тому що в них не використовувався газ, що містить свинець. Досить довго ніхто не звертав уваги на застосований для чищення РК-панелей трехфтористий азот (NF₃), і тільки в середині 2008 року вченими було доведено наявність даної хімічної речовини в атмосфері. Відкриття було вражаючим: у порівнянні з діоксидом вуглецю (CO₂) NF₃ є в 17 000 разів більше активним парниковим газом, а його атмосферний час напіврозпаду може складати від 550 до 740 світлових років (у CO₂ - від 30 до 40 років). Закону, який обмежував би рівень викиду NF₃, поки не існує. Виявлення енерговитрат є таким же проблематичним процесом, як і визначення кількості матеріалів, придатних для вторинної переробки, і важких металів, що містяться в пристроях. Дивовижний результат був отриманий організацією Greenpeace в ході порівняльного аналізу декількох моделей ідентичних ноутбуків з різних країн. В тачпаді Dell Vostro V13, доступного на китайському ринку, були виявлені сліди бромю. В моделі з Німеччини ця речовина теж присутня, тільки не в тачпаді, а кнопках. У лептопі, купленому в США, бром був знайдений в блоці живлення.

Схожа картина спостерігається і у інших виробників: при дослідженні продукції компанії Apple експерти виявили, що в кабелі ноутбука MacBook Pro 13 з США і Нідерландів міститься в три рази більше бромю, ніж в пристроях з Філіппін і Росії. При аналізі іншого кабелю сліди бромю виявили вже в пристроях з Росії та Нідерландів, а в моделі з США їх не було. Таким чином, надійним показником екологічності залишається тільки рівень енергоспоживання - серед субноутбуків першість належить лише декільком моделям, а решта різко відрізняються від лідерів за своїми характеристиками.

Видобувні виробництва руйнують поверхню Землі і часто забруднюють навколишнє повітря і воду. Видобуток рідкоземельних мінералів неможливий або нерентабельний без використання процесів, які завдають серйозної шкоди навколишньому середовищу. Полівінілхлорид[114], що позначається зазвичай

абревіатурою ПВХ, - це різновид пластику, що застосовується в самих різних цілях. З нього зроблена зовнішня оболонка кабелів, якими з'єднуються пристрої, він оточує електричний провід портативного комп'ютера. ПВХ присутній в музичній колекції любителів вінілових платівок. З нього роблять труби і одяг. Це дешевий, міцний і вельми поширений матеріал.

Разом з тим, за словами ІТ-аналітика «Грінпіс» Кейсі Харрелл, «ПВХ - найгірший з пластиків»[115]. Він є причиною виникнення гормонального дисбалансу, проблем в репродуктивній сфері та різних форм раку. Полівінілхлорид практично неможливо правильно утилізувати. Внаслідок старий матеріал виявляється зазвичай на звалищі з відходами, або, того гірше, спалюється з метою вилучення мідних жил і інших цінних компонентів. При його згорянні утворюється вкрай шкідливий канцероген діоксин. Звалища і хімічні поховання забруднюють джерела води. Єдиний спосіб правильно утилізувати ПВХ полягає в тому, щоб відправити його в центр небезпечних відходів.

Тривала робота комп'ютерів приводить до зниження концентрації кисню в повітрі, кількість озону, навпаки, збільшується. Озон є сильним окислювачем. Його концентрація вище гранично допустимих величин приводить до несприятливих обмінних реакцій організму. Великий вплив монітори роблять на іонний склад повітря робочої зони. Зміна цього балансу, що обумовлений збільшенням кількості позитивних іонів, приводить до негативних наслідків. Установлено, що фоновий спектр іонів у приміщеннях з моніторами характеризується надлишком негативних іонів. У процесі роботи терміналу структура спектра іонного складу повітря робочої зони істотно змінюється. Протягом 5 хв роботи монітора концентрація легких негативних іонів зменшувалася в 8 разів, а через 3 години — знизилася до рівня, близького до нуля. Істотно понизилася кількість середніх і важких негативних часточок. У той же час концентрація позитивних іонів зростала, і через 3 год. роботи монітора в повітрі робочої зони переважали позитивні часточки. Необхідно відзначити, що й у геопатогенних зонах прилади також реєструють

різке зменшення негативно заряджених іонів кисню[116], що підкреслює факт однакової фізичної природи торсіонових полів зон Землі і торсіонових полів, які генеруються моніторами, телевізорами й іншою електронною технікою.

Персональні комп'ютери, ноутбуки та інша інформаційна техніка, як відомо широко використовується в галузі наукових досліджень, промисловості, а також у повсякденному домашньому користуванні. Але будь-яка техніка стрімко старіє, їй на зміну приходять нові, більш потужні, більш сучасні ПК та оргтехніка. Поступово виникає проблема, що робити зі старою технікою, морально застарілою або з тих чи інших причин, що вийшла з ладу, яка захащує підсобні приміщення та склади.

Утилізація комп'ютерів це процес, який проводиться в кілька етапів. Найперша дія - це списання обладнання безпосередньо з підприємства. Етап другий - це розбір техніки і сортування отриманих матеріалів. Якщо деталі здатні служити вихідною сировиною, наприклад, кінескоп, деталі, в складі яких є дорогоцінні метали, то їх відправляють на очищення, а потім на повторну експлуатацію.

5.2 Розрахунок впливу об'єкту дослідження на навколишнє середовище

На даний момент вважається, що на весь ІТ- сектор приходить близько 2 % шкідливих викидів у світовому масштабі. Це значна цифра, особливо, якщо врахувати, що даний сектор розвивається і збільшується з великою швидкістю.

На думку дослідників з ООН, настав час для прийняття скоординованих міжнародних кроків для зменшення шкоди навколишньому природному середовищу, що наноситься комп'ютерним обладнанням[117]. За їхніми даними, при створенні одного середньостатистичного персонального комп'ютера загальна вага різних хімікатів і викопного палива в 10 разів перевищує вагу кінцевого продукту. Причому багато які, з цих хімікатів - токсичні, а застосування викопного палива погіршує процес глобального

потепління[118]. Ці відходи потім або викидаються на величезні звалища, або переробляються, найчастіше в погано відповідних умовах в країнах, що розвиваються, що створює істотну загрозу здоров'ю.

Не можна не відзначити неймовірну енергоефективність нових пристроїв. Якщо розглянути для прикладу товари торгової лінійки Mac Book, компанії Apple Inc., що вважаються одними із самих екологічних в індустрії, то можна виявити просто неможливо раніше низьке використання електроенергії.

Навіть самі неефективні з цих продуктів втричі перевищують вимоги суворих стандартів енергоспоживання Energy Star 6.0, які складають 25 кВт/год на рік для ноутбука. Якщо перерахувати на вати, то вийде, що Mac Book споживає менше 1 вата енергії в годину – в сто разів менше звичайної 100 Вт лампочки розжарювання. Втім, не все так просто. 75 % всієї енергії (яка використовується під час всього життєвого циклу пристрою) йде на виробництво Mac Book, а не на його експлуатацію. Саме енергію, що витрачається при використанні приладу, враховує стандарт Energy Star. А вона становить всього 19 %. Решта – переробка і транспортування.

У звіті тієї ж компанії Apple сказано, що, наприклад, 15-ти дюймовий Mac Book Pro з ретина-дисплеєм за час свого життя викидає в атмосферу 690 кг вуглекислого газу. По суті, це те ж саме енергоспоживання, виражене в кілограмах CO₂. За допомогою простого коефіцієнта ці кілограми можна перевести в кіловат-години електроенергії, виробленої на електростанції. Такі коефіцієнти розраховуються спеціальними організаціями по міжнародно схваленим протоколам, таким, наприклад, як Green house Gas Protocol, і широко використовуються бізнесом для розрахунку впливу їх виробництв на екологію.

Зрозуміло, що для різних джерел енергії та різних країн коефіцієнти виявляються різними. Наприклад, в Америці при отриманні кіловат-години енергії виробляється близько півкілограма вуглекислого газу. У Китаї, де знаходиться велика частина заводів з виробництва електроніки, коефіцієнт

виявляється близько 0,87 кг/кВт/год. У світі ж «середня температура по лікарні» виявляється близько 0,44 кг CO₂ на кВт/год.

Якщо перерахувати 690 кг CO₂ в кіловат-години за цими коефіцієнтами, навіть використовуючи консервативні китайські коефіцієнти, ми отримаємо 800 кВт/год. Це 200 кВт/год на рік, майже в 10 разів більше, ніж нормативи стандарту Energy Star. Це може спочатку здивувати, але достатньо розглянути тонкощі виробництва процесорів, які являються одними з найбільш шкідливих при виробництві комп'ютерних комплектуючих. Виробництво мікросхем з субмікронними розмірами елементів – один з найскладніших процесів в сучасній промисловості. Ця технологія увібрала безліч фізико-хімічних процесів і вимагає нанометрової точності, яка досяжна тільки при абсолютній стерильності виробничого приміщення. У цеху, де йде робота, дотримується так звана «електронна гігієна»: в робочій зоні обробки напівпровідникових пластин і на операціях вирощування кристала в літрі повітря не повинно бути більше п'яти пилинок розміром 0,5 мкм. Для порівняння, стандарти чистоти хірургічних операційних допускають вміст у тисячі разів більшої кількості пилу. Чіп – це не просто кремнієва пластинка, а складна багат шарова напівпровідникова конструкція, зведена на кремнієвій підкладці. Виробництво чіпів складається більш ніж з трьох сотень операцій, і один виробничий цикл може тривати до декількох тижнів. Практично на кожній стадії використовуються шкідливі хімікати, надточне обладнання та енерговитратні фізичні методи, такі як променеве травлення та іонна імплантація. Причому ці операції повторюються для кожного з пари десятків шарів, що складають процесор. Плюс витрати енергії на надпотужну систему вентиляції і фільтрації, що забезпечує стерильність.

Тому не дивно, що маса палива, необхідного для виробництва одного процесора, в тисячі разів більше маси самого чіпа (всього пара грамів). Загальні витрати енергії в сотні тисяч разів більше, ніж на звичайному виробництві, скажімо, пластика або металу, з яких в подальшому роблять корпуси комп'ютерів.

На сьогодні вважається, що для виробництва процесора вагою 2 грами потрібно 1,6 кг палива, 72 г хімічних реактивів і 32 кг води[119].

5.3 Розрахунок утворення відходів на робочому місці оператора системи

Комп'ютером і лазерним принтером оснащується робоче місце лікаря при роботі з системою. Офісна техніка за своєю конструкцією відноситься до класу високотехнологічних виробів. Що були у використанні вироби можна відновити шляхом заміни зношених частин на нові. Ремонт і відновлення офісної техніки виробляють спеціалізовані фірми.

1. При експлуатації комп'ютера до видаткових невідновлюваних матеріалів відносяться:

- маніпулятор «миша»;
- клавіатура.

Клавіатура і маніпулятор більш ніж на 90% складаються з пластика. Експлуатаційний термін служби, за даними виробників, становить 1 рік. Середня вага маніпулятора дорівнює 100 г. Вага клавіатури дорівнює 600-900г.

Розрахунок нормативної маси відходів, що утворюються при експлуатації комп'ютерів, проводиться розрахунково-аналітичним методом.

Кількість (маса) утворюються за рік використаних маніпуляторів типу "миша" і клавіатур розраховується за формулою:

$$M = Q \cdot (Mg \cdot 0.001) \cdot \frac{K}{K_1}$$

де Q - кількість пристроїв зазначеної моделі, шт .;

Mg - вага одного пристрою зазначеної моделі (г);

K - коефіцієнт перекладу (кг -> тонна), $K = 0.001$;

K_1 - експлуатаційний термін служби пристрою, років (за замовчуванням $K_1 = 1$);

Підставимо значення, щоб дізнатися рівень утворення відходів типу маніпулятор «миша»

$$M = 1 \cdot (100 \cdot 0.001) \cdot \frac{0.001}{1} = 0,0001$$

Підставимо значення, щоб дізнатися рівень утворення відходів типу маніпулятор клавіатура

$$M = 1 \cdot (750 \cdot 0.001) \cdot \frac{0.001}{1} = 0,00075$$

Загальна вага витратних комп'ютерних матеріалів складе 0,00085 т/рік.

2. При експлуатації принтерів і копіювальної техніки утворюються використані картриджі, що складаються більш ніж 90% з пластика. За даними виробників більшість моделей картриджів розраховані на одноразове використання і додатковій заправці не підлягають.

Кількість які виникають використаних картриджів (маса) розраховується за формулою:

$$M = \sum t \cdot 0,000001 \cdot k \cdot \frac{n}{r}, \text{ т/рік,}$$

де:

0,000001 - коефіцієнт з грам в тонну;

k - кількість листів в пачці паперу (стандартна кількість листів в пачці формату A4 - 500);

n - кількість використаних пачок паперу, шт .;

t - вага використаного картриджа, г;

r - ресурс картриджа, листів на одну заправку.

У паспортних даних на картриджі вказується ресурс, розрахований на 5% заповнення (економічний режим). При реальній експлуатації ресурс слід

зменшувати на 30-50 відсотків (в залежності від якості друку), відповідно вводити поправочний коефіцієнт.

Вихідні дані для розрахунку

Модель картриджа	Сумісність (тип апарату)	Ресурс картриджа, лист / 1 заправка	Вага порожнього картриджа, г
НРС 3903А (Canon EP-V)	Hewlett Packard Laser Jet HP 5P/5MP/6P/6MP	4000	715

Підставимо значення, щоб дізнатися рівень утворення відходів типу використані картриджі

$$M = 12 \cdot 715 \cdot 000001 \cdot 500 \cdot \frac{9}{\frac{4000}{2}} = 0,019 \text{ т/год}$$

Загальна вага витратних комп'ютерних матеріалів типу використані картриджі, маніпулятор клавіатура та маніпулятор «миша» складе 0,01985 т/рік.

5.4 Шляхи покращення ситуації, рекомендації чи заходи по зменшенню впливу

До складу комп'ютера входить безліч металів таких як золото, срібло, алюміній, мідь та інших. Успіху в утилізації персональних комп'ютерів можна досягнути завдяки вторинній переробці.

Переробка комп'ютерів, електронна переробка — утилізація комп'ютерів та будь-яких інших електронних пристроїв. Це повна деконструкція

електронних пристроїв для того, щоб скоротити витрати сировини та зберегти якомога більше матеріалів зі старої та зламаної техніки.

У 2009 році, 38 % комп'ютерів і 25 % від загального обсягу електронних відходів було перероблено в Сполучених Штатах Америки, в порівнянні з 2006 роком відповідно 5 % і 3 %. З моменту свого створення на початку 1990-х років, все більше і більше пристроїв переробляються в усьому світі за рахунок збільшення рівня інформованості та інвестицій. В основному електронна обробка відбувається для того, щоб відновити цінні рідкісноземельні і дорогоцінні метали, які знаходяться в дефіциті, а також пластмаси. Вони будуть перепродані або використані в нових пристроях після очищення, в результаті створюючи економіку замкненого циклу[120].

Переробка є екологічно чистою, оскільки вона запобігає потраплянню небезпечних відходів, у тому числі важких металів і канцерогенів, в атмосферу або водойми, а також утворенню звалищ. Хоча електроніка складає невелику частку від загального обсягу утворених відходів, вона набагато небезпечніша. Є суворі закони, спрямовані на дотримання і заохочення утилізації побутової техніки, найбільш впливовими з яких є Директива Електронних Відходів та Електронного Обладнання Європейського Союзу і Акт Національної Утилізації Комп'ютерів Сполучених Штатів.

Застарілі комп'ютери та стара електроніка є цінним джерелом для вторинної сировини при переробці, з іншого боку вони є джерелом токсинів та канцерогенів. Швидкий розвиток технології, низька початкова вартість та передбачене старіння призвели до швидко зростаючого профіциту комп'ютерів та інших електронних компонентів по всьому світу. Технічні рішення доступні, але в більшості випадків перед застосуванням технічного рішення необхідно здійснити нормативно-правові основи, системи зборів, логістики, а також інші послуги. За оцінками Управління з охорони навколишнього середовища США, від 30 до 40 мільйонів залишків ПК класифікуються як «небезпечні побутові відходи». Рада національної безпеки

вважає, що 75% всіх персональних комп'ютерів, проданих колись, тепер є електронним сміттям.

У 2007 Управління з охорони навколишнього середовища США заявило, що понад 63 мільйони комп'ютерів в США задля заміни були продані або викинуті. Сьогодні 15 % електронних пристроїв та устаткувань перероблюються в Сполучених Штатах. Більшість електронних відходів направляється на звалище або сміттєспалювальний завод, який випускає шкідливі елементи, такі як свинець, ртуть та кадмій в ґрунт, тим самим негативно впливаючи на навколишнє середовище[121].

Багато матеріалів, що використовуються в комп'ютерних пристроях, можуть бути відновлені для використання в майбутньому виробництві. Повторне використання олова, кремнію, заліза, які в достатній кількості присутні в комп'ютерах або інших електронних пристроях, може зменшити витрати на будівництво нових систем. Компоненти часто містять свинець, мідь, золото та інші цінні матеріали, придатні для утилізації.

Комп'ютерні компоненти містять багато токсичних сполук, таких як діоксини, поліхлоровані біфеніли (ПХБ), кадмій, хром, радіоактивні ізотопи і ртуть. А звичайний монітор комп'ютера може містити понад 6 % свинцю, велика частина якого в свинцевому склі з електронно-променевою трубкою (ЕПТ). А стандартні 15-дюймові (38 см) монітори комп'ютера можуть містити 1 кілограм (1.5 фунти) свинцю, але інші монітори можуть мати до 4 кілограм (8 фунтів) свинцю. Друковані плати містять значні кількості свинцю та олова, припої, які, швидше за все, потрапляють у ґрунтові води. Переробка (наприклад, спалювання та кислотні обробки) повинні зберегти ці дорогоцінні сполуки, але можуть створити або синтезувати отруйні побічні речовини.

Експорт відходів у країни з більш низькими екологічними стандартами є основною проблемою. Базельська конвенція включає небезпечні відходи, але не регулює обмеження по кількості, такі як ЕПТ екрани, які не можуть бути експортовані трансконтинентально без попередньої згоди обох країн експорту на отримання відходів. Компанії можуть знайти її економічно ефективною в

короткостроковій перспективі, щоб продати застарілі комп'ютери для менш розвинених країн з не суворими правилами[122]. Вважається, що більшість надлишків ноутбуків направляються до країн, що розвиваються, під виглядом «звалища електронних відходів». Висока вартість роботи та багаторазове використання ноутбуків, комп'ютерів та комплектуючих (наприклад, оперативної пам'яті) може допомогти оплатити вартість транспортування багатьох непотрібних «товарів».

Методи переробки:

- Утилізація споживачами. Варіанти утилізація споживачами складаються з продажу, пожертвуванню комп'ютерів безпосередньо організаціям, відправленню пристроїв безпосередньо до їх виробників або отриманню комплектуючих для відновлення або переробки.
- Корпоративна утилізація. Підприємства шукають економічно ефективні способи для переробки великої кількості комп'ютерної техніки, але стикаються з більш складними технологічними процесами. Підприємства також розглядають варіанти продажу або встановлення зв'язку з Виробниками Оригінального Устаткування (ВОУ) і організаціями з утилізації. Деякі компанії забирають непотрібне обладнання інших підприємств, стирають дані з систем і дають оцінку залишкової вартості продукту. Для пристроїв, які мають цінність, фірми купують запчастини, ремонтують і продають відновлені продукти тим, хто шукає більш дешеві варіанти, ніж купівля нових.
- Продаж. Інтернет-аукціони є альтернативою для споживачів, які бажають перепродати товар за готівку, з врахуванням комісії, не ризикуючи додатковими витратами, адже товар за платним оголошенням може і не продатися. Інтернет оголошення можуть бути ризикованими через шахрайство, підробку документів та мінливість користувачів.

Одне з нововведень для утилізації друкованих плат придумали Співробітники з Національної фізичної лабораторії Великобританії,

продемонстрували можливість спеціальної суміші, яку розчинюють у гарячій воді. Дія якої зумовлює відшарування електронних компонентів.

Таким чином 90% компонентів нових друкованих плат можна використовувати знову, тоді як у випадку звичайним методам - тільки 2%.

Практично жодне підприємство не зможе самостійно утилізувати комп'ютери та оргтехніку, так як цей процес вимагає сучасного обладнання та специфічних знань. Тому довірити таку роботу можна тільки професіоналам, які мають великий досвід у даній сфері.

Проблема утилізації використаних комп'ютерів, периферійного обладнання, стає гострішою з кожним роком. Обсяги виробництва продуктів інформаційно-телекомунікаційних технологій та частота їх заміни на нові моделі примушують компанії замислюватись над проблемою біодеградації. Дослідники вважають, що необхідно дати більше стимулів як виробникам комп'ютерів, так і користувачам, щоб вони удосконалили і повторно використовували своє обладнання, а не викидали його. Успіхи в цій галузі допоможуть, серед іншого, компаніям-виробникам зменшити податки, котрі вони сплачують зараз за утилізацію застарілих моделей. Останнє тим більше важливо, оскільки робить екологізацію економічно вигідною, тож спрямовує у цю сферу дедалі більше зусиль дослідників та довгострокових капіталовкладень. Таким чином, подальше поширення інформаційних технологій не збільшить, а навпаки – зменшить техногенне навантаження на довкілля.

Висновок

Питання захисту навколишнього середовища в процесі виробництва комп'ютерів виникли давно і регламентуються зараз. В останні роки в усьому світі з'явилися численні нормативні акти і стандарти (міжнародні NPR чи TCO95, TCO 99), призначені зменшити негативні впливи.

Наприклад, стандартом ТСО-95 NUTEK контролюються викиди токсичних речовин, умови роботи й ін. Згідно з ТСО-95 вироблене устаткування може бути сертифіковане лише в тому випадку, якщо не тільки контрольовані параметри самого устаткування відповідають вимогам цього стандарту, але і технологія виробництва цього устаткування відповідає вимогам стандарту.

Виробник комп'ютера в наші дні звичайно рекламує свій товар як такий, що задовольняє декілька екологічних вимог. Наприклад:

- мале споживання електроенергії;
- екологічно чисте виробництво;
- не використання фреону, який руйнує озоновий шар Землі;
- виготовлення тари, документації й упакування з матеріалів вторинної обробки, і т.д.

Часто виробник називає такий комп'ютер "зеленим" ("Green PC"), хоча єдиного стандарту "зеленого" комп'ютера поки немає. Покупець, якщо він зацікавлений у збереженні власного здоров'я і здоров'я планети, повинний у процесі покупки цікавитися не тільки функціональними, але й екологічними характеристиками комп'ютера, що купується.

Комп'ютерні технології, будучи великим досягненням людства, можуть мати негативні наслідки для навколишнього світу. Технічний рівень сучасних моніторів не дозволяє цілком виключити існування шкідливих впливів, однак цей вплив необхідно мінімізувати, регламентувавши ряд параметрів. Для зниження збитку необхідне дотримання установлених вимог і норм. Основна мета їхнього впровадження – захист тваринного і рослинного світу від пагубного впливу комп'ютерної й іншої електронної техніки.

Організація «Грінпіс» уважно стежить за тим, як великі компанії відносяться до захисту навколишнього середовища, і регулярно публікує звіт Greenpeace Guide to Greener Electronics, в якому виробникам (HP, Sony, Toshiba і ін.) присвоюються рейтинги в трьох основних категоріях: раціональність операцій, енергія і клімат, екологічно чисті продукти. Можна зробити висновок, що залишається лише сподіватися, що настане час, коли

технології будуть допомагати людині, не завдаючи незворотної шкоди здоров'ю навколишнього середовища.

6. ОХОРОНА ПРАЦІ

Розробка інтелектуальної медичної системи на основі нейронних мереж RESNET передбачає роботу з різноманітним програмним забезпеченням. При цьому лікар працює на ЕОМ, яке знаходиться в приміщенні розміром 10х6х3 м. На вікнах знаходяться жалюзі. Температура повітря - 23°C, вологість повітря - 80%, рівень шуму 80дБ. Екран монітора знаходиться на рівні очей.

6.1 Аналіз умов праці на робочому місці

На працівника (лікаря), який виконує описану вище роботу можуть впливати такі небезпечні й шкідливі виробничі фактори:

- недостатня освітленість робочої зони;
- підвищений рівень електромагнітних випромінювань;
- підвищений рівень шуму на робочому місці;
- підвищена або знижена вологість повітря;
- нервово-психічні перевантаження.

Недостатня освітленість робочої зони.

В силу тісного взаємозв'язку зору людини з роботою її мозку, освітлення здійснює істотний вплив на центральну нервову систему, що керує всією життєдіяльністю людини.

Недостатнє освітлення робочих місць є однією з причин низької продуктивності роботи. При недостатньому освітленні, очі працюючого сильно напружені, при цьому ускладнюється розрізнення оточуючих приладів, знижується темп роботи і погіршується загальний стан. Рациональне освітлення сприяє підвищенню продуктивності і безпеки праці, а також збереженню здоров'я працівників.

КАФЕДРА АКІК

НАУ 20 19 32 000 ПЗ

<i>Розроб.</i>	<i>Хоцянівський В.П.</i>			<i>Охорона праці</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Керівник</i>	<i>Синеглазов В.М.</i>						
<i>Консульт.</i>	<i>.</i>				<i>151 204М</i>		
<i>Н.контр.</i>	<i>Тупіцин М.Ф.</i>						
<i>Зав.кафедри</i>	<i>Синеглазов В.М.</i>						

Недостатнє освітлення робочих місць може виникати з таких причин: особливе розташування сусідніх будівель, які можуть створювати затемнення робочої зони; забруднення та недостатня кількість або непрацездатність деяких чи всіх освітлювальних приладів; невірно підібрані чи замінені лампи в світильниках та інші.

Підвищений рівень електромагнітних випромінювань.

Джерелами випромінювання електромагнітних полів є ЕОМ, зокрема – електронно - променева трубка монітора, а також елементи блоків живлення системного модуля, монітора, принтера. Систематичний вплив на працівника, даного виду випромінювання, з рівнями, що перевищують допустимі (0,2 мкТл), призводить до порушення стану його здоров'я. При цьому можуть виникати зміни в нервовій, серцево-судинній та інших системах організму людини.

У разі впливу на організм електромагнітного поля значної інтенсивності можуть виникати ураження кристаликів очей, нервово-психічні захворювання та такі явища, як випадання волосся чи ламкість нігтів. Згідно досліджень, вже через дві години після початку роботи перед екраном у оператора з'являються і з часом посилюються болісні симптоми: головний біль, біль у м'язах шії, а також біль в очах.

Підвищений рівень шуму на робочому місці.

Тривалий вплив шуму на людину провокує такі явища, як зниження гостроти зору, слуху, підвищення кров'яного тиску, зниження уваги. Він також призводить до підвищеної втоми, загальної слабкості, дратівливості і т.д.

У розглянутому робочому приміщенні джерелами шуму можуть бути працюючий принтер або кондиціонер. Допустимий рівень шуму складає 60 дБ, в даному приміщенні він дорівнює 80 дБ.

Підвищена або знижена вологість повітря.

Як на організм людини, так і на ЕОМ великий вплив робить відносна вологість повітря. Якщо вона менше ніж 40 %, то основа магнітних дисків стає

більш крихкою, підвищується зношування магнітних головок, пошкоджується ізоляція проводів, а також виникає статична електрика при русі гнучкого магнітного диска в дисководі. При відносній вологості повітря більше 75%, знижується опір ізоляції, змінюються робочі характеристики елементів, зростає інтенсивність відмов елементів ЕОМ.

Нервово-психічні перевантаження

Нервово-психічні навантаження можуть мати прояв у зміні напруги зору (точність роботи), нервово-емоційного напруження і інтелектуального навантаження (обсяг обробленої інформації і т.д.); монотонності трудового процесу (різноманітність, темп роботи). Елементи цієї групи, за винятком монотонності, не мають затверджених нормативів.

Крім того, при експлуатації, ремонті й обслуговуванні систем стабілізації та керування інерційними об'єктами необхідно врахувати наступний шкідливий виробничий фактор, що входить у ГОСТ 12.0.003-74 "Небезпечні і шкідливі виробничі фактори. Класифікація".

При проведенні технічної експлуатації систем стабілізації, відповідно до регламенту, зв'язаного з демонтажем регулятора та перевірки його в лабораторії має місце такий шкідливий виробничий фактор, як підвищене значення напруги змінного струму електромережі 220 В.

При технічній експлуатації електрообладнання систем стабілізації електротравми можуть виникнути за такої причини, як безпосереднє доторкання до струмопровідних частин електроустаткування під напругою.

Ступінь впливу електричного струму на організм людини визначається опором тіла людини, значенням, родом і частотою струму, шляхом струму в організмі, станом організму і зовнішнім середовищем. Електротравми відносяться до небезпечних факторів, що впливають на весь організм.

6.2 Розробка заходів з охорони праці

Кабінети, обладнанні комп'ютерною технікою, повинні розміщуватись в окремих приміщеннях з природнім освітленням і організованим обміном

повітря. Орієнтація вікон повинна бути на північ або на північний схід, вікна повинні мати жалюзі, які можна регулювати, або штори. Площа на одного працюючого за ПК повинна складати не менше 6 м², об'єм - не менше 20 м³. Не рекомендується розміщувати кабінети обчислювальної техніки у підвальних приміщеннях будівлі.

Стіни, стеля, підлога та обладнання кабінетів з комп'ютерною технікою повинні мати покриття із матеріалів з матовою структурою з коефіцієнтом відбиття: стін – 40-50%, стелі - 70-80%, підлоги – 20-30%, предметів обладнання – 40-50% (робочого столу – 40-50%, корпусу дисплею та клавіатури – 30-50%, шаф та стелажів – 40-60%). Поверхня підлоги повинна мати антистатичне покриття та бути зручною для вологого прибирання.

Приміщення з ПК повинні мати природне та штучне освітлення. Штучне освітлення в приміщеннях з ПК повинно здійснюватись системою загального освітлення. Як джерела світла при штучному освітленні повинні застосовуватись люмінесцентні лампи.

Будівельно-акустичні методи захисту приміщень від шуму: звукоізоляція граничних конструкцій, ущільнення по периметру притворів вікон і дверей; звукопоглинаючі конструкції й екрани; глушники шуму. Крім того, рекомендується заміна матричних принтерів на лазерні.

Забороняється використовувати для оздоблення інтер'єру комп'ютерних приміщень полімерні матеріали (дерев'яно-стружкові плити, шпалери, що придатні для миття, плівкові та рулонні синтетичні матеріали, шаровий пластик та ін.), що виділяють у повітря шкідливі хімічні речовини, які перевищують гранично допустимі концентрації.

Основним засобом захисту від електромагнітних випромінювань, які присутні в обчислювальній техніці є екранування джерел випромінювання. Сучасні монітори і блоки живлення в більшості випадків мають корпус, виконаний із спеціального матеріалу, який практично повністю затримує електромагнітне випромінювання. Застосовуються також спеціальні екрани, які зменшують ступінь впливу електромагнітних променів на оператора.

Системний блок і монітор повинен знаходитися якомога далі від працівника: на відстані не менше 50-60 см.

Для зниження електромагнітного впливу на оператора використовуються також раціональні режими роботи (робота на ПК не повинна перевищувати 50% робочого часу). У зв'язку з тим, що електромагнітне випромінювання від стінок монітора набагато більше, рекомендоване розміщення комп'ютера в кутку кімнати.

Для встановлення нормального рівня вологості в приміщенні необхідне регулярне провітрювання. Оптимальні параметри повітряного середовища забезпечуються використанням опалення, вентиляції і кондиціонування повітря відповідно до СНіП 2.04.05-92 “Опалення, вентиляція і кондиціонування повітря”.

Основними заходами щодо зниження нервово-психічної напруженості є наступні: організація прийомів і методів роботи; оптимізація темпу роботи; оптимізація режиму роботи та відпочинку; науково-обґрунтоване встановлення норм обслуговування і норм часу його обслуговування з урахуванням обсягу інформації, яку працівник може правильно сприйняти, переробити і прийняти своєчасне і правильне рішення; черговість робіт, які вимагають участі різних аналізаторів (слуху, зору, дотику та ін); черговість робіт різної складності та інтенсивності; оптимізація режимів роботи і відпочинку; попередження і зниження монотонності роботи шляхом підвищення змістовності роботи та ін.

Вибір, розміщення, виконання, спосіб встановлення і клас ізоляції використовуваних машин, апаратів та іншого електрообладнання при налаштуванні систем стабілізації приводити відповідно з вимог державних стандартів ССБТ та “Правил побудови електроустаткування” ПУЕ – 86 та правил експлуатації електроустаткування відповідно до ДНАОП 0.00-4.03-93 “Правила безпечної експлуатації електроустаткування споживачів”, а також ОСТ 12.1.019-79 ССБТ “Електробезпека. Загальні вимоги безпеки.”

Усі припустимі норми по даному розділі вказані в ОСТ 12.1.038-82 "ССБТ. Електробезпека. Гранично припустимі рівні напруги дотику і струму".

6.3 Розрахунок захисного заземлення комп'ютера

Захисне заземлення є одним з найбільш ефективних заходів захисту від небезпеки ураження струмом у випадку дотику до металевих неструмоведучих частин електроустановок, які опинилися під напругою.

Метою розрахунку є визначення основних конструктивних параметрів заземлення (числа, розмірів, порядку розміщення вертикальних стрижнів і довжини сполучної смуги, що об'єднує їх в груповий заземлювач), при яких опір розтікання струму обраного групового заземлювача ($R_{гр}$) не перевищить нормативного значення ($R_{зн}$).

Грунт	Питомий опір ґрунту $\rho_{роз}$, Ом*м	Дані для вибору нормованого опору заземлювального пристрою	Тип	Діаметр заземлювача, d, м	Довжина заземлювача, l, м	Глибина закладення заземлювача, h, м	Відстань між заземлювачами, a, м	Ширина полоси, b, м
Пісок	400	U = 350В N = 560кВА	труба	0,05	3,0	0,7	6,0	0,02

Таблиця 7.1. Вихідні дані для розрахунку захисного заземлення

Обчислюємо опір розтікання струму одиночного вертикального заземлювача $R_в$. У разі стрижневого круглого перетину (трубчастого) заземлювача, заглибленого в землю, розрахункова формула має вигляд:

$$R_в = \frac{\rho_{роз}}{2\pi l} \left(\ln \frac{2l}{d} + \frac{1}{2} \ln \frac{4t+l}{4t-l} \right),$$

де $\rho_{роз}$ - розрахунковий питомий опір ґрунту;

l - довжина вертикального стрижня, м;

d - діаметр перетину, мм;

t - відстань від поверхні ґрунту до середини довжини вертикального стрижня, м;

$$t = \frac{l}{2} + 0.7 = \frac{3}{2} + 0.7 = 2.2$$

$$R_B = \frac{300}{2 \cdot 3.14 \cdot 3} \left(\ln \frac{2 \cdot 3}{0.05} + \frac{1}{2} \ln \frac{4 \cdot 2.2 + 3}{4 \cdot 2.2 - 3} \right) = 15.57 \text{ Ом}$$

Розрахуємо приблизну кількість вертикальних стрижнів:

$$n' = \frac{R_B}{R_H}$$

де R_B - опір розтікання струму одиночного вертикального заземлювача, Ом;

R_H - нормативний опір штучного заземлювача, Ом;

$$n' = \frac{15.57}{4} = 3.89 \approx 4$$

Так як $U = 350\text{В}$, а $N = 560\text{кВА}$ нормативний опір штучного заземлювача $R_H = 4\text{Ом}$.

Ефект екранування:

$$k = \frac{a}{l} = \frac{6}{3} = 2$$

Визначимо число одиночних заземлювачів з урахуванням коефіцієнта використання $k_{\text{вик}} = 0,8$

$$n = \frac{n'}{k_{\text{вик}}} = \frac{3.89}{0.8} = 4.86 \approx 5$$

Знайдемо довжину смуги L , що з'єднує заземлювачі:

$$L = a \cdot (n - 1)$$

$$L = 6 \cdot 4 = 24$$

Опір смугового заземлювача R_n , покладеного на глибину h від поверхні землі, так само:

$$R_n = \frac{\rho_{\text{роз}}}{2 \cdot \pi \cdot L} \left(\frac{\ln 2L^2}{bh} \right) = \frac{400}{2 \cdot 3.14 \cdot 24} \ln \frac{2 \cdot 24^2}{0.02 \cdot 0.7} = 30.03 \text{ Ом}$$

Загальний опір заземлюючого пристрою:

$$R_3 = \frac{R_b \cdot R_n}{n \cdot R_n \cdot k_{\text{ВИК}} + R_{\text{В}} \cdot 3_{\text{П}}} = \frac{30.03 \cdot 15.57}{5 \cdot 30.03 \cdot 0.8 \cdot 15.57 \cdot 1.1} = 3.41 \text{ Ом}$$

Таким чином, визначено основні конструктивні параметри заземлювача, при яких загальний опір обраного заземлювального пристрою ($R_3 = 3,41 \text{ Ом}$) не перевищує нормативного ($R_n = 4 \text{ Ом}$).

6.4 Пожежна безпека робочого приміщення

Відповідно до ГОСТ 12.1.004-91 “Заходи пожежної безпеки” пожежна та вибухова безпека забезпечується: системою запобігання пожежі системою протипожежного захисту та організаційно-технічними заходами.

Пожежна небезпека може бути обумовлена утворенням електричної дуги та розпеченими частинами металу, іскор, перегрівом струмопровідних елементів, запаленням легкозаймистих рідин та пароповітряних сумішей речовин, використаних при експлуатації та технічному обслуговуванні.

Заходи по забезпеченню пожежної та вибухової безпеки в лабораторії:

- 1) Для оздоблення стін слід використовувати негорючі або важко займисті матеріали;
- 2) Система електроживлення повинна мати блокування, яке забезпечує її відключення;
- 3) Використовувати заходи та засоби запобігання іскріння;
- 4) Для можливості виявлення пожежі необхідно використовувати датчики, які реагують при з’явленні диму, на підвищену температуру, реагуючі на відкрите полум’я;
- 5) Для гасіння пожежі використовувати вуглекислі вогнегасники та інші засоби.

Для гасіння пожежі в приміщенні використовуються переносні вуглекислі вогнегасники. Вогнегасники встановлюються в приміщенні з розрахунку один вогнегасник на 40-50 м² підлоги, але не менше двох.

Для попередження пожежі, вибуху через перевантаження і коротких замикань у блоці необхідно здійснити правильний вибір елементів електричних ланцюгів і забезпечити систему електрозахисту.

6.5. Інструкція з охорони праці під час роботи з інтелектуальною медичною системою на основі нейронних мереж RESNET

1. До обслуговування технічної системи допускаються особи інженерно-технічного складу, які вивчили пристрій, інструкції з експлуатації, ці інструкції та склавши залік по техніці безпеки та пожежній безпеці.
2. Технік, який буде проводити технічне обслуговування, повинний твердо знати та суворо виконувати правила внутрішнього розпорядку, не допускати на виробничому місці паління та розпивання спиртних напоїв.
3. Небезпечним виробничим фактором при дослідженні технічних систем в лабораторії є ураження електричним струмом.

Перед початком дослідження:

4. Переконатися в тому, що захисне заземлення надійне, вся КВА повинна бути заземлена;
5. Інструменти, що використовуються, повинні мати маркування і зберігатися в спеціальній шафі.

Перед виконанням робіт:

6. Надягнути та упорядкувати робочу одягу, одягнути головний убір та прибрати під нього волосся;
7. Провести огляд технічного стану робочого місця, прибрати з робочого місця все, що може заважати роботі;

8. Перевірити справність обладнання, приладів, інструмента, вентиляції, освітлення. Не працювати на обладнанні з простроченими рядками освітлення;
9. Вивісити попереджувальні таблички;
10. Ужити заходів попередження мимовільного включення апаратури;
11. Переконатися в наявності медичної аптечки з необхідним мінімумом медикаментів, вогнегасників, індивідуальних засобів захисту.

Під час роботи:

12. Включення і вимикання устаткування виконується тільки особою, яка знає правила ТБ;
13. При змінах параметрів необхідно визначити величину вимірюваного параметра, підібрати і вивчити інструкцію з експлуатації вимірювальної апаратури і підготувати до вимірів;
14. Перед дослідженнями прилади обов'язково заземлити;
15. Регульовані роботи виконуються на основі убрання на виконання робіт або по усному розпорядженні відповідального керівника робіт;

По закінченню роботи:

16. Зняти напругу з обслуговуючого блоку;
17. Упорядкувати виробниче місце;
18. Здати інструменти, КПА, засоби індивідуального захисту.

Вимоги до безпеки в аварійних випадках:

19. Відімкнути струм від обладнання;
20. Зупинити роботу;
21. З'явитись в розпорядження старшого зміни та доповісти про випадок.

За порушення чи невиконання цих вимог винні несуть відповідальність відповідно до чинного законодавства.

Висновок

Робота з інтелектуальною медичною системою на основі нейронних мереж RESNET передбачає широке використання ЕОМ та здійснення нетривалих робіт з налаштування таких систем. Тому було проаналізовано умови основних робочих місць проектувальника та розроблено заходи і рекомендації з охорони праці. При цьому, обов'язковим елементом стало формування заходів з пожежної безпеки. Оскільки налаштування системи стабілізації здійснюється поза межами звичного робочого місця проектувальника, було розроблено відповідну інструкцію з охорони праці.

ВИСНОВКИ

Для досягнення роботи було вирішено такі основні завдання:

1. Досліджено існуючі методики і апаратно програмні засоби оцінки та формування достовірної навчальної вибірки.
2. Розроблено структурно-функціональну схему для алгоритму контрастування зображення.
3. Розроблено моделі та засоби підтримки формування діагностичних висновків на основі технології загорткових нейронних мереж з урахуванням патогенетичних і симптоматичних чинників, пов'язаних з виявленням захворювання пацієнта.
4. Апаратно-програмна реалізація запропонованих алгоритмів, моделей і методів для поглибленого обстеження

Основні результати дослідження отримані за допомогою методів системного і статистичного аналізу, аналізу алгоритмів. Запропоновано моделі оцінки якості системи підтримки діагностики захворювань пацієнта і виявлена сукупність додаткових функціональних можливостей системи, що розробляється з використанням діаграм причин-результатів і часових графів як основи виключення суб'єктивних факторів недостовірності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Боровников В.П. Statistica для студентов и инженеров. – М.: Компьютер Пресс, 2011. – 301 с.
2. Котов Ю.Б. Новые математические подходы к задачам медицинской диагностики. - М.: Единоториал УРСС, 2004. -328 с.
3. URL: <http://www.science-education.ru/113-11527> (дата обращения: 18.02.2014).
4. Корневский Н.А., Рябкова Е.Б. Метод синтеза нечетких решающих правил по информации о геометрической структуре многомерных данных // Вестник Воронежского государственного технического университета 2011г. - Т.7, №8. - С.128-137.
5. Жуковский А. Е. Синтез обучающей выборки на основе реальных данных в задачах распознавания изображений / А. Е. Жуковский, С. А. Усилин, Н. А. Тарасова, Д. П. Николаев // Информационные технологии и системы (ИТиС'12): сборник трудов конференции. – М., 2012. – С. 377–382.
6. Ахметов Б. С. Морфинг-размножение примеров родителей в нескольких поколениях примеров потомков / Б. С. Ахметов, А. И. Иванов, А. Ю. Малыгин, С. В. Качалин, Н. А. Сейлова // Инженерное образование и наука в XXI веке: Проблемы и перспективы: Тр. междунар. форума посвящ. 80-летию КазНТУ им. К. И. Сатпаева. – Алматы : КазНТУ, 2014. – Т. 2. – С. 200–203.
7. Качалин С. В. Повышение устойчивости обучения больших нейронных сетей дополнением малых обучающих выборок примеров-родителей, синтезированными биометрическими примерами-потомками /Качалин С. В. // Труды научно-технической конференции кластера пензенских предприятий, обеспечивающих безопасность информационных технологий. – Пенза-2014. – Т. 9. – С. 32–35.

8. Донских А. О. , Сирота А. А. МЕТОД ИСКУССТВЕННОГО РАЗМНОЖЕНИЯ ДАННЫХ В ЗАДАЧАХ МАШИННОГО ОБУЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ НЕПАРАМЕТРИЧЕСКИХ ЯДЕРНЫХ ОЦЕНОК ПЛОТНОСТИ РАСПРЕДЕЛЕНИЯ ВЕРОЯТНОСТЕЙ. ВЕСТНИК ВГУ, СЕРИЯ: СИСТЕМНЫЙ АНАЛИЗ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, 2017, № 3.- С.142- 155
9. Мангалова, Е. Прогнозирование мощности ветряных электростанций на основе непараметрического алгоритма к ближайших соседей / Е. Мангалова, И. Петрунькина // Доклады всероссийской научной конференции АИСТ'2013. – 2013 – С. 1–8.
10. Canavet, O. Efficient sample mining for object detection. / O. Canavet, F. Fleuret // Proceedings of the Asian Conference on Machine Learning (ACML). – 2014 – P. 48–63.
11. Dropout: A simple way to prevent neural networks from overfitting / N. Srivastava, G.E. Hinton, A. Krizhevsky et al. // The Journal of Machine Learning Research. – 2014 – Vol. 15, no. 1. – P. 1929–1958.
12. Amazon Mechanical Turk. – <https://www.mturk.com/mturk/welcome>.
13. Гарсия Глория Буэно. Обработка изображений с помощью OpenCV Г М.: ДМК Пресс, 2015. 387 с.
14. <https://www.medicinenet.com/script/main/hp.asp>
15. https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf
16. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. F. M. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, “Top 10 algorithms in data mining,”*Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.
17. Q. Yang and X. Wu, “10 challenging problems in data mining research,” *International Journal of Information Technology and Decision Making*, vol. 5, no. 4, pp. 597–604, 2006.

- 18.G. P. C. Fung, J. X. Yu, H. Lu, and P. S. Yu, "Text classification without negative examples revisit," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 6–20, 2006.
- 19.H. Al-Mubaid and S. A. Umair, "A new text categorization technique using distributional clustering and learning logic," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 9, pp. 1156–1165, 2006.
- 20.K. Sarinnapakorn and M. Kubat, "Combining subclassifiers in text categorization: A dst-based solution and a case study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1638–1651, 2007.
- 21.W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, Oregon, USA, June 2007, pp. 193–200.
- 22.S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for wifi-based indoor localization," in *Proceedings of the Workshop on Transfer Learning for Complex Task of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, Illinois, USA, July 2008.
- 23.J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, 2007, pp. 432–439.
- 24.J. Ramon, K. Driessens, and T. Croonenborghs, "Transfer learning in reinforcement learning problems through partial policy recycling," in *ECML '07: Proceedings of the 18th European conference on Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 699–707.
- 25.M. E. Taylor and P. Stone, "Cross-domain transfer for reinforcement learning," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 879–886.
- 26.X. Yin, J. Han, J. Yang, and P. S. Yu, "Efficient classification across multiple database relations: A crossmine approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 770–783, 2006.

- 27.L. I. Kuncheva and J. J. Rodríguez, “Classifier ensembles with a random linear oracle,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 4, pp. 500–508, 2007.
- 28.E. Baralis, S. Chiusano, and P. Garza, “A lazy approach to associative classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 156–171, 2008.
- 29.X. Zhu, “Semi-supervised learning literature survey,” University of Wisconsin–Madison, Tech. Rep. 1530, 2006
- 30.K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, “Text classification from labeled and unlabeled documents using em,” *Machine Learning*, vol. 39, no. 2-3, pp. 103–134, 2000.
- 31.A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998, pp. 92–100.
- 32.T. Joachims, “Transductive inference for text classification using support vector machines,” in *Proceedings of Sixteenth International Conference on Machine Learning*, 1999, pp. 825–830.
- 33.X. Zhu and X. Wu, “Class noise handling for effective cost-sensitive learning by cost-guided iterative classification filtering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1435–1440, 2006.
- 34.S. Thrun and L. Pratt, Eds., *Learning to learn*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.
- 35.R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28(1), pp. 41–75, 1997.
- 36.R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, “Self-taught learning: Transfer learning from unlabeled data,” in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, Oregon, USA, June 2007, pp. 759–766.
- 37.H. Daumé III and D. Marcu, “Domain adaptation for statistical classifiers,” *Journal of Artificial Intelligence Research*, vol. 26, pp. 101–126, 2006.

- 38.B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in Proceedings of the 21st International Conference on Machine Learning, Banff, Alberta, Canada, July 2004.
- 39.H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, pp. 227–244, 2000.
- 40.W. Dai, Q. Yang, G. Xue, and Y. Yu, "Self-taught clustering," in Proceedings of the 25th International Conference of Machine Learning. ACM, July 2008, pp. 200–207.
- 41.Z. Wang, Y. Song, and C. Zhang, "Transferred dimensionality reduction," in Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008. Antwerp, Belgium: Springer, September 2008, pp. 550–565.
- 42.W. Dai, G. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in Proceedings of the 22rd AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada, July 2007, pp. 540–545.
- 43.J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.
- 44.J. Jiang and C. Zhai, "Instance weighting for domain adaptation in nlp," in Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 264–271.
- 45.X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in Proceedings of the 21st International Conference on Machine Learning, Bonn, Germany, August 2005, pp. 505–512.
- 46.J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in Proceedings of the 19th Annual Conference on Neural Information Processing Systems, 2007.

- 47.S. Bickel, M. Brückner, and T. Scheffer, “Discriminative learning for differing training and test distributions,” in Proceedings of the 24th international conference on Machine learning. New York, NY, USA: ACM, 2007, pp. 81–88.
- 48.M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, “Direct importance estimation with model selection and its application to covariate shift adaptation,” in Proceedings of the 20th Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 2008.
- 49.W. Fan, I. Davidson, B. Zadrozny, and P. S. Yu, “An improved categorization of classifier’s sensitivity on sample selection bias,” in Proceedings of the 5th IEEE International Conference on Data Mining, 2005.
- 50.W. Dai, G. Xue, Q. Yang, and Y. Yu, “Co-clustering based classification for out-of-domain documents,” in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 2007.
51. R. K. Ando and T. Zhang, “A high-performance semi-supervised learning method for text chunking,” in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 2005, pp. 1–9.
- 52.J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” in Proceedings of the Conference on Empirical Methods in Natural Language, Sydney, Australia, July 2006, pp. 120–128.
53. H. Daumé III, “Frustratingly easy domain adaptation,” in Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, June 2007, pp. 256–263.
54. A. Argyriou, T. Evgeniou, and M. Pontil, “Multi-task feature learning,” in Proceedings of the 19th Annual Conference on Neural Information

- Processing Systems, Vancouver, British Columbia, Canada, December 2007, pp. 41–48.
- 55.A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying, “A spectral regularization framework for multi-task structure learning,” in Proceedings of the 20th Annual Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2008, pp. 25–32.
- 56.S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, “Learning a meta-level prior for feature relevance from multiple related tasks,” in Proceedings of the 24th International Conference on Machine Learning. Corvallis, Oregon, USA: ACM, July 2007, pp. 489–496.
- 57.T. Jebara, “Multi-task feature and kernel selection for svms,” in Proceedings of the 21st International Conference on Machine Learning. Banff, Alberta, Canada: ACM, July 2004.
- 58.C. Wang and S. Mahadevan, “Manifold alignment using procrustes analysis,” in Proceedings of the 25th International Conference on Machine learning. Helsinki, Finland: ACM, July 2008, pp. 1120–1127.
59. N. D. Lawrence and J. C. Platt, “Learning to learn with the informative vector machine,” in Proceedings of the 21st International Conference on Machine Learning. Banff, Alberta, Canada: ACM, July 2004.
- 60.E. Bonilla, K. M. Chai, and C. Williams, “Multi-task gaussian process prediction,” in Proceedings of the 20th Annual Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2008, pp. 153–160.
- 61.A. Schwaighofer, V. Tresp, and K. Yu, “Learning gaussian process kernels via hierarchical bayes,” in Proceedings of the 17th Annual Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2005, pp. 1209–1216.
62. T. Evgeniou and M. Pontil, “Regularized multi-task learning,” in Proceedings of the 10th ACM SIGKDD International Conference on

- Knowledge Discovery and Data Mining. Seattle, Washington, USA: ACM, August 2004, pp. 109–117.
63. J. Gao, W. Fan, J. Jiang, and J. Han, “Knowledge transfer via multiple model local structure mapping,” in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas, Nevada: ACM, August 2008, pp. 283–291.
 64. L. Mihalkova, T. Huynh, and R. J. Mooney, “Mapping and revising markov logic networks for transfer learning,” in Proceedings of the 22nd AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada, July 2007, pp. 608–614.
 65. L. Mihalkova and R. J. Mooney, “Transfer learning by mapping with minimal target data,” in Proceedings of the AAAI-2008 Workshop on Transfer Learning for Complex Tasks, Chicago, Illinois, USA, July 2008.
 66. J. Davis and P. Domingos, “Deep transfer via second-order markov logic,” in Proceedings of the AAAI-2008 Workshop on Transfer Learning for Complex Tasks, Chicago, Illinois, USA, July 2008.
 67. P. Wu and T. G. Dietterich, “Improving svm accuracy by training on auxiliary data sources,” in Proceedings of the 21st International Conference on Machine Learning. Banff, Alberta, Canada: ACM, July 2004.
 68. M. Richardson and P. Domingos, “Markov logic networks,” *Machine Learning Journal*, vol. 62, no. 1-2, pp. 107–136, 2006.
 69. S. Ramachandran and R. J. Mooney, “Theory refinement of bayesian networks with hidden variables,” in Proceedings of the 14th International Conference on Machine Learning, Madison, Wisconsin, USA, July 1998, pp. 454–462.
 70. A. Arnold, R. Nallapati, and W. W. Cohen, “A comparative study of methods for transductive transfer learning,” in Proceedings of the 7th IEEE International Conference on Data Mining Workshops. Washington, DC, USA: IEEE Computer Society, 2007, pp. 77–82.

- 71.T. Joachims, “Transductive inference for text classification using support vector machines,” in Proceedings of the Sixteenth International Conference on Machine Learning, San Francisco, CA, USA, 1999, pp. 200–209.
- 72.S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of representations for domain adaptation,” in Proceedings of the 20th Annual Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2007, pp. 137–144.
- 73.J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, “Learning bounds for domain adaptation,” in Proceedings of the 21st Annual Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2008, pp. 129–136.
- 74.D. Xing, W. Dai, G.-R. Xue, and Y. Yu, “Bridged refinement for transfer learning,” in 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, ser. Lecture Notes in Computer Science. Warsaw, Poland: Springer, September 2007, pp. 324–335.
- 75.X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, “Spectral domain transfer learning,” in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas, Nevada: ACM, August 2008, pp. 488–496.
- 76.G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, “Topic-bridged pls for cross-domain text classification,” in Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Singapore: ACM, July 2008, pp. 627–634.
- 77.S. J. Pan, J. T. Kwok, and Q. Yang, “Transfer learning via dimensionality reduction,” in Proceedings of the 23rd AAAI Conference on Artificial Intelligence, Chicago, Illinois, USA, July 2008, pp. 677–682.
- 78.S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” in Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, 2009.

79. M. M. H. Mahmud and S. R. Ray, "Transfer learning using kolmogorov complexity: Basic theory and empirical evaluations," in Proceedings of the 20th Annual Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2008, pp. 985–992.
80. E. Eaton, M. desJardins, and T. Lane, "Modeling transfer relationships between learning tasks for improved inductive transfer," in Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008, ser. Lecture Notes in Computer Science. Antwerp, Belgium: Springer, September 2008, pp. 317–332.
81. M. T. Rosenstein, Z. Marx, and L. P. Kaelbling, "To transfer or not to transfer," in a NIPS-05 Workshop on Inductive Transfer: 10 Years Later, December 2005.
82. S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in Proceedings of the Sixteenth Annual Conference on Learning Theory. San Francisco: Morgan Kaufmann, 2003, pp. 825–830.
83. B. Bakker and T. Heskes, "Task clustering and gating for bayesian multitask learning," Journal of Machine Learning Research, vol. 4, pp. 83–99, 2003.
84. A. Argyriou, A. Maurer, and M. Pontil, "An algorithm for transfer learning in a heterogeneous environment," in Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008, ser. Lecture Notes in Computer Science. Antwerp, Belgium: Springer, September 2008, pp. 71–85.
85. R. Raina, A. Y. Ng, and D. Koller, "Constructing informative priors using transfer learning." in Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, Pennsylvania, USA, June 2006, pp. 713–720.
86. J. Yin, Q. Yang, and L. M. Ni, "Adaptive temporal radio maps for indoor location estimation," in Proceedings of the 3rd IEEE International

- Conference on Pervasive Computing and Communications, Kauai Island, Hawaii, USA, March 2005.
- 87.S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan, “Adaptive localization in a dynamic WiFi environment through multi-view learning,” in Proceedings of the 22nd AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada, July 2007, pp. 1108–1113.
- 88.[V. W. Zheng, Q. Yang, W. Xiang, and D. Shen, “Transferring localization models over time,” in Proceedings of the 23rd AAAI Conference on Artificial Intelligence, Chicago, Illinois, USA, July 2008, pp. 1421–1426.
- 89.S. J. Pan, D. Shen, Q. Yang, and J. T. Kwok, “Transferring localization models across space,” in Proceedings of the 23rd AAAI Conference on Artificial Intelligence, Chicago, Illinois, USA, July 2008, pp. 1383–1388.
- 90.V. W. Zheng, S. J. Pan, Q. Yang, and J. J. Pan, “Transferring multi-device localization models using latent multi-task learning,” in Proceedings of the 23rd AAAI Conference on Artificial Intelligence, Chicago, Illinois, USA, July 2008, pp. 1427–1432.
- 91.H. Zhuo, Q. Yang, D. H. Hu, and L. Li, “Transferring knowledge from another domain for learning action models,” in Proceedings of 10th Pacific Rim International Conference on Artificial Intelligence, December 2008
- 92.V. C. Raykar, B. Krishnapuram, J. Bi, M. Dundar, and R. B. Rao, “Bayesian multiple instance learning: automatic feature selection and inductive transfer,” in Proceedings of the 25th International Conference on Machine learning. Helsinki, Finland: ACM, July 2008, pp. 808–815.
- 93.X. Ling, G.-R. Xue, W. Dai, Y. Jiang, Q. Yang, and Y. Yu, “Can chinese web pages be classified with english data source?” in Proceedings of the 17th International Conference on World Wide Web. Beijing, China: ACM, April 2008, pp. 969–978.
- 94.Q. Yang, S. J. Pan, and V. W. Zheng, “Estimating location using Wi-Fi,” *IEEE Intelligent Systems*, vol. 23, no. 1, pp. 8–13, 2008.

- 95.X. Shi, W. Fan, and J. Ren, “Actively transfer domain knowledge,” in Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008, ser. Lecture Notes in Computer Science. Antwerp, Belgium: Springer, September 2008, pp. 342–357.
- 96.G. Kuhlmann and P. Stone, “Graph-based domain mapping for transfer learning in general games,” in 18th European Conference on Machine Learning, ser. Lecture Notes in Computer Science. Warsaw, Poland: Springer, September 2007, pp. 188–200.
- 97.W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, “Translated learning,” in Proceedings of 21st Annual Conference on Neural Information Processing Systems, 2008.
- 98.B. Li, Q. Yang, and X. Xue, “Transfer learning for collaborative filtering via a rating-matrix generative model,” in Proceedings of the 26th International Conference on Machine Learning, Montreal, Quebec, Canada, June 2009.
- 99.B. Li, Q. Yang, and X. Xue, “Can movies and books collaborate? - crossdomain collaborative filtering for sparsity reduction,” in Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 2009.
100. Гилой В. Интерактивная машинная графика: структуры данных, алгоритмы, языки: Пер. с англ. – М.: Мир, 1981. – 384 с.
101. Васюхин М.И. Алгоритмические и программно-аппаратные методы и средства построения интерактивных геоинформационных комплексов оперативного взаимодействия: Дис... д-ра техн. наук: 05.13.13 / Институт кибернетики НАН Украины. – К., 2002. – 414 с.
102. Васюхин М.И., Зорич И.С. Отображение динамики единичного видеосимвола в реальном времени // УСИМ. – 1997. – № 1/3. – С. 87 – 92.
103. Смолий В.В. Применение конформных отображений в процессе геометрических преобразований изображений динамических объектов

- // Наукові праці Донецького державного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. – Донецьк: ДонДТУ, 2000. – Вип. 15. – С. 150 – 155.
104. Rvacheva O.V., Strjukova A.O., Chmutin A.M. Jarkostnoe preobrazovanie i svetovoj sdvig: problemy Microsoft, Corel, Adobe, etc. // Aktual'nye problemy gumanitarnyh i estestvennyh nauk. 2010. #10. S. 32-38. Rezhim dostupa: <http://www.elibrary.ru/item.asp?id=15606653> (data obrashhenija: 03.08.2019)
105. Бондарь О.В., Чмутин А.М. Насыщенность цвета, контраст насыщенностей и переносимая ими изобразительная информация. // Современные научные исследования и инновации. 2018. № 3. 15 с. [Электронный ресурс]. – Режим доступа: <http://web.snauka.ru/issues/2018/03/85975> (дата обращения: 03.08.2019).
106. K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. Microsoft Research. Proceedings of Computer Vision and Pattern Recognition Conference (CVPR 2015), Boston, June 8-10, 2015, Pp. 770-778.
107. Бейдер Дэн. Чистый Python. Тонкости программирования для профи. — "Издательский дом ""Питер""", 2018. — С. 64—65. — 288 с. — ISBN 978-5-4461-0803-9
108. Using Python on Windows — Python 3.5.9 documentation (англ.). Python Documentation. Python Software Foundation.
109. Cybenko, G.V. (2006). Approximation by Superpositions of a Sigmoidal function. У van Schuppen, Jan H. Mathematics of Control, Signals, and Systems[en]. Springer International. с. 303–314.
110. Персональні суперкомп'ютери: архітектура, проектування, застосування: монографія / А. О. Мельник, В. А. Мельник ; М-во освіти і науки, молоді та спорту України, Нац. ун-т «Львів.

- політехніка». — Л. : Вид-во Львів. політехніки, 2013. — 516 с. : іл. —
Бібліогр.: с. 496—513
111. <https://tcocertified.com/>
112. Кондратьев В. В. Организация энергосбережения (энергомeneджмент). Решения ЗСМК - НКМК - НТМК - ЕВРАЗ. — М.: Инфра-М, 2011. — 108 с..
113. Желібо Є. П., Заверуха Н. М., Зацарний В. В. Безпека життєдіяльності. Навч. посіб. / за ред. Є. П. Желібо. 6-е вид. —К.: "Каравела", 2009 ,
114. В.І. Саранчук, М.О.Ільяшов, В.В. Ошовський, В.С.Білецький. Хімія і фізика горючих копалин. - Донецьк: Східний видавничий дім, 2008. – с. 600
115. <https://www.greenpeace.org/international/>
116. S. Darby, D. Hill, R. Doll (2001). Radon: A likely carcinogen at all exposures. Annals of Oncology
117. Програма ООН з навколишнього природного середовища // Словник – довідник з екології : навч.-метод. посіб. / уклад. О. Г. Лановенко, О. О. Остапішина. — Херсон : ПП Вишемирський В.С., 2013. — С. 149.
118. НЕБЕЗПЕЧНИЙ ВПЛИВ ВИРОБНИЦТВА КОМП'ЮТЕРНОЇ ТЕХНІКИ НА НАВКОЛИШНЄ СЕРЕДОВИЩЕ Невжинський К. В.
119. Микросхемы вредны для окружающей среды. [Електронний ресурс]. Режим доступу: URL: <https://www.osp.ru/cw/2002/47/59385/>.
120. Экология производства компьютеров. [Електронний ресурс]. Режим доступу: URL: <https://polymus.ru/ru/pop-science/blogs/channels/himiyabuduschego/15476/>.
121. <https://www.epa.gov/>
122. https://zakon.rada.gov.ua/laws/show/995_022#Text
123. К. Хе, Х. Чжан, С. Рен и Дж. Сун, «Глубокое остаточное обучение для распознавания изображений», в CVPR, 2016