

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Т.в.о. завідувача кафедри
_____ Сафонова С.О.
« ____ » _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Дослідження методів класифікації зображень та реалізація методу опорних
векторів

Освітньо-кваліфікаційний рівень “Магістр”
Спеціальність 123 “Комп’ютерна інженерія”

Науковий керівник роботи:

(підпис)

С.О. Сафонова

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О. Критська

(ініціали, прізвище)

Студент:

(підпис)

В.В. Білов

(ініціали, прізвище)

Група:

КІ-18дм

Сєвєродонецьк 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра комп'ютерних наук та інженерії
Освітньо-кваліфікаційний
рівень “магістр”
Спеціальність 123 – “Комп'ютерна інженерія”
(шифр і назва)
Спеціалізація _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о.зав. кафедри КНІ
д.т.н., доц. С.О. Сафонова
« _____ » _____ 20 ____ р.

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Білову Владиславу Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів класифікації зображень та реалізація методу опорних векторів

керівник проекту (роботи) Сафонова С.О., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «11» 10 2019 р. № _____

2. Строк подання студентом роботи 15.01.2020

3. Вихідні дані до роботи Матеріали науково-дослідної практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

1. Огляд методів класифікації зображення та аналіз їх можливостей.
2. Огляд базової математичної моделі класифікації та способів опису об'єктів.
3. Математична модель методу опорних векторів, правила побудови вектора для лінійних та нелінійних випадків. Огляд функцій ядра.
4. Аналіз інваріантних моментів зображення.
5. Розробка програмного додатку класифікації зображень, порівняння результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	<i>Критська Я.О.</i>		

7. Дата видачі завдання 06.09.2020

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання до магістерської роботи	10.09.19-15.09.19	
2	Аналіз завдання, огляд літератури	16.09.19-25.09.19	
3	Аналіз технічних засобів	26.09.19-15.10.19	
4	Розробка методу	16.10.19-06.11.19	
5	Програмна реалізація	17.11.19-06.12.19	
6	Охорона праці	07.12.19-15.12.19	
7	Оформлення пояснювальної записки	16.12.19-28.12.19	
8	Підготовка презентації та доповіді	03.01.20-13.01.20	

Студент

_____ (підпис)

Білов В.В.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Сафонова С.О.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Білов В.В. Дослідження методів класифікації зображень та реалізація методу опорних векторів.

Метою магістерської роботи є дослідження питання класифікації інваріантів зображень за допомогою методу машинного навчання SVM (метод опорних векторів).

Результатом роботи є підготовлена тренувальна база та програмна реалізація методу.

Ключові слова: класифікація, інваріанти зображень, машинне навчання, SVM, метод опорних векторів, функції ядра.

АННОТАЦИЯ

Белов В.В. Исследование методов классификации изображений и реализация метода опорных векторов.

Целью магистерской работы является исследование вопроса классификации инвариантов изображений с помощью метода машинного обучения SVM (метод опорных векторов).

Результатом работы является подготовленная тренировочная база и программная реализация метода.

Ключевые слова: классификация, инварианты изображений, машинное обучение, SVM, метод опорных векторов, функции ядра.

ABSTRACT

Belov V.V. Investigation of image classification methods and implementation of the reference vector method.

The aim of the work is to study of image moments classification using a machine learning method SVM (support vector machine).

The result of the work is a prepared training base and software implementation of the method.

Key words: classification, image moment invariants, machine learning, SVM, support vector machine, kernel functions.

ЗМІСТ

Перелік умовних позначень.....	7
Вступ.....	8
1 Сучасний стан питання класифікації зображень	9
1.1 Класифікація як важливий етап обробки зображень	9
1.2 Методи класифікації елементів зображень.....	10
1.3 Машинне навчання у методах класифікації	14
1.3.1 Нейронні мережі в задачах обробки зображень	18
1.3.2 Лінійні методи класифікації	22
1.4 Постановка задачі дослідження	22
2 Математичні моделі класифікації та методу опорних векторів.....	23
2.1 Огляд розпізнавання та класифікації	23
2.1.1 Поняття розпізнавання та її математична модель	23
2.1.2 Поняття класифікації та її математична модель	27
2.1.3 Вид правил класифікації	29
2.1.4 Способи опису об`єктів.....	30
2.2 Принцип побудови опорних векторів	33
2.3 Складність навчання методу. Порівняння результатів.....	43
2.4 Побудова опорних векторів для нелінійно роздільних класів.....	44
2.5 Моментні інваріанти	52
3 Комп'ютерна модель класифікації зображень методом SVM.....	55
3.1 Вибір мови програмування та середовища розробки	55
3.2 Схематична структура комп'ютерної моделі	57
3.3 Ілюстрація роботи програми	58
3.4 Результати роботи системи класифікації зображень	62
4 Охорона праці та безпека в надзвичайних ситуаціях.....	65
4.1 Загальні питання з охорони праці.....	65
4.2 Аналіз стану умов праці	66
4.2.1 Вимоги до приміщень.....	66
4.2.2 Вимоги до організації місця праці	67
4.2.3 Навантаження та напруженість процесу праці	67
4.3 Виробнича санітарія.....	68

4.3.1	Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу	68
4.3.2	Пожежна безпека	70
4.3.3	Електробезпека.....	70
4.4	Гігієнічні вимоги до параметрів виробничого середовища	71
4.4.1	Мікроклімат.....	71
4.4.2	Освітлення	71
4.4.3	Шум та вібрація, електромагнітне випромінювання.....	73
4.5	Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій.....	73
4.6	Екологія.....	76
4.7	Висновки до розділу 4	77
	Висновки	78
	Перелік джерел посилань.....	79
	Додаток А Лістинг «Svm.py»	82
	Додаток Б Лістинг «Test_svm.py».....	87
	Додаток В Електронна презентація	89

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SVM – Support Vector Machine

НМ – нейронна мережа

ІЗ – інваріанти зображень

TF – Tensor Flow

ВСТУП

Безперервне зростання об'ємів інформації, що зберігається і обробляється, створює гостру необхідність створення методів і систем розпізнавання зображень. Ключову роль в цьому процесі відіграє класифікація зображень. В даний час розроблена величезна кількість методів і алгоритмів класифікації зображень, значну частину яких займають методи машинного навчання. Кожен з них має право на існування і деяку певну область застосування, яка залежить від характерних ознак, класифікаторів, тренувального набору, вимог до об'ємів обчислень і швидкості ухвалення рішень.

Класифікація зображень – дуже трудомісткий і, в загальному вигляді, до кінця не алгоритмізований для довільних зображень процес. Різні умови отримання даних, побічні чинники – це далеко не повний перелік причин, що переводять вирішення задач порівняння в розряд неординарних.

Роботи в області обробки, сегментації, класифікації та пошуку об'єктів на зображеннях велися і ведуться досить інтенсивно як вітчизняними, так і зарубіжними вченими. До теперішнього часу розроблено безліч підходів до класифікації зображення, але жоден з них не став універсальним. Залежно від характерних ознак та тренувального набору має сенс застосовувати різні класифікатори й відповідні методи. Причому застосовність конкретного способу для зображень певного типу досліджується, в основному, експериментально, і, як правило, обраний спосіб має досить вузьку спеціалізацією і може бути непридатним для зображень, що відхиляються від “типових випадків”.

1 СУЧАСНИЙ СТАН ПИТАННЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

1.1 Класифікація як важливий етап обробки зображень

Автоматична класифікація зображень розглядається як одна з найбільш важливих задач в області комп'ютерного зору, розпізнавання образів, пошуку інформації, комп'ютерної діагностики і т.д. Як правило, процес класифікації зображень складається з чотирьох основних етапів: збір даних, попередня обробка даних, витяг ознак, класифікація зображень. Насправді, класифікація зображень відноситься до процесу класифікації нового екземпляра або зображення в зумовленому класі або категорії зображення на основі набору навчальних матеріалів. Процес класифікації зображень зазвичай виконується за допомогою алгоритмів класифікації:

- лінійні класифікатори (логістична регресія, наївний баєсів класифікатор);
- метод опорних векторів;
- дерево рішень;
- штучна нейронна мережа;
- метод k-найближчих сусідів.

Етап вилучення функцій пов'язаний з проблемою отримання набору функцій, які містять достатню інформацію про класи і які забезпечують відповідні характеристики вмісту зображення. Існує великий список методів отримання об'єктів, серед яких: дескриптори Фур'є, масштаб-інваріантна функція, гістограма орієнтованих градієнтів, локальні бінарні шаблони і т. д.

В останні десятиліття мотиви зображення і моментальні інваріанти привертали увагу інженерно-наукової спільноти і широко застосовуються в різних областях, таких як аналіз зображень, класифікація зображень і розпізнавання образів. Це пов'язано з їх здатністю витягувати фігурні елементи незалежно від геометричних перетворень. Історично Ху в 1962 році представив набір геометричних інваріантів моменту, які використовуються, зокрема, в розпізнаванні образів. Проте представлений набір не дає опису повного аспекту зображення, оскільки їх базисна функції не ортогональна. Щоб подолати цю проблему, вчені запропонували використовувати ортогональні поліноми як базисну функцію для визначення нової безлічі ортогональних моментів, наприклад Лежандра, Зернике, Чебишева-Фур'є, Фур'є-Мелліна моменти. Ці моменти можуть використовуватися як дескриптори інваріанта зображення, використовуючи відповідну процедуру нормалізації, щоб отримати інваріанти трансляції, масштабу і обертання.

Нещодавно був введений новий набір дискретних моментів, які називаються сепарабельними моментами. По-перше, були представлені нові двовимірні дискретні і безперервні ортогональні поліноми, щоб визначити нові безлічі відокремлюваних двовимірних ортогональних моментів для аналізу 2D-зображення. Далі був знайдений новий набір відокремлюваних ортогональних моментів для класифікації 2D-зображень, заснований на тензорному творі многочленів Мейкснера. Фактично всі ці приведені роботи зосереджені на генерації сепарабельних інваріантів моментів з двовимірних многочленів, визначених тільки на рівномірній решітці.

З цього можна зробити висновок, що ця тема дуже актуальна в колі науковців, що займаються обробкою зображень, бо класифікація є одним з найважливіх етапів обробки зображень.

1.2 Методи класифікації елементів зображень

Так як задачі розпізнавання об'єктів заключаються в класифікації зображень на основі певних критеріїв, то важливим етапом є вибір оптимального класифікатора. Серед існуючих методів класифікації можна виділити наступні:

- ймовірностний критерій якості класифікації;
- оптимальна стратегія статистичної класифікації;
- класифікатор Байєса;
- мінімакський класифікатор;
- класифікатор Неймана-Пірсона.

Класифікатор Байєса є найбільш широко розповсюдженим з перелічених класифікаторів і застосовується при наявності повної апіорної інформації про класи, тобто коли відомі функція правдоподібності для кожного з класів, матриця штрафів, апіорні ймовірності для кожного з класів. Класифікатор Байєса ґрунтується на основі принципу максимуму апостеріорної ймовірності, що базується на трьох гіпотезах:

а) множина $X \times Y$ є імовірнісним простором з імовірнісною мірою p . Прецеденти $(x_1, y_1) \dots (x_n, y_n)$ з'являються випадково і незалежно у відповідності з розподілом p ;

б) відомі щільності розподілу класів $p_y(x) = p(x | K_y), y \in Y$, що називаються функціями правдоподібності;

в) відомі ймовірності появи об'єктів кожного з класів $P_y = P(K_y), y \in Y$, що називаються апіорними ймовірностями.

Базуючись на даних гіпотезах, принцип максимуму апостеріорної ймовірності записується в наступному вигляді:

$$F(x) = \arg \max_{y \in Y} P(K_y | x) = \arg \max_{y \in Y} p_y(x) P_y. \quad (1.1)$$

Доведено, що такий вибір вирішального правила є оптимальним з погляду мінімізації загального ризику. Основна проблема, полягає в тому, що на практиці гіпотези “б” і “в” майже ніколи не виконуються. Спроби оцінити ці функції розподілу по навчальній вибірці могли б привести до деякого результату, якби не погана обумовленість задачі, що приводить до вироджених рішень.

Існують системи виявлення об'єктів зображення, що базуються на “наївному” баєсовому методі. Даний метод ґрунтується на побудові емпіричної щільності розподілу ймовірностей класів по навчальній вибірці за припущення про незалежність компонентів вектора ознак. Практичні результати є наступними:

- алгоритм побудови “наївного” баєсового класифікатора схильний до перенавчання;
- алгоритм побудови “наївного” баєсового класифікатора чутливий до шуму, тому що базується на емпіричних функціях щільності розподілу;
- швидкість роботи самого класифікатора висока, основний час може займати обчислення вектора ознак.

На основі сполучення баєсового підходу і теорії графів утворюють баєсові мережі. Суть даного підходу в тому, що будують граф, кожна вершина якого відповідає якому-небудь компоненту вектора ознак, дуги позначають причинно-наслідковий зв'язок. Побудова мережі може бути здійснена автоматично шляхом аналізу кореляції компонентів вектора ознак.

Проблемою для баєсової мережі є погана обумовленість, тому що велика розмірність вектора ознак робить граф зв'язків складним для побудови й аналізу. Також сильно зростає обчислювальна складність. Одним з варіантів розв'язання даної проблеми є скорочення розмірності вектора ознак, що приводить до погіршення узагальнюючої здатності.

Класифікація за статистичними признаками можлива після попередньої обробки зображення за допомогою фільтрів, що дозволяють зменшити варіацію параметрів ознак [1].

Наряду з статистичними методами класифікації в обробці зображень використовують евристичні методи. Це ряд підходів, що можна розділити на наступні групи:

- евристичні методи;
- методи порівняння з шаблоном;
- методи з навчанням по прецедентах.

У методах порівняння з шаблоном складається шаблон для зображення всього об'єкта чи його характерних ознак та вводиться функція перевірки відповідності. Ця група є більш універсальним підходом, однак даний метод вимагає наявності дуже точного шаблону об'єкта зображення. Шаблон може бути складною структурою і допускати різні деформації і перетворення, таким чином, сприяючи інваріантності системи до просторових спотворень об'єкта зображення і змін освітленості. Системи, засновані на порівнянні з шаблоном, найчастіше використовуються для розв'язання задач відстеження об'єктів у відео з ініціалізацією на першому кадрі – до початку роботи системи існує загальна модель шаблону, при ініціалізації вона уточнюється і коректується під час роботи системи. При добре заданому шаблоні досягається висока точність і дуже низький рівень збоїв [1]. Інваріантність до просторових спотворень і зміни освітлення залежить від складності шаблону.

У методах з навчанням по прецедентах модель автоматично будується на основі набору зображень об'єкта, складених заздалегідь з можливих вхідних даних системи. Вони є найбільш загальним підходом. Задача розпізнавання об'єктів зображення зводиться до задачі класифікації і для неї застосовується добре розроблений математичний апарат побудови моделі (навчання) по прецедентах. Модель будується автоматично по заздалегідь зібраному наборі прецедентів – зображень, для яких відомо, чи є вони зображеннями об'єкта чи ні. Спостереженням, у даному випадку, є деякий “вектор ознак”, отриманий з вихідного зображення за допомогою перетворення, що відображає зображення в просторі дійсних векторів. Гіпотеза, що підлягає перевірці – приналежність зображення до класу зображень шуканого об'єкта. Таким чином, система розпадається на два модулі – модуль перетворення зображення у вектор ознак і модуль класифікації [4]. Задачею модуля перетворення є найбільш повне й інформативне представлення зображення у виді числового вектора. Задачею модуля класифікації є перевірка гіпотези приналежності зображення класу зображень об'єкта на підставі спостереження, яким є вектор ознак.

Евристичні методи є історично найбільш ранніми. Вони досить прості в реалізації і працюють з високою швидкістю, однак жорстко запрограмовані правила позбавляють

систему гнучкості і стійкості. Як правило, евристичні системи орієнтовані на відносно вузький клас задач. Ця група методів має повну евристичну модель, де експертом складається набір правил, що описують зображення об'єкта (будується модель), згідно з якими здійснюється розпізнавання, а пошук виконується по характерним інваріантним ознакам, де евристично описуються не зображення шуканого об'єкта в цілому, а його інваріанти спотворень (зміна освітленості, поворот, масштабування). Серед евристичних найбільш поширені методи:

- метод опорних векторів (SVM, supporting vectors method);
- sparse network of Winnows (SNoW);
- посилення слабких класифікаторів (classifier boosting).

Метод SVM полягає в побудові оптимального лінійного класифікатора. Класичний алгоритм полягає в побудові лінійної поверхні (гіперплощини), яка рівновіддалена від опуклих оболонок класів, опукла оболонка будується по прецедентах. Стверджується, що така гіперплощина буде оптимальна, з погляду загального ризику, щодо будь-яких інших можливих гіперплощин [3]. Якщо така гіперплощина не існує (класи лінійно не роздільні), то для здійснення нелінійної класифікації застосовується ядрове перетворення, що проектує вихідний простір у простір ще більшої, можливо нескінченної, розмірності.

Метод опорних векторів був успішно застосований для задачі розпізнавання об'єктів зображення. Метод характеризується наступними характеристиками:

- висока стійкість до перенавчання;
- чутливість до шуму може регулюватися за рахунок зменшення точності;
- в системах розпізнавання об'єктів метод дає прискорення в декілька разів у порівнянні з нейронними мережами.

SNoW – особливий вид нейронної мережі. Вектор ознак бінарний. Мережа складається з двох (по числу можливих класів) лінійних нейронів, зв'язаних з компонентами вектора ознак. Класифікація проходить за принципом “переможець забирає все”. Метод SNoW вважається досить ефективним методом для розв'язання задач виявлення об'єктів зображення. Відповідно до деяких досліджень SNoW перевершує по своїм параметрам метод опорних векторів [2].

Метод посилення слабких класифікаторів classifier boosting – це підхід до розв'язання задачі класифікації, шляхом комбінування примітивних класифікаторів в один більш сильний класифікатор. Основна ідея методу полягає в ітеративній мінімізації опуклого функціонала помилки класифікації, шляхом додавання в набір класифікаторів чергового слабого класифікатора. Для систем розпізнавання об'єктів зображення був застосований каскадний підхід, який полягає в побудові каскаду із комплексу слабких

класифікаторів, що працює за принципом послідовних наближень. Характеристики каскадного методу перевершують всі інші системи.

По показникам роботи в реальних системах розпізнавання об'єктів зображення найбільш вдалими виявилися алгоритми boosting (посилення слабких класифікаторів) і SNoW.

1.3 Машинне навчання у методах класифікації

Найчастіше на практиці використовують методи машинного навчання для досягнення результату поставленої задачі. На відміну від методів, заснованих на правилах, вони дозволяють враховувати більшу кількість характеристик. Даний підхід дозволяє абстрагуватись від прив'язаності до конкретної мови та являється автоматизованим.

Машинне навчання (Machine Learning) - великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися. Розрізняють два типи навчання. Машинне навчання відрізняється від класичного алгоритмічного програмування тим, що розроблена система покращує результати своєї роботи із “досвідом”, тобто зі зростанням кількості оброблених даних. Навчання по прецедентах або індуктивне навчання, засноване на виявленні загальних закономірностей по приватним емпіричним даним. Дедуктивне навчання передбачає формалізацію знань експертів і їх перенесення в комп'ютер у вигляді бази знань. Дедуктивне навчання прийнято відносити до області експертних систем, тому терміни машинне навчання і навчання по прецедентах можна вважати синонімами.

Машинне навчання знаходиться на стику математичної статистики, методів оптимізації та класичних математичних дисциплін, але має також і власну специфіку, пов'язану з проблемами обчислювальної ефективності та перенавчання. Багато методів індуктивного навчання розроблялися як альтернатива класичним статистичним підходам.

Машинне навчання - не тільки математична, а й практична, інженерна дисципліна. Чиста теорія, як правило, не призводить відразу до методів і алгоритмів, які можуть застосовуватися на практиці. Щоб змусити їх добре працювати, доводиться винаходити додаткові евристичні, що компенсують невідповідність зроблених в теорії припущень умов реальних завдань. Практично жодне дослідження в машинному навчанні не обходиться без експерименту на модельних або реальних даних, що підтверджує практичну працездатність методу.

Загальна постановка задачі навчання по прецедентах звучить наступним чином. Дана кінцева множина прецедентів (об'єктів, ситуацій), по кожному з яких зібрані

(виміряні) деякі дані. Дані про прецедент називають також його описом. Сукупність усіх наявних описів прецедентів називається навчальною вибіркою. Потрібно за цими даними виявити загальні залежності, закономірності, взаємозв'язки, властиві не тільки цій конкретній вибірці, але взагалі всім прецедентам, в тому числі й тим, які ще не спостерігалися. Кажуть також про відновлення залежностей за емпіричними даними. Основна, але не вичерпна, класифікація задач машинного навчання за прецедентами із деякими класичними методами їхнього вирішення наведена на рисунку 1.1.

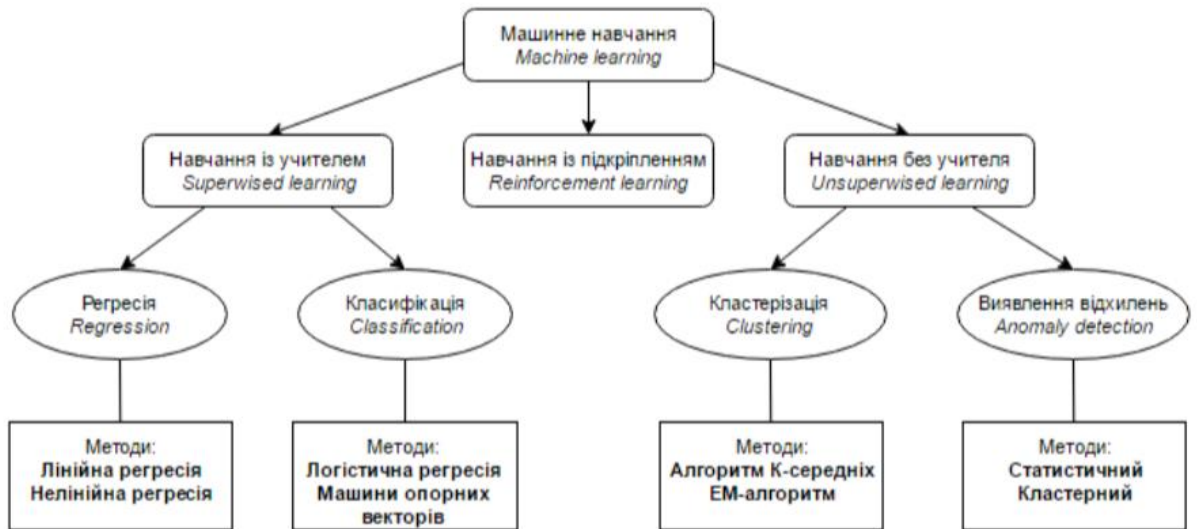


Рисунок 1.1 – Базова класифікація задач машинного навчання за прецедентами

Найбільш поширеним способом опису прецедентів є ознаковий опис. Фіксується сукупність показників n , вимірюваних у всіх прецедентів. Якщо всі показники числові, то признакові описи представляють собою числові вектори розмірності n . Можливі й більш складні випадки, коли прецеденти описуються сигналами, зображеннями, текстами і т. д.

Для розв'язування задачі навчання по прецедентах в першу чергу фіксується модель відновлюваної залежності. Потім вводиться функціонал якості, значення якого показує, наскільки добре модель описує спостережувані дані. Алгоритм навчання шукає такий набір параметрів моделі, при якому функціонал якості на заданій навчальній вибірці приймає оптимальне значення. Процес налаштування моделі за вибіркою даних в більшості випадків зводиться до застосування чисельних методів оптимізації.

У даній роботі проводиться оцінка кількох відомих і загальноприйнятих методів машинного навчання (нейронні мережі, наївний баєсів класифікатор, дерева рішень, метод опорних векторів, прихована марковська модель), а також деякі алгоритми, які досягли

високої точності на різних завданнях машинного навчання з вчителем (випадкові ліси, збірка декількох моделей через голосування та метод стимулювання градієнта).

Первинний аналіз предметної області дає розуміння, що дана задача відноситься до категорії класифікації текстів на основі виділених з них ознак. Одним із способів вирішення таких, коли рішення приймається на основі лінійного оператора над вхідними даними. Він підходить до класу задач, що володіють властивістю лінійної сепарабельності. Кажучи простими словами, операцію лінійної класифікації для двох класів можна представити як відображення об'єктів в багатовимірному просторі на гіперплощину, в якій об'єкти, що попали по одну сторону розділяючої лінії, відносяться до умовно першого класу, а об'єкти по іншу – до другого класу.

Визначення для сказаного формулюється наступним чином. Нехай вектор \bar{x} з дійсних чисел представляє собою вхідні дані, а на виході класифікатора визначається показник y за формулою:

$$y = f(\bar{w} \cdot \bar{x}) = f\left(\sum_j (w_j x_j)\right), \quad (1.2)$$

де \bar{w} – дійсний вектор ваг;

f – функція перетворення скалярного добутку.

Другими словами, вектор ваг \bar{w} – коваріантний вектор або лінійна форма відображення \bar{x} в R . Значення ваг для вектору \bar{w} визначається в ході машинного навчання на прикладах. Функція зазвичай являє собою просту порогову функцію, що відділяє один клас від іншого. В більш складних випадках для неї має місце бути вірогідність для того чи іншого рішення.

На рисунку 1.2 схематично показане відображення об'єктів в просторі, де H_1, H_2, H_3 – гіперплощини, а H_3 – максимальної різниці та гіперплощина, що розділяє точки на два класи найкращим чином.

Існує два основних підходи для визначення параметрів лінійного класифікатора \bar{w} – породжувальні та розрізнявальні моделі. Перша використовує умовний розподіл $P(\bar{x} | class)$.

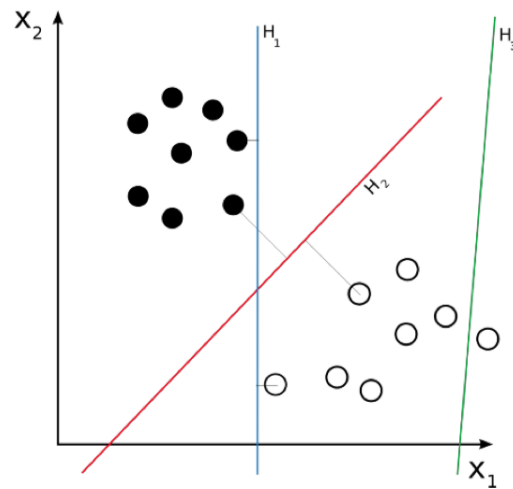


Рисунок 1.2 – Схематичне відображення об’єктів в просторі та їх розділення

Сюди можна віднести:

- лінійний дискримінантний аналіз (лінійний дискримінант Фішера) – передбачає гаусові моделі умовної густини;
- наївний баєсів класифікатор з поліноміальними або багатовимірними моделями подій Бернуллі.

Розрізнявальні моделі (їх ще називають дискримінативні) прагнуть покращити якість вхідних даних на наборі прикладів для навчання. До таких можна віднести:

- логістичну регресію – у цьому алгоритмі не відбувається передбачення значення числової змінної виходячи з вибірки вхідних значень, натомість, значенням функції являється вірогідність того, що дане вихідне значення належить до певного класу;
- простий перцептрон, у якому виконується корекція усіх помилок на вхідному наборі прикладів;
- метод опорних векторів – алгоритм, який максимізує розділення між гіперплощиною рішення та зразками тренувального набору.

В ході досліджень наявних моделей машинного навчання було визначено, що найпопулярнішими являються: наївний баєсів класифікатор, SVM (метод опорних векторів) та логістична регресія для класифікації. Саме тому детальніше розглянемо деякі з них.

1.3.1 Нейронні мережі в задачах обробки зображень

Для розпізнавання складних об'єктів створюють системи на основі нейронних мереж (neural network). Штучна нейронна мережа (ШНМ) є різновидом алгоритмів машинного навчання, який імітує біологічну нейронну мережу. Загалом, для прийняття рішення щодо застосування НМ та машинного навчання потрібно відповісти на наступне запитання: чи можна вирішити задачу без машинного навчання? Якщо з огляду на умови задачі витікає чіткий алгоритм вирішення, слід уникати ускладнення через застосування машинного навчання. В іншому разі наступним кроком є визначення: чи можна вирішити проблему одним з алгоритмів машинного навчання без застосування НМ? Знову ж таки, якщо відповідь – “так”, слід уникати застосування нейронної мережі.

Як правило, НМ має перевагу перед класичними алгоритмами машинного навчання у випадку великої кількості характеристик об'єкта дослідження. Якщо об'єкт можна охарактеризувати кількома змінними (порядку десяти), класичні методи є більш доцільними за нейронну мережу завдяки своїй швидкодії та економії ресурсів. Коли ж кількість характеристик сягає сотень, складність застосування базових алгоритмів (наприклад, нелінійної регресії) робить ШНМ доцільнішою для використання.

Типовим прикладом є розпізнавання зображень. Характеристиками зображення є цифрові значення яскравості та кольору (у випадку кольорових зображень) кожної точки (пікселя). Навіть невелике чорно-біле зображення розміром 50x50 точок у такому разі матиме 2500 характеристик, що їх не взмозі з достатньою швидкістю обробити алгоритм регресії. Якщо помножити це на кількість зображень у навчальній виборці, яке інколи може сягати мільйонів, застосування будь-якого базового алгоритму стає фактично неможливим.

НМ знаходять практичне застосування у багатьох галузях: медичинська діагностика захворювань, біржевий аналіз у економіці, виявлення шахрайства, робототехніка, комп'ютерний зір, розпізнавання мовлення, розпізнавання рукописного тексту тощо. Для різних цілей застосовуються різні типи НМ. Наприклад, рекурентні нейронні мережі вирізняються тим, що отримують на вході свій попередній вихідний стан. Вони можуть застосовуватися для розпізнавання емоцій на відео, коли для точного визначення емоції система повинна аналізувати також попередній кадр (свій попередній стан).

Одним з недоліків застосування НМ є складність підтримки створеної системи. Оскільки розробник не має точних даних щодо процесів, що відбуваються у такій системі, це ускладнює ідентифікацію помилок та їх виправлення.

Вони можуть мати топологію, орієнтовану на розв'язання конкретної задачі із врахуванням властивостей об'єкта – просторово-часову орієнтацію, масштаб, геометричні параметри об'єкта, включаючи координати, кутове положення, лінійний розмір, відстань, тощо. У той же час істотним недоліком типових НМ є відсутність ефективних засобів для розв'язання задач розпізнавання динамічних образів. Основною проблемою інтерпретації динамічних візуальних сцен є висока розмірність простору ознак, наявність геометричних перетворень над об'єктом. Стиск простору ознак виконують методом витягу інтегральних і інваріантних до геометричних перетворень параметрів зображень. Метод геометричних та більш загальних алгебраїчних інваріантів відіграє значну роль у розв'язанні задач розпізнавання зображень. Так, наприклад, інваріанти, у тому числі інваріантні моменти, були успішно використані для розпізнавання профілів літаків і танків, друкованих і рукописних букв, параметрів стикувального вузла космічного апарата, а також багатьох інших об'єктів. Математичне обґрунтування інваріантних особливостей напівтонових зображень базується на теорії алгебраїчних інваріантів [5].

Суть НМ полягає в тому, що мережа складається з елементів, котрі називаються формальними нейронами (formal neuron). Кожен нейрон приймає набір сигналів, що надходять на його входи від одної групи таких же нейронів, обробляє сигнали з врахуванням попередніх сигналів і адаптації до них на основі процедур навчання і передає результати обробки другій групі нейронів. Зв'язки між нейронами кодуються вагами, що відображають важливість їх інформації для визначення загального результату. Основний принцип настроювання нейронної мережі полягає в застосуванні процедур оптимізації та адаптації на основі певних критеріїв, здатності до перенавчання. Однією з переваг НМ є те, що всі елементи можуть функціонувати паралельно, тим самим істотно підвищуючи ефективність розв'язання задач, особливо при обробці зображень в реальному часі. Системи розпізнавання об'єктів зображення, що засновані на нейронних мережах, використовують ієрархічну архітектуру. Спочатку вектор ознак обробляється грубою з високим рівнем похибок, але швидкою мережею, далі, якщо вектор не був класифікований як не об'єкт, алгоритм розв'язання коректується більш точною і більш повільною мережею.

Переважає кількість прикладних нейронних систем передбачає використання багат шарових персептронів (назва “персептрон” походить з англійського perceptron – сприйняття, оскільки перші зразки таких структур призначались для моделювання зору). Популярність персептронів зумовлена широким колом доступних для них задач.

Загалом вони вирішують задачу апроксимації багатовимірних функцій, іншими словами – побудову багатовимірного відображення $F : x \rightarrow y$, яке узагальнює заданий набір прикладів (еталонних пар даних) $\{x^a, y^a\}$ [5].

Залежно від типу вихідних змінних (тип вхідних не має вирішального значення), апроксимація функції може набувати вигляду:

- класифікації (дискретний набір вихідних значень);
- регресії (неперервні вихідні дані).

Множина практичних задач розпізнавання зображень, фільтрації шумів, передбачення часових рядів та інші зводяться до цих базових задач.

Розглянемо алгоритм навчання персептрона на простій модельній задачі. Персептрон навчають, подаючи сукупність (множину) зображень по одному на його вхід і змінюють ваги доти, доки для всіх зображень не буде досягнуто необхідний вихід. Припустимо, що вхідні зображення нанесено на демонстраційні карти. Кожну карту розбито на квадрати і від кожного квадрата на персептрон подається вхідний сигнал. Якщо в квадраті є лінія, то від неї подається одиниця, у протилежному випадку – нуль. Сукупність квадратів на карті задає сукупність нулів і одиниць, котрі подаються на входи персептрона. Мета полягає в тому, щоб навчити персептрон вмикати індикатор за умови подавання на нього сукупності входів, що задають непарне число, і не вмикати у випадку парного. На рисунку 1.3 показана така персептронна конфігурація.

Припустимо, що вектор x є зображенням демонстраційної карти, яка піддається розпізнаванню. Кожну компоненту x_i (квадратик зображення карти) вектора x перемножують на відповідну компоненту w_i вектора ваг w . Ці добутки сумують. Якщо сума перевищує поріг q , то вихід нейрона y дорівнює одиниці (індикатор запалюється), у протилежному випадку – нуль. Цю операцію компактно записують у векторній формі: $y = xw$.

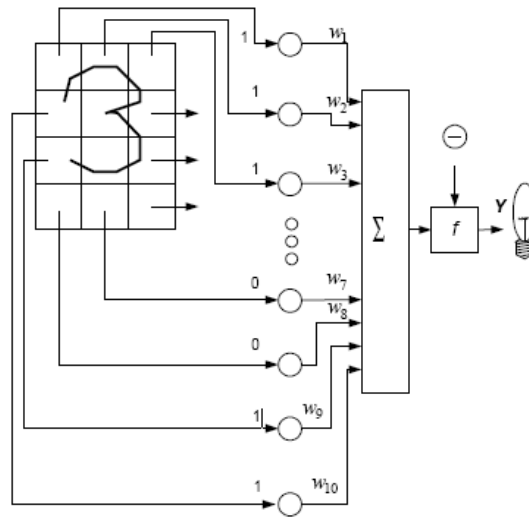


Рисунок 1.3 – Персептронна система розпізнавання зображень

Для навчання мережі образ x подають на вхід і обчислюють вихід y . Якщо вихід правильний, то нічого не змінюється. Однак якщо вихід неправильний, то ваги, приєднані до входів, що підсилюють помилковий результат, модифікуються, щоб зменшити помилку.

Пересвідчимося, як це відбувається. Припустимо, що демонстраційна карта з цифрою 3, подана на вхід і вихід y , дорівнює одиниці (мережа вказує на непарність). Оскільки це правильна відповідь, то ваги не змінюються. Якщо на вхід подають карту з номером 4 і вихід y дорівнює одиниці (непарне число), то ваги, приєднані до одиничних входів, повинні бути зменшені, оскільки вони прагнуть дати невірний результат. Аналогічно, якщо карта з номером 3 дає нульовий вихід, то ваги, приєднані до одиничних входів, необхідно збільшити, щоб скорегувати помилку.

За скінчене число кроків мережа навчиться розділяти карти на парні і непарні за умови, що сукупність цифр лінійно роздільна. Отже для всіх непарних карт вихід буде більшим від порогу, а для всіх парних – нижчим. Зазначимо, що це навчання глобальне, тобто мережа навчається на всій можливій множині вхідних сигналів.

Причина популярності персептронів в тому, що для свого кола задач з погляду обчислювальної складності вони є універсальними та ефективними пристроями.

Недоліком використання нейронних мереж є їх перенавантаження при надмірному збільшенні кількості нейронів у мережі. Іншим недоліком є те, що існує великий клас функцій, які неможливо розділити за допомогою одношарової мережі. Про ці функції говорять, що вони є лінійно нероздільними, і саме вони накладають вагомні обмеження на можливості одношарових мереж.

1.3.2 Лінійні методи класифікації

У галузі машинного навчання метою статистичної класифікації є використання характеристик об'єкту для ідентифікації класу (або групи), до якої він належить.

Лінійний класифікатор досягає цього за допомогою ухвалення рішення про класифікацію на основі значення лінійної комбінації цих характеристик. Характеристики об'єкту відомі також як значення ознак, і зазвичай представляються машині у векторі, що називається вектором ознак. Такі класифікатори добре працюють для таких практичних задач, як класифікація документів, і, загальніше, для задач із багатьма змінними (ознаками), досягаючи рівнів точності, порівнянних з нелінійними класифікаторами, у той же час беручи менше часу на тренування та застосування. Далі будуть розглянуті декілька підходів, що відносяться до даної групи. Зважаючи на те, що основним методом у цій роботі є метод опорних векторів, який відноситься до цієї групи, то розглянемо його детальніше у другому розділі.

1.4 Постановка задачі дослідження

Незважаючи на досягнуті успіхи у питанні класифікації зображень для систем технічного зору ці питання вирішені лише для обмеженого числа задач.

Метою даної роботи є розробка методу класифікації зображень методом машинного навчання SVM. Для даної роботи був обран алгоритм, який базується на інваріантних моментах зображення для того, щоб побудувати вектор ознак, який буде використовувати метод під час класифікації. Для цього необхідно виконати наступні завдання:

- дослідити математичну модель класифікації;
- дослідити математичну модель методу опорних векторів;
- дослідити інваріантні моменти зображень та правила виділення ознак з зображення;
- підготувати тренувальну базу зображень;
- розробити систему класифікації зображень;
- провести порівняльний аналіз результатів.

2 МАТЕМАТИЧНА МОДЕЛЬ КЛАСИФІКАЦІЇ

2.1 Огляд розпізнавання та класифікації

2.1.1 Поняття розпізнавання та її математична модель

Розпізнавання – це здатність живих організмів виявляти в потоці інформації, що надходить від органів чуття, певні об'єкти, закономірності, явища. Воно може здійснюватися на основі зорової, слухової, тактильної інформації. Так, людина без особливих зусиль може дізнатися іншу знайому йому людину, глянувши на нього або почувши його голос. Деякі тварини активно використовують нюх для розпізнавання інших тварин і пошуку їжі.

Можливість розпізнавання спирається на схожість однотипних об'єктів. Незважаючи на те, що всі предмети і ситуації унікальні в строгому сенсі, між деякими з них завжди можна знайти подібності з тою чи іншою ознакою. Звідси виникає поняття класифікації – розбиття всієї множини об'єктів на підмножини – класи, елементи яких мають деякі схожі властивості, що відрізняють їх від елементів інших класів. І, таким чином, завданням розпізнавання є віднесення розглянутих об'єктів або явищ по їх опису до потрібних класів. Тобто поняття розпізнавання можна розширити, якщо говорити про виявлення об'єктів в потоці не тільки чуттєвої, а й будь-якої іншої інформації. Наприклад, можна говорити про розпізнаванні хвороби по її симптомам у хворого або про розпізнаванні соціальних явищ по статистичній інформації.

Розвиток і поширення комп'ютерної обробки інформації привели до виникнення в середині ХХ століття потреб в технологіях, що дозволяють машинам здійснювати розпізнавання в оброблюваній ними інформації. Розробка методів машинного розпізнавання дозволяє розширити коло виконуваних комп'ютерами завдань і зробити машинну переробку інформації більш інтелектуальною. Прикладами сфер застосування розпізнавання можуть служити системи розпізнавання тексту, машинний зір, розпізнавання мови, відбитків пальців та інше. Незважаючи на те, що деякі з цих завдань вирішуються людиною на підсвідомому рівні з великою швидкістю, до теперішнього часу ще не створено комп'ютерних програм, які б вирішували їх в настільки ж загальному вигляді. Існуючі системи призначені для роботи лише в спеціальних випадках зі строго обмеженою сферою застосування.

Кожна система розпізнавання в загальному виді працює за наступною схемою:

– фаза налаштування (навчання);

– фаза функціонування.

Фаза функціонування містить у собі етап попередньої обробки, під час якого відбувається аналіз усіх вхідних даних, та етап прийняття рішень. У цій фазі вирішуються 3 основні задачі функціонування:

- розпізнавання;
- класифікація;
- створення формалізованого опису об'єктів.

Розпізнавання (навчання з учителем) – віднесення пропонованих об'єктів до певних класів за допомогою застосування відомих правил класифікації. Це найбільш типова задача систем розпізнавання. Перед тим, як система зможе виконувати цю функцію, передбачається її навчання на безлічі прикладів – навчальній вибірці об'єктів розпізнавання. Розпізнавання використовується у таких системах:

- OCR;
- розпізнавання відбитків пальців;
- розпізнавання мови;
- біомедичні програми: аналіз і автоматична обробка рентгенограм, електрокардіограм, діагностика захворювань.

Математичну постановку задачі розпізнавання можна представити як множину M об'єктів ω . Об'єкти задаються значеннями деяких ознак x_i , $i = \overline{1, n}$, набори яких однакові для всіх об'єктів. Сукупність ознак об'єкта ω визначає деяким чином його опис $I(\omega) = (x_1(\omega), x_2(\omega), \dots, x_n(\omega))$. Ознаки можуть виражатися в термінах так / ні, так / ні / не відомо, числовими значеннями, значеннями з набору можливих варіантів і т.д.

На всій множині M існує розбиття на підмножини (класи об'єктів):

$$M = \bigcup_{i=0}^m \Omega_i \quad (2.1)$$

Розбиття на класи може бути задано повністю або визначатися деякою апріорною інформацією I_0 про класи Ω_i – наприклад, характеристичним описом об'єктів, що входять в них.

Завдання розпізнавання полягає в тому, щоб для кожного даного об'єкта ω по його опису $I(\omega)$ і апріорній (навчальній) інформації I_0 обчислити значення предикатів:

$$P_i = (\omega \in \Omega_i) \quad \overline{i=1,m}. \quad (2.2)$$

Для опису неможливості розпізнавання об'єктів предикати (2.2) замінюються величинами $\alpha_i \in \{0(\omega \notin \Omega_i), 1(\omega \in \Omega_i), \Delta(\text{unknown})\}$. Таким чином, для даного об'єкту ω необхідно обчислити його інформаційний вектор:

$$\alpha(\omega) = (\alpha_1(\omega), \dots, \alpha_m(\omega)). \quad (2.3)$$

Процедура, яка будує інформаційний вектор (2.3) в даному випадку висловлює алгоритм прийняття рішення про віднесення об'єкта ω до того чи іншого класу і називається “вирішальною функцією”.

Розбиття даної множини об'єктів на класи Ω_i може бути задано переліченням, де кожен клас задається шляхом прямої вказівки його членів. Такий підхід використовується в тому випадку, якщо доступна повна апіорна інформація про всі можливі об'єкти розпізнавання. Образи, які пред'являються системі, порівнюються з заданими описами представників класів і відносяться до того класу, якому належать найбільш схожі з ними зразки. Такий підхід називають методом порівняння з еталоном. Його, наприклад, можна застосувати при розпізнаванні машинопечатних символів певного шрифту. Його недоліком є слабка стійкість до шумів і спотворень в образах.

Можна розбивати множину за рахунок задавання загальних властивостей. Клас задається вказівкою деяких ознак, притаманних усім його членам. Об'єкт, що розпізнається, в такому разі не порівнюється безпосередньо з групою етальонних об'єктів. В його первинному описі виділяються значення певного набору ознак, які потім порівнюються з заданими ознаками класів. Такий підхід називається порівняння за ознаками. Він економічніше методу порівняння з еталоном в питанні кількості пам'яті, необхідної для зберігання описів класів. Крім того, він допускає деяку варіативність розпізнаються образів. Однак, головною складністю є визначення повного набору ознак, які точно відрізняють членів одного класу від членів всіх інших.

Останнім методом розбиття множини є кластеризація. У разі, коли об'єкти описуються векторами ознак або вимірювань, клас можна розглядати як кластер. Розпізнавання здійснюється на основі розрахунку відстані опису об'єкта до кожного з наявних кластерів. Якщо кластери досить рознесені в просторі, при розпізнаванні добре працює метод оцінки відстаней від розглянутого об'єкта до кожного з кластерів. Складність розпізнавання зростає, якщо кластери перекриваються. Зазвичай це є

наслідком недостатності вихідної інформації і може бути вирішено збільшенням кількості вимірювань об'єктів. Для завдання початкових кластерів доцільно використовувати процедуру навчання.

Альтернативою ручному введенню в систему описів класів об'єктів і параметрів вирішальної функції є навчання системи. Воно являє собою процедуру самоналаштування системи розпізнавання на основі сприйнятої інформації і може відбуватися як при підготовці системи, так і в процесі її роботи в міру “накопичення досвіду”. Вид навчання, в якому системі представляється набір зразків розпізнаваних об'єктів із зазначенням їх приналежності класам, називається навчанням з учителем. Набір зразкових об'єктів називається навчальною вибіркою. Ненавчена система виконує розпізнавання об'єктів і порівнює свої результати з правильними відповідями, які надходять від вчителя. За результатами порівняння система виправляє параметри вирішальної функції. У разі використання методу перелічення для завдання класів система просто зберігає в пам'яті всю доступну про об'єкти інформацію. Якщо класи задаються описом загальних властивостей, то система визначає різницю в значеннях ознак об'єктів різних класів. При використанні кластерного підходу з аналізу векторів ознак об'єктів навчальної вибірки формуються вихідні кластери в просторі ознак.

Отже, навчальна вибірка в задачі розпізнавання є апріорною інформацією про безліч розпізнаваних об'єктів і представляє опис всіх класів об'єктів:

$$T = I_0(\Omega_1, \dots, \Omega_m). \quad (2.4)$$

Її складають описи об'єктів із зазначенням їх приналежності класам, які запропоновані вчителем, тобто можна визначити її як сукупність описів об'єктів:

$$(I(\omega_1), \dots, I(\omega_{r_1}), I(\omega_{r_1+1}), \dots, I(\omega_{r_2}), \dots, I(\omega_{r_m})), \quad (2.5)$$

де об'єкти $\omega_1, \dots, \omega_{r_1}$ належать класу Ω_1 , об'єкти $\omega_{r_1+1}, \dots, \omega_{r_2}$ – класу Ω_2 і т.д.

Таким чином, навчальна вибірка представляє собою таблицю 1.1, рядки якої позначені назвами об'єктів $\omega_i, i = \overline{1, r_m}$, а стовпці – назвами ознак $x_j, j = \overline{1, n}$. Елементами таблиці є значення ознак об'єктів $x_j(\omega_i)$. Рядки таблиці згруповані за класами $\Omega_j, j = \overline{1, m}$.

Таблиця 1.1 – Навчальна вибірка

	x_1	x_2	...	x_n	Класи
ω_1	$x_1(\omega_1)$	$x_2(\omega_1)$...	$x_n(\omega_1)$	Ω_1
ω_2	$x_1(\omega_2)$	$x_2(\omega_2)$...	$x_n(\omega_2)$	
...	
ω_{r_1}	$x_1(\omega_{r_1})$	$x_2(\omega_{r_1})$...	$x_n(\omega_{r_1})$	
...
$\omega_{r_{m-1}+1}$	$x_1(\omega_{r_{m-1}+1})$	$x_2(\omega_{r_{m-1}+1})$...	$x_n(\omega_{r_{m-1}+1})$	Ω_m
...	
ω_{r_m}	$x_1(\omega_{r_m})$	$x_2(\omega_{r_m})$...	$x_n(\omega_{r_m})$	

2.1.2 Поняття класифікації та її математична модель

Класифікація (навчання без вчителя) – розбиття множини об'єктів на непересічні класи по їх формалізованим описам. Дане завдання вирішується в тих випадках, коли від системи не потрібно віднесення вхідних образів до будь-яких певних класів, а потрібно лише здатність розрізнити їх будь-яким способом за певними ознаками. Можна говорити про оперування безіменними класами об'єктів. Вони призначені, щоб кластеризувати безліч об'єктів або для зниження розмірності описів для спрощення подальшого аналізу.

Математичну постановку задачі класифікації можна представити як множину M об'єктів ω , які мають свої описи $I(\omega) = (x_1(\omega), x_2(\omega), \dots, x_n(\omega))$. Заданий критерій $K(I(\omega))$, що дозволяє відрізнити об'єкти один від одного відповідно до визначеної умови. Наприклад, такий критерій може видавати ціле число, що визначає положення об'єкта в деякій ранговій системі. Потрібно відповідно до заданого критерію K побудувати розбиття множини M на класи (2.1). Дія критерію в цьому випадку визначається наступним виразом:

$$\forall i, j (i, j = \overline{1, m}) \quad i \neq j \Leftrightarrow K(I(\omega \in \Omega_i)) \neq K(I(\omega \in \Omega_j)). \quad (2.6)$$

Слід зазначити, що число класів m спочатку невідомо і визначається за результатами виконання класифікації.

Критерій класифікації можна вибрати за допомогою тих же методів, що і при розпізнаванні:

- пряме перелічення, де можна задати приклади свідомо різних об'єктів та приймати рішення про віднесення об'єктів до того чи іншого класу на основі міри схожості з еталонами. Вирішальна функція будується на основі аналізу еталонних об'єктів і апріорної інформації про їх належність до різних класів;

- опис загальних властивостей, що дозволяє сформулювати правило класифікації на основі аналізу характеристик вхідного об'єкта;

- кластеризація, що дозволяє розрізняти вхідні об'єкти на основі аналізу характерних властивостей рознесених в просторі кластерів.

Для навчання системи класифікації, як і системи розпізнавання, можна скористатися навчальною вибіркою. На цей раз від системи не потрібно запам'ятовування конкретних класів об'єктів і подальшого віднесення до них образів. Достатньо лише визначення, чи відносяться представлені об'єкти до одного або до різних класів. Тому при навчанні системі повинна бути представлена вибірка об'єктів із зазначенням, які з них різні, а які – подібні. Іншими словами, системі потрібно показати приклади відрізняння образів. Не важливо, які значення буде виробляти вирішальна функція, важливо, щоб ці значення були різними для різнотипних об'єктів і однаковими для однотипних. Таке навчання, в якому відсутні підказки правильних відповідей вчителем, називається навчанням без вчителя. Під час навчання без вчителя система будує кластери об'єктів навчальної вибірки, а на етапі функціонування виробляє визначення близькості вхідних образів до побудованих кластерів.

Навчальні вибірки можуть бути двох типів:

- фіксована вибірка;
- послідовна вибірка.

Фіксована вибірка будується на основі навчальної вибірки, яка відома системі до початку виконання її функції розпізнавання. Надалі вона не змінюється; всі об'єкти, що підлягають розпізнаванню, оцінюються на основі отриманих при навчанні даних. Помилки, що здійснюються системою при розпізнаванні, не враховуються і носять постійний характер. Системи, побудовані на цьому принципі, вимагають ретельного навчання і використовуються в основному там, де об'єкти, які потрібно розпізнати, добре відомі, варіативність об'єктів передбачувана або добре відпрацьовується правилами класифікації. В цілому, такі системи більш прості і надійні.

Послідовна вибірка виконується у декілька етапів. Первісна вибірка формує лише попередню версію правил класифікації. При подальшій роботі система оцінює якість

розпізнавання і підлаштовує правила класифікації або виробляє нові з метою підвищення якості розпізнавання. Процес припиняється тоді, коли правила класифікації починають задовольняти деяким критеріям оптимальності. Цей підхід називається адаптивним розпізнаванням або машинним навчанням. Властивості процедури послідовного навчання:

- збіжність. Процес зміни правил класифікації повинен призводити через кінцеве число коригувань до остаточного варіанту;
- оптимальність. Вирішальне правило, до якого сходиться процес навчання, має бути оптимальним в сенсі мінімізації функції вартості можливих помилок;
- обчислювальна складність. Фінальне правило класифікації має мінімізувати число кроків, необхідних для розпізнавання образу та час їх виконання.

Системи з послідовною процедурою навчання проявляють більшу гнучкість при розпізнаванні об'єктів з високою варіативною складовою. Крім того, “накопичення досвіду” дозволяє їм досягати високої якості і швидкості розпізнавання. Однак, алгоритм роботи таких систем більш складний і вимагає обґрунтування з точки зору виконання зазначених вище властивостей.

2.1.3 Вид правил класифікації

Правила класифікації можуть бути двох типів: паралельні або послідовні.

У випадку паралельної класифікації для розпізнавання система виробляє ряд тестів над усіма компонентами x_i , $i = \overline{1, n}$ опису вхідного об'єкта одночасно. Вирішальна функція в цьому випадку є функцією не більше, ніж n змінних $g(x_1, \dots, x_n)$. Система може бути організована у вигляді множини паралельних функцій, кожна з яких виробляє оцінку належності об'єкта до відповідного їй класу, і має вигляд:

$$F = \{f_i(x_1, \dots, x_n)\}, j = \overline{1, m}. \quad (2.7)$$

В такому випадку вирішальна функція приймає рішення на основі максимального отриманого значення – $g = \max_i(f_j)$.

Паралельна процедура є достатньо надійною і займає для розпізнавання сталий час, який дорівнює часу виконання найтривалішої з процедур розпізнавання. Однак, вона не є гнучкою, що властиво послідовним процедурам [14].

Послідовний спосіб класифікації, як і паралельний, передбачає ряд тестів над ознаками об'єкта. Однак, виконуються вони не одночасно, а послідовно, причому порядок їх виконання може залежати (і найчастіше залежить) від результатів попередніх тестів.

Послідовна класифікація дозволяє скоротити число необхідних для розпізнавання тестів. Час виконання в порівнянні з паралельною процедурою залежить від обчислювального пристрою, що виконує її. Якщо пристрій дозволяє виробляти паралельні обчислення, то послідовна процедура потребує значно більшого часу. Якщо ж пристрій послідовний, то в гіршому випадку на виконання послідовної процедури потрібно рівно стільки ж часу, скільки і на виконання паралельної. В цілому послідовна процедура дозволяє реалізувати більш складні правила класифікації.

Можливе комбінування цих двох принципів з метою досягнення компромісу між простотою паралельних процедур і можливостями послідовних.

2.1.4 Способи опису об'єктів

Відображення об'єктів у вигляді точок Евклідового простору будується наступним чином. Над первинним зображенням образу, який треба розпізнати, проводиться серія обчислень, що визначають необхідні для класифікації характеристики. Далі в багатовимірному Евклідовому просторі (параметричному просторі, або просторі характеристик), кожний вимір якого відповідає одній з обчислюваних характеристик, будується точка, відповідна сукупності отриманих вимірювань. За сукупностями точок, Евклідова відстань між якими мала, виділяють усі простори в область, яка відповідає певному класу зображень.

Як правило, методи, які працюють з Евклідовим простором, використовують паралельну процедуру подачі вхідного зображення. Розпізнавання ґрунтується на проведенні ряду математичних обчислень над повною множиною точок зображення. Можна виділити наступні основні види правил класифікації, що використовуються найчастіше: вирішальні (дискримінантні) функції, функції відстані та функції правдоподібності. Також до даного класу методів можна також віднести методи, що використовують в своїй основі нейронні мережі, їх детального опису не буде у даній роботі, але знати про їх наявність потрібно. Вони активно використовують процедуру самонавчання (послідовна навчальна вибірка) і відносяться до паралельних методів. Нейронні мережі здатні вирішувати як завдання розпізнавання, так і автоматичної класифікації, допускаючи навчання як з учителем, так і без нього.

У випадку дискримінантних функцій, які зображені на рисунку 2.1, побудова правил класифікації полягає в пошуку функцій, що описують розподіл параметричного простору на відповідні класам об'єктів області. При цьому реалізується спосіб завдання класів у вигляді вказівки загальних властивостей належних їм об'єктів.

При використанні функцій відстані, схематичне зображення яких можна побачити на рисунку 2.2, визначення приналежності об'єктів різних класів проводиться на основі аналізу відстаней між точками об'єктів одного класу та між точками об'єктів різних класів.

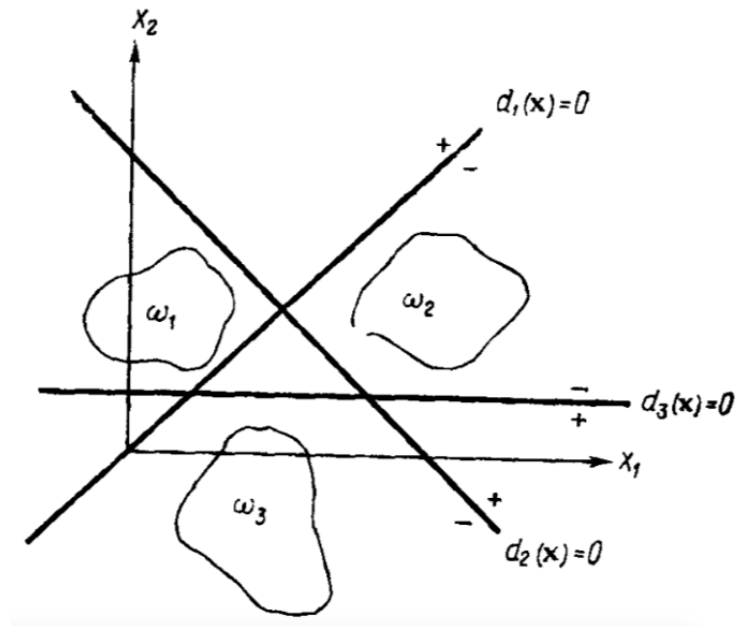


Рисунок 2.1 – Схематичне зображення вирішальних функцій

Класи представляються у вигляді кластерів в параметричному просторі. Побудова правил класифікації полягає в пошуку функцій, що забезпечують оптимальну побудову кластерів об'єктів. Даний підхід активно використовує методи кластерного аналізу [17].

У функціях правдоподібності використовується математична статистика, Байєсовські оцінки, теорія ігор. Процес розпізнавання представляється як гра пристрою, що розпізнає об'єкти, з реальним світом, в якій при розпізнаванні кожного символу машина намагається вгадати символ, задуманий природою. Побудова правила класифікації зводиться до пошуку функцій вигравів і втрат гравців. Ухвалення рішення про клас об'єкта будується на значенні функції правдоподібності передбачуваного рішення при заданих значеннях параметрів об'єкта. Даний метод можна застосовувати при наявності достатньої кількості апріорної інформації про множину, що розпізнається.

Наступною категорією опису об'єкта є списки ознак. Методи даної категорії спираються на можливість класифікації розпізнаваних об'єктів на основі наявності в них деяких характерних ознак. Існують два основні підходи.

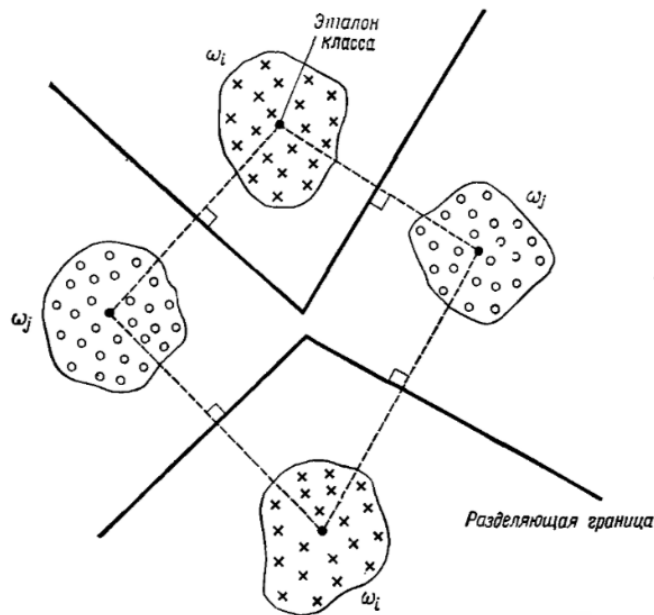


Рисунок 2.2 – Схематичне зображення функцій відстані

Перший ґрунтується на припущенні, що прості вимірювання, проведені над зображенням, є результат дії сукупності невеликого числа ознак. Завданням розробки методу розпізнавання є визначення простору ознак, до якого потрібно звести простір прямих вимірювань. При цьому простір ознак має меншу розмірність. При такому підході використовується апарат факторного аналізу. Другий підхід визначає ознаки як підмножини множин простих вимірювань. При розпізнаванні бінарних зображень такими ознаками можуть служити наявність чорних крапок в певних областях зображення (наприклад, діагональ або горизонтальна риса в середині зображення), число чорних крапок на характеристичній лінії, проведеної через зображення і т.д. Об'єкти, що розпізнаються, представляються як різні сукупності спостережуваних ознак. Методи даного класу є в великій мірі евристичними. Їх ефективність визначається правильністю вибору набору розглянутих ознак. Можна розглянути приклад з робіт Кадакін. При розпізнаванні рукописних шифрів джерел картотеки для кожного рядка шифру здійснюється визначення значень декількох ознак. В ролі цих ознак виступають кількість чорних крапок на середній і нижній лінії рядка з урахуванням її можливого нахилу, а також число верхніх і нижніх виступаючих елементів букв.

Далі за отриманими даними проводиться оцінка числа букв в словах. Число букв є складною ознакою більш високого рівня. Вона використовується далі для зіставлення

отриманих даних з відомостями про наявні шифри і ідентифікації спостережуваного шифру за результатами зіставлення [6].

Також варто звернути увагу на структурний опис ознак, який ґрунтується на поданні об'єктів у вигляді сукупності “непохідних елементів” і відносин між ними. Під непохідними елементами розуміються фрагменти образів, які, з одного боку, формують ці образи, з іншого - прості в сенсі власної структури, тобто не містять інших непохідних елементів, хоч трохи значущих для опису образу. Як правило системи, що використовують структурний опис об'єктів, реалізують послідовну процедуру розпізнавання. У процесі рішення інформаційного завдання вони обробляють вхідний образ, обходячи його структуру елемент за елементом. Процес розпізнавання ділиться на два потоки: виділення в образі структурних елементів певного виду і узгодження одержуваної структурної інформації з наявними в системі моделями для класів зображень. Одним з видів структурних методів є методи граматичної класифікації образів. Вони використовують уявлення образів у вигляді пропозицій спеціальної мови. На етапі ініціалізації алгоритму потрібно визначити види можливих структурних елементів зображень. Побудова правила класифікації зводиться до висновку граматики, яка описує мову образів, що класифікуються. Розпізнавання полягає у визначенні виводимості розглянутої пропозиції за допомогою знайденої граматики. Прикладами граматичного опису зображень служать: мова опису зображень PDL (Picture Definition Language), плекс-граматики, веб-граматики, синтаксичний опис рукописних символів. Наприклад, граматики Еванса використовують в якості термінальних символів прості елементи зображень, такі як коло, прямокутник, лінія. Як термінали, що описують взаємини елементів, використовуються спеціальні оператори, такі як “над”, “всередині” і т.д.

В якості структурних елементів зображень можуть використовуватися траєкторії руху інструменту, яким воно було написано або намальовано. У такому випадку ці елементи можуть характеризуватися послідовністю напрямків кроків переміщення пера. Напрями можуть кодуватися кодами з кінцевого набору наперед заданих напрямів. Іншим способом є опис елементів за допомогою кривих Безьє.

2.2 Принцип побудови опорних векторів

Метод опорних векторів, відомий як support vector machine (SVM), є машинним алгоритмом, котрий навчається на прикладах та використовується для класифікації об'єктів. Цей метод класифікації належить до групи граничних методів; визначає класи за

допомогою меж просторів. Опорними векторами вважаються об'єкти множини, що лежать на цих межах. Класифікація вважається вдалою, якщо простір між межами — порожній. Наприклад, SVM може розрізнити аварійний режим роботи електромеханічної системи та класифікувати його за наявності попередніх досліджень, можливих за технологічними вимогами режимів роботи. Такий підхід розкриває значні можливості для будування адаптивних систем автоматичного керування.

В основі SVM лежить деяка математична сутність – алгоритм максимізації деякої математичної функції відносно наявного набору даних. Для розуміння того, як працює SVM, потрібно мати уявлення про чотири ключові поняття:

- відділяюча гіперплощина (the separating hyperlane);
- гіперплощина максимальної межі (the maximum-margin hyperlane);
- м'яка межа (the soft margin);
- функція ядра (the kernel function).

Відділяюча гіперплощина є математичною сутністю, що відділяє між собою класи об'єктів з однаковими ознаками. Наприклад, так як це показано на рисунку 2.3, де у тривимірному просторі площина відділяє кульки світлого кольору від темних кульок.

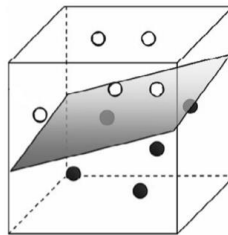


Рисунок 2.3 – Приклад відділяючої площини

Можна екстраполювати цю процедуру математично до вимірів, значно вищих за третій. Загальний термін для лінії, котра відділяє елементи різних класів, – багатовимірна гіперплощина. Спосіб, яким можна провести відділяючу гіперплощину за методом SVM, не є унікальним. Завжди існує багато різних можливостей розташування гіперплощини, як це показано на рисунку 2.4.

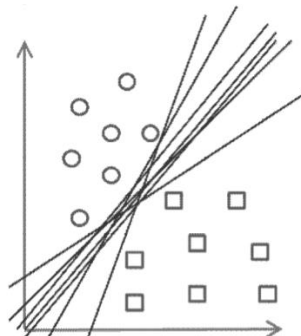


Рисунок 2.4 – Можливі варіанти розміщення гіперплощини у двовимірному просторі

SVM відрізняється від інших гіперплощинних методів класифікації тим, що він дозволяє обирати оптимальне розташування гіперплощини. Гіперплощина обирається таким чином, щоб бути розташованою на максимальній відстані від елементів кожного з класів, тобто посередині деякої зони, що відділяє між собою ці елементи [11]. Цей процес зображен на рисунку 2.5, де граничні елементи мають сірий колір. В цьому полягає сутність другого ключового поняття – гіперплощина максимальної межі.

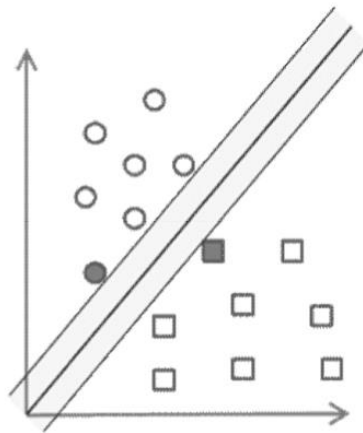


Рисунок 2.5 – Розташування гіперплощини максимальної межі

Об'єкти, що класифікуються, не завжди можуть бути розділені гіперплощиною. У реальних системах будуть наявними похибки в даних, внаслідок яких гіперплощина не виконає розподіл абсолютно точно. Наявність похибок під час розподілу даних можна побачити на рисунку 2.6, де декілька елементів з кожної категорії можна побачити серед елементів іншого класу.

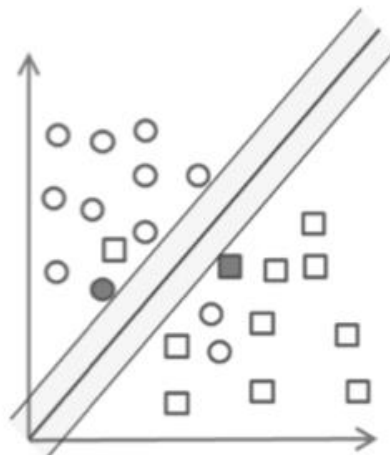


Рисунок 2.6 – Похибки під час класифікації об'єктів

Тому для роботи методу SVM вводять допустиму похибку класифікації, що називається м'якою межею.

Звісно, що метод SVM не повинен враховувати забагато похибок класифікації об'єктів, тому потрібно вводити додатковий параметр, котрий встановлює скільки невірно класифікованих об'єктів можуть перетинати гіперплощину максимальної межі і як далеко вони можуть розташовуватись відносно неї. Таким чином, вводиться так звана м'яка межа похибки навколо гіперплощини.

Об'єкти, що класифікуються, можуть бути поділені лінійно лише в окремих випадках. Здебільшого вони не є такими, що допускають лінійне розподілення. Для вирішення проблеми лінійного розподілення використовують функції ядра, що проєктують дані з низьковимірному простору у багатовимірний. При вірному виборі функції ядра об'єкти можуть бути розділені лінійно гіперплощиною у багатовимірному просторі. Таким чином, функції ядра виконують роль спрямляючого простору.

Графічний приклад переходу від задачі, що не має лінійного розподілення об'єктів, до такої, яка дозволяє будівництво гіперплощини максимальної межі, наведено на рисунку 2.7.

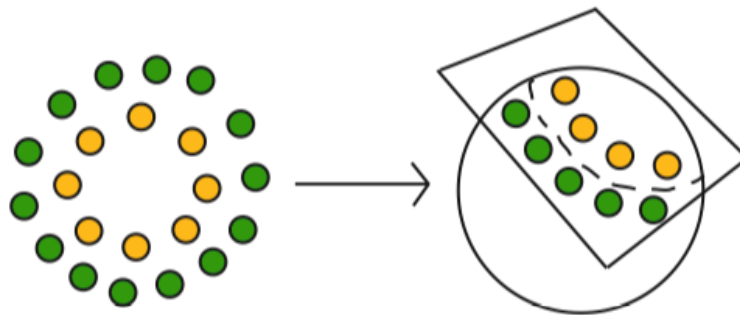


Рисунок 2.7 – Приклад спрямлення простору

Розглянемо головні математичні залежності, на базі яких працює SVM, та поставимо типову задачу класифікації. Кожен з об'єктів класифікації розглядається як вектор у n -вимірному просторі.

Кожна координата вектора – це деяка ознака, і вона тим більша, чим більше ця ознака відображена у даного об'єкта. Чим менше ця координата, тим менше ознака відповідає об'єкту [15].

Під час навчання SVM користуються учбовими колекціями, тобто множиною векторів (x_1, x_2, \dots, x_n) , які належать до гіперповерхні R^d та чисел (y_1, y_2, \dots, y_n) , значення яких належать до множини $\{-1, 1\}$. Причому число y_i дорівнює 1, якщо відповідний йому вектор x_i належить до категорії, що розглядається, та -1 – у

протилежному випадку. Нехай є деяка гіперплощина, що розділяє позитивні та негативні приклади. Тоді точки x , що лежатимуть на цій площині, будуть задовольняти умові:

$$w^* x + b = 0, \quad (2.8)$$

де w – нормальний до гіперплощини вектор.

Перпендикуляр, що визначає відстань від гіперплощини до початку координат, визначиться як $|b|/\|w\|$, де вираз w називають Евклідовою нормою або довжиною вектора w .

Позначимо найкоротшу відстань від відділяючої гіперплощини до найближчого “позитивного” прикладу d_+ , до найближчого “негативного” – d_- . Тоді межа навколо гіперплощини матиме ширину $(d_+ + d_-)$. У тому випадку, коли задача є лінійно розділюваною, SVM шукає відділяючу гіперплощину з максимальною межею (таким чином, щоб відстань між елементами, які належать до класу та тими елементами, що не належать до нього, була найбільшою).

Тоді розташування векторів у лінійно розділюваній задачі можна описати наступною системою нерівностей:

$$\begin{cases} x_i w + b \geq +1, y_i = +1, \\ x_i w + b \leq -1, y_i = -1. \end{cases} \quad (2.9)$$

Систему нерівностей (2.9) можна спростити до вигляду:

$$y_i(x_i w + b) - 1 \geq 0, \forall i. \quad (2.10)$$

Межі навколо гіперплощини максимальної межі також являють собою гіперплощини, а саме – гіперплощину H_1 , що описується рівнянням $x_i w + b = 1$, та гіперплощину H_2 з рівнянням $x_i w + b = -1$. Ці гіперплощини будуть паралельними одна до одної та матимуть один нормальний вектор w . Відстань від площини H_1 до початку координат буде $|1 - b|/\|w\|$, а від гіперплощини H_2 до початку координат – $|-1 - b|/\|w\|$. Відповідно, значення $d_+ = d_- = 1/\|w\|$, а ширина максимальної межі визначиться $d_+ + d_- = 2/\|w\|$. Таким чином, при умові того, що задача є лінійно

розділюю, виконується пошук пари гіперплощин, котрі розташовані одна від одної на максимальній відстані, для чого мінімізують $\|w\|$. Гіперплощина максимальної межі буде розташовуватись посередині на рівній відстані від H_1 і H_2 паралельно до них.

Лінійне розділення точок за зазначеним вище принципом для двовимірної задачі наведено на рисунку 2.8. Точки, що лежать у гіперплощинах H_1 та H_2 , та виключення яких з прикладів призвело б до зміни положення гіперплощини максимальної межі, називають опорними векторами, які на рисунку 2.8 обведені колами.

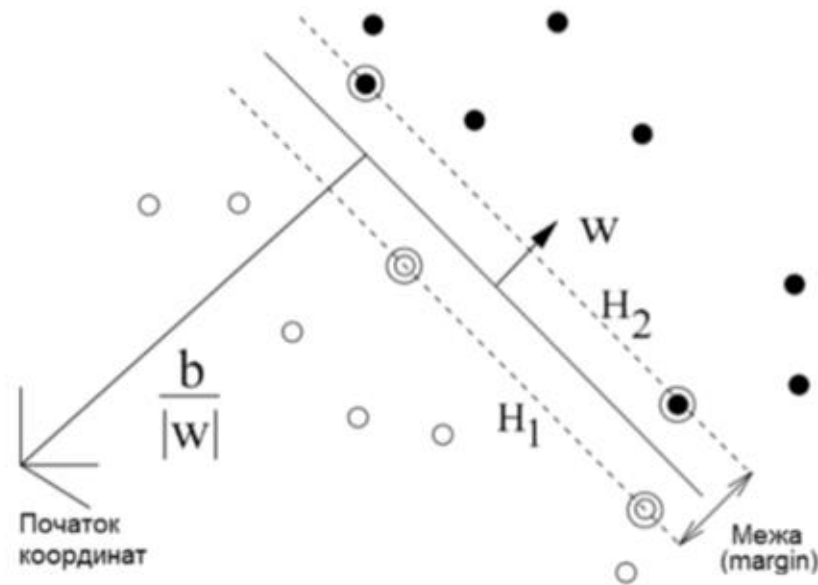


Рисунок 2.8 – Лінійне розділення точок

Відомо, що для знаходження мінімуму функції потрібно дослідити її похідну. У випадку мінімізації $\|w\|$ задача ускладнюється тим, що існують лінійні обмеження, які слід враховувати при мінімізації $\|w\|$. Множина точок, які задовольняють обмеженням, в n -вимірному просторі являє собою багатогранник: простір ділиться гіперплощинами або напівгіперплощинами у залежності від того стоїть в обмеженнях знак рівності чи нерівності. Пошук мінімуму відбувається в цьому обмеженому просторі.

Розв'язати задачу мінімізації в обмеженому просторі можна за допомогою метода Лагранжа.

Лагранж звів задачу пошуку умовного мінімуму до задачі мінімізації без обмежень, щоб потім скористатись стандартним методом пошуку мінімуму функції [21]. Для використання метода Лагранжа потрібно змінити функцію, що підлягає мінімізації. Для

формування нової функції потрібно ввести множники Лагранжа $\alpha_i, i = \overline{1, n}$ – один для кожного елементу нерівності (2.9). Лагранжیان має наступний вигляд:

$$L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^n \alpha_i. \quad (2.11)$$

Тепер потрібно мінімізувати лагранжیان (2.11) відносно w , b , одночасно вимагаючи, щоб похідні відносно всіх α_i дорівнювали нулю для $\alpha_i \geq 0$. Ці вимоги призводять до необхідності виконання наступних умов (lagrangian trick):

$$w = \sum_i \alpha_i y_i x_i, \quad (2.12)$$

$$\sum_i \alpha_i y_i = 0 x_i. \quad (2.13)$$

Тоді з урахуванням формул (2.12, 2.13) вираз (2.11) можна представити у вигляді:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j. \quad (2.14)$$

У формулах (2.11) і (2.14) для позначення лагранжіана використовуються різні індекси: у формулі (2.11) індекс “P” (основний), а у формулі (2.14) індекс “D” (подвійний). Ці два вирази в деякій мірі не збігаються, бо вони формуються на основі однієї і тієї ж функції оптимізації, але з різними обмеженнями, саме тому рішення полягає в мінімізації L_p або максимізації L_D .

Якщо завдання оптимізації сформульовано так, що $b = 0$, тобто гіперплощини проходять через початок координат, то обмеження (2.13) не виникає. У наведених рівняннях точки, для яких $\alpha_i > 0$, називаються опорними векторами, вони лежать на одній з гіперплощин H_1 або H_2 . Для всіх інших точок $\alpha_i = 0$.

Для методів класу віртуальної машини, які навчаються за таким принципом, опорні вектори є критичними точками навчальної множини. Якщо інші точки змінюються або переміщуються в просторі, не зачіпаючи опорні вектори, то у цьому випадку результат навчання (відділяюча гіперплощина) не зміниться.

При оптимізації в просторі з обмеженнями застосовують умови Каруша-Куна-Таккера (Karush-Kuhn-Tucker) або ККТ, які є узагальненням методу множників Лагранжа. У теорії оптимізації умови ККТ є необхідними умовами для вирішення завдань нелінійного програмування. Для того, щоб рішення було оптимальним, повинні бути виконані наступні умови для лагранжіана L_p :

$$\frac{\partial}{\partial w_v} L_p = w_v - \sum_i \alpha_i y_i x_{iv} = 0 \quad v = \overline{1, d}, \quad (2.15)$$

$$\frac{\partial}{\partial b} L_p = -\sum_i \alpha_i y_i = 0, \quad (2.16)$$

$$y_i(x_i w + b) - 1 \geq 0 \quad i = \overline{1, n}, \quad (2.17)$$

$$\alpha_i \geq 0 \quad \forall i, \quad (2.18)$$

$$\alpha_i(y_i(w x_i + b) - 1) = 0 \quad \forall i. \quad (2.19)$$

Таким чином, оптимізація за ККТ у випадку лінійно розділюваної задачі буде являти собою один з варіантів категоризації за методом SVM.

Під час виконання категоризації реальних даних можуть виникати похибки, котрі призводять до неможливості лінійного розподілення. Для усунення впливу похибок вводяться так звані коефіцієнти вартості ξ_i , $i = \overline{1, n}$ у нерівності обмежень, які після цього приймають наступний вигляд (при цьому $\xi_i \geq 0 \quad \forall i$):

$$\begin{cases} x_i w + b \geq +1 - \xi_i, y_i = +1, \\ x_i w + b \leq -1 + \xi_i, y_i = -1. \end{cases} \quad (2.20)$$

Коли виникає похибка, тоді відповідний коефіцієнт вартості ξ_i змінює праву частину нерівності i вона починає відрізнятися від $+1$ або -1 . Тоді $\sum_i \xi_i$ буде верхньою

межею значення похибки навчання методу. Логічно змінити функцію, що підлягає мінімізації з $\frac{1}{2}\|w\|^2$, до вигляду:

$$\frac{1}{2}\|w\|^2 + C\left(\sum_i \xi_i\right)^k, \quad (2.21)$$

де C – параметр, який обирається користувачем, збільшення C робить вимоги до точності більш жорсткими.

Головний лагранжян L_p при наявності коефіцієнтів вартості ξ_i представляється наступним чином:

$$L_p = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i \{y_i(x_i w + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i, \quad (2.22)$$

де μ_i – множники Лагранжа, що вводяться для підсилення позитивних значень коефіцієнтів вартості ξ_i .

Для виконання вимог ККТ та лагранжіана L_p при умові зміни значення i від 1 до кількості точок n , що приймають участь у навчанні методу, а значення v від 1 до розмірності простору даних d . Тоді одержимо такі умови ККТ при категоризації з похибкою:

$$\frac{\partial L_p}{\partial w_v} = w_v - \sum_i \alpha_i y_i x_{iv} = 0, \quad (2.23)$$

$$\frac{\partial L_p}{\partial b} = -\sum_i \alpha_i y_i = 0, \quad (2.24)$$

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - \mu_i = 0, \quad (2.25)$$

$$y_i(x_i w + b) - 1 + \xi_i \geq 0, \quad (2.26)$$

$$\xi_i \geq 0, \alpha_i \geq 0, \mu_i \geq 0, \quad (2.27)$$

$$\alpha_i \{y_i(x_i w + b) - 1 + \xi_i\} = 0 \quad (2.28)$$

$$\mu_i \xi_i = 0. \quad (2.29)$$

Як і в попередніх визначеннях, значення b можна розрахувати з рівнянь (2.28) та (2.29). Комбінуючи рівняння (2.23) та (2.29), можна встановити, що $\xi_i = 0$, якщо $\alpha_i < C$. Лінійне розділення точок за наявності похибки наведено на рисунку 2.9.

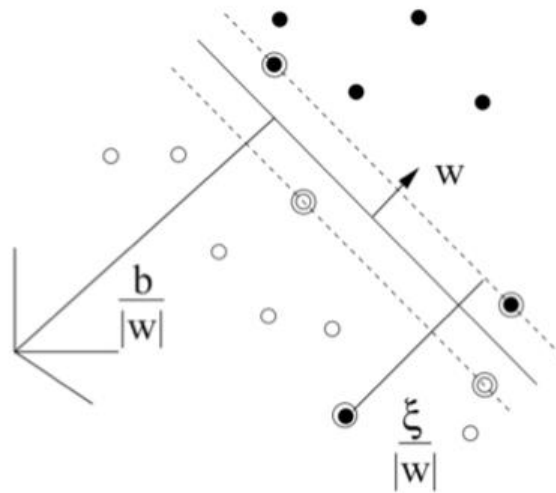


Рисунок 2.9 – Лінійне розділення точок за наявності похибки

На практиці для побудови SVM вирішують саме це завдання, так як гарантувати лінійну роздільність вибірки в загальному випадку не представляється можливим. Цей варіант алгоритму називають SVM з м'яким зазором (soft-margin SVM), тоді як в лінійно роздільному випадку говорять про SVM з жорстким зазором (hard-margin SVM) [12].

Для алгоритму класифікації використовують формулу:

$$a(x) = \text{sign}\left(\sum_{i=1}^l \lambda_i y_i \langle x_i, x \rangle - \omega_0\right). \quad (2.30)$$

За умови, що ненульовими λ_i володіють не тільки опорні об'єкти, але і об'єкти-порушники. У певному сенсі це недолік SVM, оскільки порушниками часто виявляються шумові викиди, і побудоване на них вирішальне правило, по суті справи, спирається на шум.

Константу C зазвичай вибирають за критерієм змінного контролю. Це трудомисткій спосіб, так як завдання доводиться вирішувати заново при кожному значенні C . Якщо є підстави вважати, що вибірка майже лінійно роздільна, і лише об'єкти-викиди класифікуються невірно, то можна застосувати фільтрацію викидів. Спочатку завдання вирішується при деякому C , а з вибірки видаляється невелика частка об'єктів, що мають найбільшу величину помилки. Після цього завдання вирішується заново на усіченій вибірці. Можливо, доведеться виконати кілька таких ітерацій, поки об'єкти, залишилися, не виявляться лінійно нероздільні.

Це узагальнення SVM на випадок лінійної нероздільності, тому дозволимо алгоритму допускати помилки на навчальних об'єктах, але при цьому постараємося, щоб помилок було менше.

2.3 Складність навчання методу. Порівняння результатів

Розглядаючи метод опорних векторів, варто звернути увагу на складність навчання. Час навчання в методі опорних векторів в основному визначається часом розв'язування відповідної задачі квадратичного програмування, тому теоретична й емпірична складність залежить від способу розв'язування цієї задачі. Також час навчання можна вважати часом, який метод навчання затрачає на настроювання класифікатора за допомогою множини, а час тестування – це час, що класифікатор затрачає на класифікацію одного документа. Для методу опорних векторів передбачається, що класифікація по декількох класах виробляється за допомогою сукупності бінарних класифікаторів – це середня кількість лексем на документ, а середній розмір лексикона документа (кількість ненульових ознак) і кількість лексем і різних термінів у тестовому документі відповідно. Вважається, що часова складність стандартного рішення задачі квадратичного програмування пропорційна кубу обсягу набору даних. Усі недавні роботи з методу опорних векторів спрямовані на зниження цієї складності, причому досить часто це відбувається за рахунок того, що точний розв'язок замінюється наближеним. Як правило лінійна складність традиційних алгоритмів опорних векторів ускладнює і навіть іноді унеможливорює їхнє застосування до великих наборів навчальних даних.

Альтернативні алгоритми опорних векторів, складність яких лінійно залежить від кількості навчальних вибірок, погано масштабуються для великої кількості ознак, що є характерним для задач класифікації текстів. Однак нові багатообіцяючі алгоритми навчання, засновані на методі площин, лінійно залежать від кількості навчальних

прикладів і кількості ненульових ознак у них. І все ж реальна швидкість квадратичної оптимізації набагато нижче швидкості простого підрахунку термінів у наївної байєсівської моделі. Заміна лінійного методу опорних векторів нелінійним звичайно приводить до підвищення часової складності навчання в рази (оскільки необхідно обчислювати скалярний добуток елементів навчальної множини), що зовсім неприйнятно. На практиці частіше дешевше створити ознаки більш високого порядку і провести навчання за допомогою лінійного методу опорних векторів. Складність навчання і тестування за допомогою лінійного методу опорних векторів показана в таблиці 2.2.

Таблиця 2.2 – Складність навчання і тестування різних класифікаторів

Класифікатор	Вид	Метод	Часова складність
Наївний байєсівський	Навчання		$\theta(R L_{ave} + C V)$
Наївний байєсівський	Тестування		$\theta(C M_a)$
Роккіо	Навчання		$\theta(R L_{ave} + C V)$
Роккіо	Тестування		$\theta(C M_a)$
kNN	Навчання	Попередня обробка	$\theta(R L_{ave})$
kNN	Тестування	Попередня обробка	$\theta(R M_{ave}M_a)$
kNN	Навчання	Без попередньої обробки	$\theta(1)$
kNN	Тестування	Без попередньої обробки	$\theta(R L_{ave}M_{ave})$
SVM	Навчання	Звичайний	$\theta(C R ^3M_{ave})$
SVM	Навчання	Площини, що відтиняють	$\theta(C R L_{ave}M_{ave})$
SVM	Тестування		$\theta(C M_a)$

2.4 Побудова опорних векторів для нелінійно роздільних класів

Все, що розглядалося вище, стосується лінійно розділимих задач. На практиці такий вид категоризації зустрічається, проте він є лише винятком, здебільшого категоризаційні задачі не допускають лінійного розподілення об'єктів.

Для того, щоб задача знову стала лінійно розділюмою і можна було застосовувати розглянуті вище лагранжіани, треба розмістити дані у просторі більш високого порядку, де можливе лінійне розділення об'єктів гіперплощиною. Цю операцію називають спрямленням простору або мапінгом даних, яка зображена на рисунку 2.10.

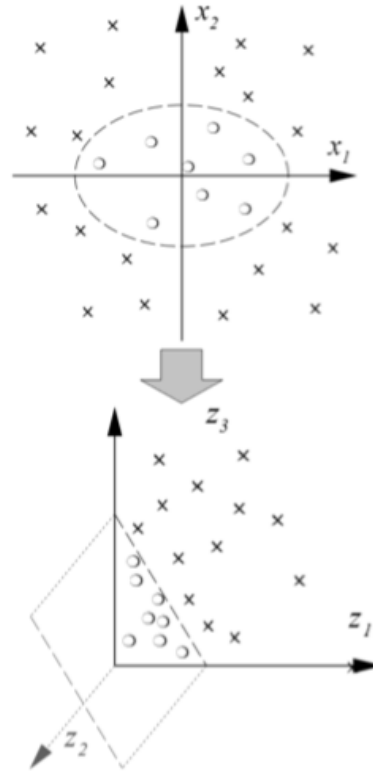


Рисунок 2.10 – Мапінг даних

Мапінг даних позначається Φ і виконує відображення даних із вихідного гіперпростору R^d до евклідового простору \aleph :

$$\Phi = R^d \mapsto \aleph. \quad (2.31)$$

Отже, для того, щоб задача стала лінійно розділюмою, до кожної з точок треба застосувати перетворення (2.31) і надалі оперувати не з самими точками, а з відповідними їм $\Phi(x)$. Логіка такого розподілення добре ілюструється на рисунку 2.11.

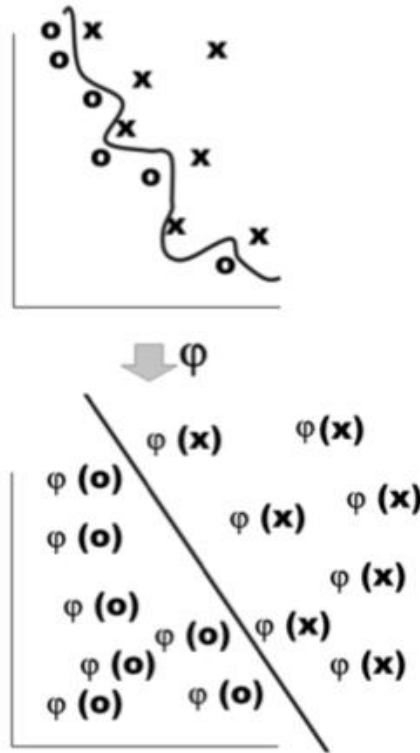


Рисунок 2.11 – Застосування $\Phi(x)$ до вихідних даних

Функції ядра та спрямлення простору. У розглянутих лагранжіанах часто застосовуються скалярні добутки точок. Для спрощення оптимізаційної задачі було розроблено спосіб, що називається *kernel trick*, в основі якого лежить перехід від скалярних добутків до так званих функцій ядра. При цьому кожен скалярний добуток замінюється нелінійною функцією ядра (скалярним добутком у просторі з більшою розмірністю). Функція ядра може бути представлена наступним чином:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j). \quad (2.32)$$

Використання функції ядра взагалі дає можливість відійти від необхідності користуватись $\Phi(x)$. Користувач методу може навіть не знати, яку $\Phi(x)$ використовує та чи інша функція ядра.

На етапі навчання SVM використовується для обчислення $f(x)$ за вихідними даними навчання з використанням формули:

$$f(x) = \sum_{i=1}^n \alpha_i y_i \Phi(x) + b = \sum_{i=1}^n \alpha_i y_i K(x_i, x) + b. \quad (2.33)$$

Можна піти ще далі і зовсім відмовитися від признакових описів об'єктів. У багатьох практичних завданнях об'єкти спочатку задаються інформацією про їх попарне взаємовідношення, наприклад, щодо схожості. Якщо ця інформація допускає зображення у вигляді двомісної функції $K(x_i, x_j)$, що задовольняє аксіомам скалярного добутку, то задача може вирішуватися шляхом SVM. Для такого підходу недавно був придуманий термін безпризнакового розпізнавання (featureless recognition), хоча давно відомі метричні алгоритми класифікації (kNN, RBF та інші), які також не вимагають завдання признакових описів.

Але спочатку треба дати визначення функції, яка невід'ємно визначена. Функція $K(x_i, x_j)$ невід'ємно визначена, якщо для будь-якої кінцевої вибірки $X^p = (x_1, \dots, x_p)$ з X матриця $K = \|K(x_i, x_j)\|$ розміром $p \times p$ невід'ємно визначена: $z^T K z \geq 0$ для будь-якого $z \in \mathbb{R}^p$. Перевірка невід'ємної визначеності функції в практичних ситуаціях може виявитися справою нетривіальною. Часто обмежуються перебором кінцевого числа функцій про які відомо, що вони є ядрами. Серед них вибирається найкраща, як правило, за критерієм змінного контролю. Очевидно, що це не оптимальне рішення. На сьогоднішній день проблема вибору ядра, оптимального для конкретного завдання, залишається відкритою. Теорема Мерсера заключається в тому, що функція $K(x_i, x_j)$ є ядром тоді і тільки тоді, коли вона симетрична $K(x_i, x_j) = K(x_j, x_i)$, і невід'ємно визначена: $\iint_{X \times X} K(x_i, x_j) g(x_i) g(x_j) dx_i dx_j \geq 0$ для будь-якої функції $g : X \rightarrow \mathbb{R}$.

Отже, якщо функція двох аргументів відповідає всім умовам теореми Мерсера, то вона може бути використана у ролі ядра. Існує декілька правил породження, які дозволяють будувати ядра в практичних завданнях:

а) довільний скалярний добуток $K(x_i, x_j) = \sum_{ij} x_i x_j$ є ядром;

б) константа $K(x_i, x_j) = 1$ є ядром;

в) добуток ядер $K(x_i, x_j) = K_1(x_i, x_j) K_2(x_i, x_j)$ є ядром;

г) для будь-якої функції Φ добуток $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ є ядром;

г) лінійна комбінація ядер з невід'ємними коефіцієнтами $K(x_i, x_j) = \alpha_1 K_1(x_i, x_j) + \alpha_2 K_2(x_i, x_j)$ є ядром;

д) композиція довільної функції $\varphi : X \rightarrow X$ і довільного ядра K_0 є ядром: $K(x_i, x_j) = K_0(\varphi(x_i), \varphi(x_j))$;

е) якщо $s : X \times X \rightarrow \mathbf{R}$ – довільна симетрична інтегрована функція, то $K(x_i, x_j) = \int_X s(x_i, z)s(x_j, z)dz$ є ядром;

є) функція виду $K(x_i, x_j) = k(x_i - x_j)$ є ядром тоді і тільки тоді, коли Фур'є-образ

$$F[k](\omega) = (2\pi)^{\frac{n}{2}} \int_X e^{-\sum_i \omega x_i} k(x) dx \text{ невід'ємний};$$

ж) межа локально-рівномірно збіжної послідовності ядер є ядром;

з) композиція довільного ядра K_0 і довільної функції $f : \mathbf{R} \rightarrow \mathbf{R}$, яка представима у вигляді статежного ряду, який сходиться, з невід'ємними коефіцієнтами $K(x_i, x_j) = f(K_0(x_i, x_j))$ є ядром. Зокрема, функції $f(z) = e^z$ і $f(z) = \frac{1}{1-z}$ від ядра є ядрами.

Існує кілька “стандартних” ядер, які при найближчому розгляді призводять до вже відомих алгоритмів: поліноміальних розділяючих поверхонь, двошарових нейронних мереж, потенційним функціям (RBF-мережам), і іншим. Таким чином, ядра претендують на роль універсальної мови для опису широкого класу алгоритмів навчання по прецедентах. Спостерігається парадоксальна ситуація. З одного боку, ядра – одне з найкрасивіших і плідних винаходів в машинному навчанні. З іншого боку, досі не знайдено ефективного загального підходу до їх підбору в конкретних завданнях.

Розглянемо приклади ядер. Найпростіше реалізувати нелінійний метод SVM на базі L_D , або так званого lagrangian trick із застосуванням kernel trick. При цьому максимізується наступне рівняння відносно нерівностей (2.11) та (2.12):

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j. \quad (2.34)$$

Функція ядра при цьому може бути представлена рівнянням:

$$K(x_i, x_j) = (x_i x_j + 1)^p, \quad (2.35)$$

де p – деякий параметр, що підлягає налаштуванню користувачем.

Ускладнимо ситуацію. Нехай $X = \mathbf{R}^2$. Розглянемо ядро $K(u, v) = \langle u, v \rangle^2$, де $u = (u_1, u_2)$, $v = (v_1, v_2)$ і спробуємо зрозуміти, який простір і перетворення ψ йому відповідають. Розкладемо квадрат скалярного добутку:

$$\begin{aligned}
K(u, v) &= \langle u, v \rangle^2 = \langle (u_1, u_2), (v_1, v_2) \rangle^2 = (u_1 v_1 + u_2 v_2)^2 = \\
&= u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 u_2 v_2 = \quad . \quad (2.36) \\
&= \langle (u_1^2 u_2^2, \sqrt{2} u_1 u_2), (v_1^2 v_2^2, \sqrt{2} v_1 v_2) \rangle
\end{aligned}$$

Ядро K представляється у вигляді скалярного добутку в просторі $H = \mathbf{R}^3$. Перетворення буде мати вигляд функції $\psi: \mathbf{R}^2 \rightarrow \mathbf{R}^3$, де $\psi: (u_1, u_2) \mapsto (u_1^2 u_2^2, \sqrt{2} u_1 u_2)$. Лінійної поверхні в просторі H відповідає квадратична поверхня в вихідному просторі X . Зокрема, дане ядро дозволяє лінійно розділити внутрішню і зовнішню частину еліпса, що неможливо в вихідному двовимірному просторі. Нехай тепер $X = \mathbf{R}^n$, $K(u, v) = \langle u, v \rangle^d$. Тоді компонентами вектора $\psi(u)$ є різні $(u_1)^{d_1}, \dots, (u_n)^{d_n}$ при всіляких цілих невід'ємних d_1, \dots, d_n , що задовольняють умові $d_1 + \dots + d_n = d$. Число таких мономів, а отже і розмірність простору H дорівнює C_{n+d-1}^d . Простір H ізоморфний простору всіх поліномів, які складаються з мономів ступеня d від змінних u_1, \dots, u_n . Якщо $X = \mathbf{R}^n$, $K(u, v) = (\langle u, v \rangle + 1)^d$, то H – простір всіх мономів ступеня не вище d від змінних u_1, \dots, u_n . В цьому випадку простір H ізоморфний простору всіх поліномів ступеня d . Лінійна роздільність множин в цьому просторі еквівалентна поліноміальній роздільності множин в початковому просторі X .

При розгляді методу опорних векторів варто звернути увагу на зв'язок SVM з двошаровими нейронними мережами. Розглянемо структуру алгоритма класифікації (2.31). Після заміни в (2.31) скалярного добутку $\langle x_i, x \rangle$ ядром $K(x_i, x)$. Наступним шагом потрібно поміняти об'єкти так, щоб перші h об'єктів виявилися опорними. Оскільки $\lambda_i = 0$ для всіх неопорних об'єктів, $i = \overline{h+1, l}$, то алгоритм (2.34) буде мати вигляд:

$$a(x) = \text{sign}\left(\sum_{i=1}^l \lambda_i y_i K(x_i, x) - \omega_0\right). \quad (2.37)$$

Якщо $X = \mathbf{R}^n$, то алгоритм $a(x)$ можна розглядати як двошарову нейронну мережу, яка схематично зображена на рисунку 2.10, що має n вхідних нейронів і h

нейронів в прихованому шарі. Мережа, налаштована методом SVM, має кілька чудових особливостей.

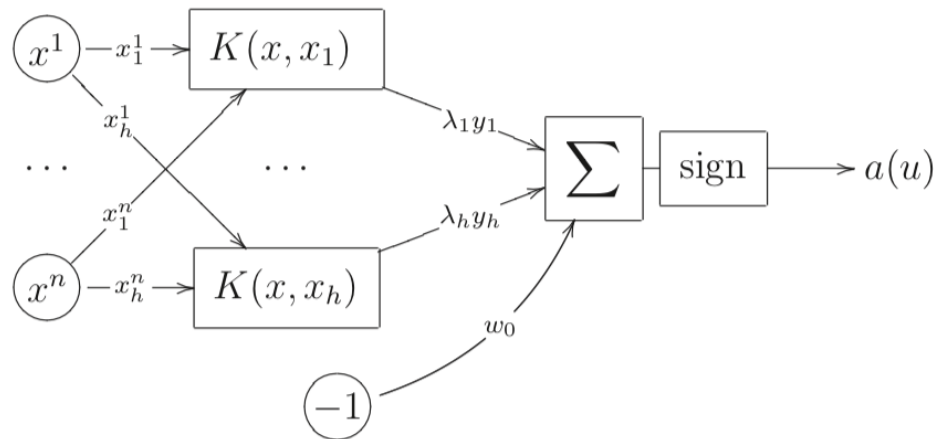


Рисунок 2.12 – Машина опорних векторів як двошарова нейромережа

По-перше, число нейронів прихованого шару визначається автоматично. По-друге, вектори ваг у нейронів прихованого шару збігаються з ознаковими описами опорних об'єктів. По-третє, пояснюється сенс двоїстих змінних: λ_i – це вага, що виражає ступінь важливості ядра $K(x_i, x)$.

Класична нейронна мережа з сигмоїдними функціями активації вийде, якщо в якості ядра взяти функцію:

$$K(u, v) = th(k_0 + k_1 \langle u, v \rangle). \quad (2.38)$$

Ця функція задовольняє умовам Мерсера не при всіх значеннях параметрів k_0 і k_1 . Зокрема, вона їм не задовольняє при $k_0 < 0$ або $k_1 < 0$. Однак це не перешкоджає її успішному практичному застосуванню. Замість гіперболічного тангенса th з часто використовують також логістичну функцію $\sigma(z) = \frac{1}{1 + e^{-z}}$.

Що поганого станеться, якщо функція $K(u, v)$ не задовольнятиме умовам Мерсера? Постановка завдання залишиться тією ж і в цьому випадку. Однак квадратична форма втратить властивість невід'ємної визначеності, функціонал, що мінімізується, вже не буде опуклим, і рішення може виявитися не єдиним. Найнеприємніше те, що на межах гіперпаралелепіпеда $0 \leq \lambda_i \leq C$ виникне величезна кількість локальних мінімумів, і пошук

рішення серед них в загальному випадку буде вимагати повного перебору. У цій ситуації багато методів квадратичного програмування видаватимуть якийсь локальний мінімум, не обов'язково хороший.

Нейронна мережа з радіальними базисними функціями (radial basis functions, RBF) вийде, якщо взяти гауссовське ядро:

$$K(u, v) = \exp(-\beta \|u - v\|^2), \quad (2.39)$$

де β - параметр.

Ядро $K(x_i, x)$ обчислює оцінку найближчої відстані об'єкта x до опорного об'єкту x_i . Чим ближче об'єкти, тим більше значення ядра. Вихідний нейрон складає всі ці оцінки, домножаючи їх на коефіцієнти $\lambda_i y_i$. При цьому найближча відстань до опорних об'єктів класу +1 підсумовуються з позитивною вагою, а до об'єктів класу -1 – з негативною. Вихідний нейрон виробляє голосування, порівнюючи сумарні відстані об'єкта, що розпізнається, x до обох класів. Існує і альтернативний метод навчання RBF-мереж, заснований на EM-алгоритмі. Тоді гауссові ядра грають роль компонент суміші імовірнісних розподілів. Центри ядер розміщуються не в опорних об'єктах, а в місцях локальних згущень щільності об'єктів. В цьому і полягає основна відмінність SVM-RBF від EM-RBF. Метод SVM зрушує центри гауссіанов ближче до меж класів, в результаті форма поверхні, що розділяє класи, описується більш чітко. Таким чином, SVM-RBF краще підходить для опису класів з межами складної форми. З іншого боку, EM-RBF більш стійкий до викидів і кращий в задачах з “розмитими” межами класів.

Аналізуючи вищесказане можна виділити переваги і недоліки даного методу. До переваг можна віднести:

- найшвидший метод знаходження вирішальних функцій;
- метод зводиться до вирішення задачі квадратичного програмування у випуклій області, яка завжди має єдине вирішення;
- метод знаходить роздільну смугу максимальної ширини, що дозволяє надалі здійснювати кращу класифікацію;
- за умови різного вибору ядер можна емулювати інші підходи. Наприклад, великий клас нейронних мереж можна представити у вигляді методу опорних векторів з визначеними ядрами;
- теоретичне обґрунтування: кінцеве правило обирається не за допомогою деяких евристик, а відповідно до оптимізації деякої функції.

До недоліків методу опорних векторів можна віднести:

- мала кількість параметрів для налаштування: після того як ядро зафіксували, єдиним варіативним параметром лишається коефіцієнт помилки C ;
- метод чутливий до шумів та стандартизації даних;
- повільне навчання;
- не існує загального підходу до автоматичного вибору ядра у випадку лінійної нероздільності класів.

2.5 Моментні інваріанти

Наступним шагом варто детальніше описати інваріанти зображень, з якими буде працювати система, що розробляється. Моменти і моментні інваріанти відіграють дуже важливу роль як ознаки інваріантного розпізнавання образів. Підхід з використанням інваріантних ознак є найбільш перспективним і широко використовується з 1970 року. Його основна ідея полягає в тому, щоб описати об'єкти набором вимірюваних величин, які називаються інваріантами, нечутливими до певних деформацій і які забезпечують достатню дискримінаційну здатність розрізняти об'єкти, що належать до різних класів.

Глобальні інваріанти, такі як інваріанти моменту, набагато надійніші, ніж локальні інваріанти по відношенню до шуму, неточне виявлення меж та інші подібні фактори порівняно з іншими інваріантами моменту. Моментальні інваріанти вперше були введені в розпізнавання і обробку зображень у 1962 р., коли Ху використовував результати теорії алгебраїчних інваріантів і вивів свої сім відомих інваріантів до обертання 2D-об'єктів.

Двовимірний геометричний момент m порядку $(p + q)^{th}$ функції $f(x, y)$ визначається як:

$$m_{pq} = \int_{a_1}^{a_2} \int_{a_1}^{a_2} x^p y^q f(x, y) dx dy, \quad (2.40)$$

де $p, q = \overline{1, \infty}$, а і x, y дає розташування пікселя на зображенні по осі x і осі y відповідно і $f(x, y)$ дає значення інтенсивності в конкретному місці.

Зауважимо, що маломісний продукт $x^p y^q$ є базовою функцією для визначення цього моменту. набір з n моментів складається з усіх m_{pq} для $p + q \leq n$, тобто набір містить $\frac{1}{2}(n+1)(n+2)$ елементів.

Використовуючи нелінійні комбінації геометричних моментів, Ху вивів набір інваріантних моментів, який має бажані властивості бути інваріантним до перекладу зображення, масштабування і обертання. Однак реконструкція зображення з цих моментів вважається досить складною.

Інваріанти моменту є дуже корисним способом для отримання ознак з двовимірних зображень. Моментальні інваріанти – це властивості зв'язних областей у бінарних образах, які інваріантні до трансляції, обертання та масштабу.

Нормалізовані центральні моменти визначаються як:

$$\eta_{pq} = \frac{\mu_{pq}}{\gamma}, \quad (2.41)$$

$$\text{де } \gamma = \frac{p+q}{2} + 1, \quad p+q = 2, 3, 4, \dots$$

Набір з семи інваріантів може бути отриманий з другого і третього нормалізованих центральних моментів:

$$\varphi_1 = \eta_{20} + \eta_{02}, \quad (2.42)$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \quad (2.43)$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \quad (2.44)$$

$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \quad (2.45)$$

$$\varphi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right], \quad (2.46)$$

$$\begin{aligned} \varphi_6 = & (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \quad (2.47)$$

$$\begin{aligned} \varphi_7 = & \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + \\ & + (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right]. \end{aligned} \quad (2.48)$$

Цей набір з семи моментних інваріантів (2.42) – (2.48) інваріантний до переклада, обертання та зміни масштабу.

3 КОМП'ЮТЕРНА МОДЕЛЬ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ МЕТОДОМ SVM

3.1 Вибір мови програмування та середовища розробки

Для розробки програмного забезпечення було обрано мобільну систему iOS та реалізація серверного додатку на мові Python з використанням бібліотеки TensorFlow, де і буде відбуватися процес класифікації зображень.

Інтерфейс iOS заснований на концепції прямої маніпуляції з використанням жестів. Елементи інтерфейсу управління складаються з повзунків, перемикачів і кнопок. Він призначений для безпосереднього контакту користувача з екраном пристрою. Внутрішній акселерометр використовується деякими програмами для реагування на струшування пристрою, яке є також загальною командою скасування, або обертати пристрій у трьох вимірах, що є загальною командою перемикання між книжковим та альбомним режимами. Cocoa Touch являє собою самий верхній шар в iOS-архітектурі. Він містить деякі ключові платформи, на яких ґрунтуються нативні додатки iOS, і однією з них є платформа UIKit. Cocoa Touch визначає основну інфраструктуру додатків і пропонує ряд життєво важливих технологій наприклад мультизадачності або введення за допомогою сенсорів.

Рідні iOS-додатки не можуть вправно виконуватись, якщо вони не прив'язані до платформ UIKit і Foundation. Платформа UIKit тісно пов'язана з iOS. Середовище розробки UIKit надає інфраструктуру для графічних iOS-додатків, керованих подіями. Тим не менш, вона також працює з іншими аспектами ядра, які властиві платформі iOS.

Apple дає можливість створювати нативні додатки тільки за допомогою середовища розробки Xcode. Xcode включає в себе велику частину документації розробника від Apple і Interface Builder – додаток, який використовується для створення графічних інтерфейсів. Xcode дозволяє нам вільно використовувати сторонні бібліотеки, наприклад, OpenCV, за допомогою технології CocoaPods або використовуючи динамічне підключення бібліотеки.

Для розширення функціоналу додатка потрібно користувачу дати можливість обробляти зображення, які зберігаються в його галереї. Для цього використовується бібліотека для отримання фотографій з пристрою або iCloud – Photo Kit. Вона виконує витяг об'єктів і з проханням про зміну, бо, по замовчуванню, вони тільки для читання, не змінюються, і містять лише метадані. Архітектура PHPhotoLibrary дозволяє легко, безпечно і ефективно працювати з тими ж активами з декількох потоків або декількох додатків. У додатку будемо використовувати клас PHAsset для ідентифікації серійної зйомки фотографій, панорамних фотографій і відео з високою частотою кадрів.

Для розробки серверного додатку було обрано мову Python. Python – мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Багата стандартна бібліотека є однією з привабливих сторін Python. Тут є засоби для роботи з багатьма мережевими протоколами і форматами Інтернету, наприклад, модулі для написання HTTP-серверів і клієнтів, для розбору і створення поштових повідомлень, для роботи з XML і т.д. Існують модулі для роботи з регулярними виразами, текстовими кодуваннями, мультимедійними форматами, криптографічними протоколами, архівами, серіалізації даних, підтримка юніт-тестування та ін.

Стандартна бібліотека Python не включає інструментів для роботи з комп'ютерним зором. Для цих потреб обрано бібліотеку OpenCV. OpenCV – бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. OpenCV написана на мові високого рівня (C/C++) і містить алгоритми для: інтерпретації зображень, калібрування камери за зразком, усунення оптичних спотворень, визначення подібності, аналізу переміщення об'єкта, визначення форми об'єкту та стеження за об'єктом, 3D-реконструкції, сегментації об'єкта, розпізнавання жестів і т.д. Також бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях. Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору.

На даний момент реалізовано інтерфейс на мовах Python, Java і MATLAB/OCTAVE. API для цих інтерфейсів можна знайти в онлайн документації. Оболонки для інших мов, таких як C#, C#, Ruby були розроблені з метою охоплення ширшої аудиторії. Всі нові розробки та алгоритми OpenCV у даний момент розробляються у C++ інтерфейсі.

Зважаючи на таку потужність та зручність бібліотеки та велику кількість позитивних сторін був обраний саме цей інструмент.

Для машинного навчання була обрана бібліотека TensorFlow. TensorFlow – відкрита програмна бібліотека для машинного навчання, розроблена компанією Google для вирішення завдань побудови і тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття. Застосовується як для досліджень, так і для розробки власних продуктів Google. Основний

API для роботи з бібліотекою реалізований для Python, також існують реалізації для C++, Haskell, Java, Go і Swift. Спочатку TensorFlow була розроблена командою Google Brain для внутрішнього використання в Google, в 2015 році система була переведена в вільний доступ з відкритою ліцензією Apache 2.0.

TensorFlow має важливі характеристики, завдяки яким він користується такою популярністю:

- дуже потужний фреймворк для навчання нейронних мереж;
- дуже велика база методів машинного навчання;
- зручний у використанні, має простий інтерфейс для виконання поставленої функції;
- можливо створювати нейронні мережі ліній за лінією;
- програмний інтерфейс доступен для популярних мов програмування;
- є оптимізована версія для малопотужних девайсів.

3.2 Схематична структура комп'ютерної моделі

Запропонована система класифікації зображення наведена нижче на рисунку 3.1. Система реалізована у вигляді двох фаз. Під час першої фази система навчається з декількома наборами тренувальних зображень. Спочатку тренувальні зображення попередньо обробляються для отримання точної інформації для отримання ознак. На етапі попередньої обробки шум у зображенні видаляється, а заточування робиться для зменшення ефекту освітленості та відсутності контрасту на навчальних зображеннях. Після попередньої обробки тренувальне зображення застосовується для процесу виявлення інваріантів. У процесі отримання ознак витягуються інваріанти моменту для кожного з попередньо оброблених навчальних зображень. Отримання ознак визначається як процес перетворення отриманого зображення в унікальну, відмітну і компактну форму. Обчислені моментні інваріанти для всіх попередньо оброблених навчальних зображень розташовані таким чином, щоб побудувати вектор ознак. Побудований вектор ознак є високовимірним. Щоб зменшити високий розмір даних до низького розміру без втрати важливих властивостей, виконується аналіз основних компонентів ядра. Це призводить до опорного вектору, який може бути використаний для класифікатора Support Vector Machine.

Під час етапу тестування, як тільки тестове зображення подається як вхідний сигнал до запропонованої системи, процес попередньої обробки та отримання ознак

виконуються, як зазначено у фазі навчання. Обчислюваний вектор ознак задається в якості вхідних даних для класифікатора машин опорного вектора, на основі опорного вектора, сформованого під час фази навчання; вхідне зображення розпізнається і позначено. Щоб порівняти результати класифікатора Support Vector, нейронна мережа зворотного розповсюдження навчається з вектором ознак і тестується.

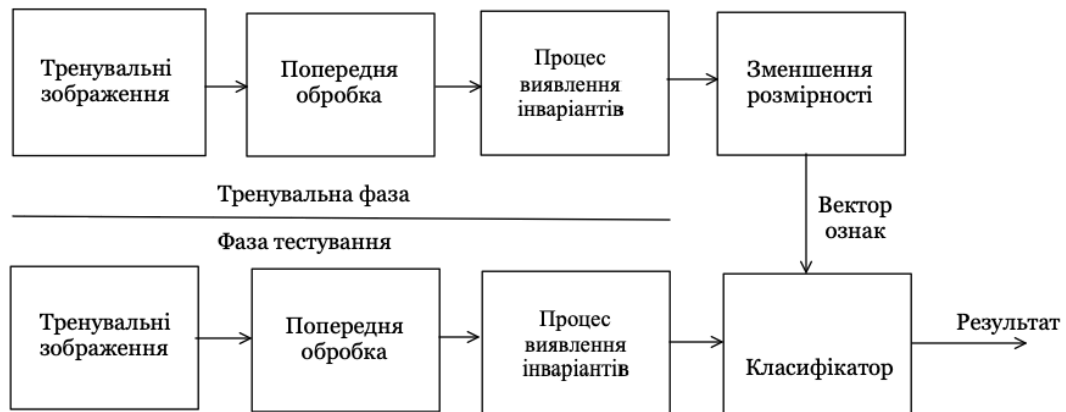


Рисунок 3.1 – Схематична модель запропонованої системи

3.3 Ілюстрація роботи програми

Під час виконання роботи була розроблена комп'ютерна система, яка демонструє класифікацію зображення за допомогою методу машинного навчання SVM (дод. А). Ця система складається з iOS додатку й сервера, який був реалізований на Python з використанням бібліотек TensorFlow та OpenCV.

Програму потрібно запускати у середовищі XCode, яка може бути встановлена лише у системі macOS. Користувач повинен відкрити файл під назвою Image Classification.xcodeproj, обрати симулятор, на якому він хоче запустити додаток, й натиснути кнопку старт. Одразу відкривається головне вікно програми, яке показано на рисунку 3.2, на якому є дві функціональні кнопки та області, одна з яких для зображення, яке обрав користувач.

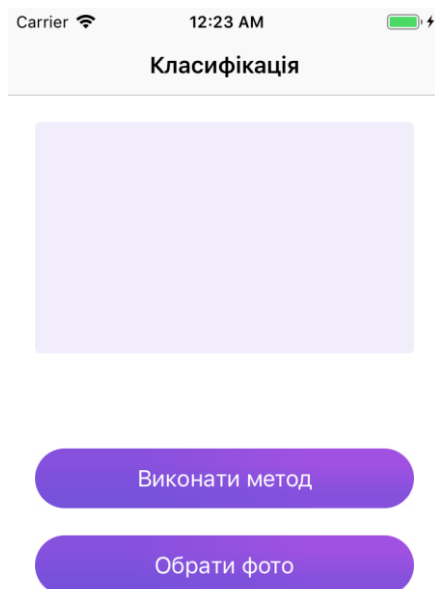


Рисунок 3.2 – Головне вікно програми

Програмний додаток дозволяє обрати зображення з галереї чи зробити знімок. На рисунку 3.3 показано як користувач може це зробити.

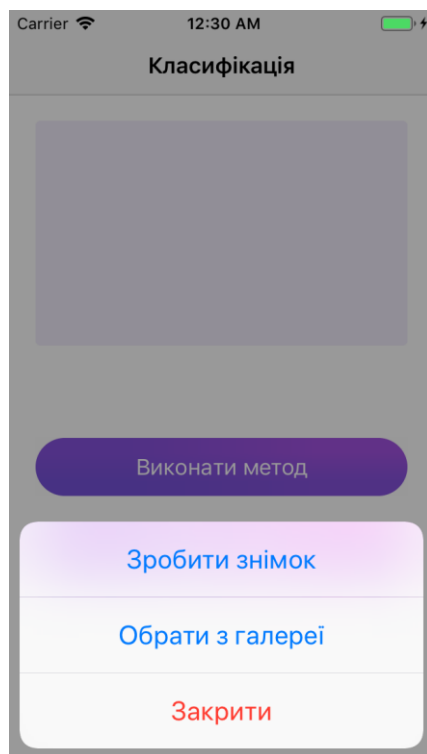
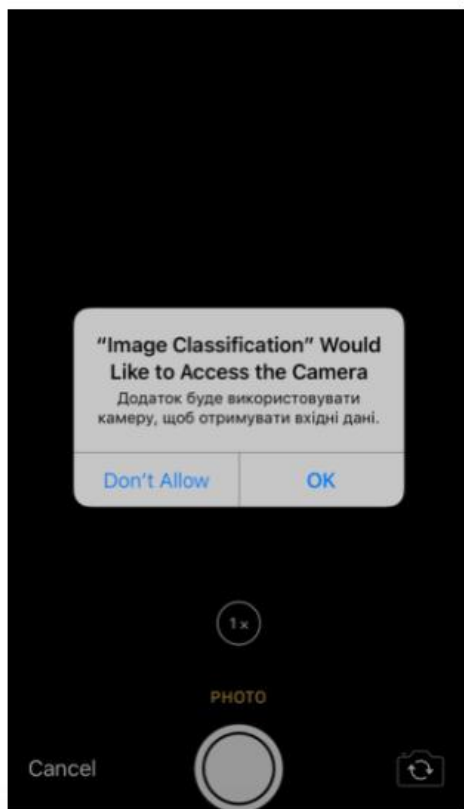


Рисунок 3.3 – Варіанти для вибору зображення

Для того, щоб зробити знімок, йому потрібно дати дозвіл на використання камери додатком, після цього він може робити знімки. Цей процес зображен на рисунку 3.4. Також користувач має можливість обрати зображення з галереї, де зберігаються його фото, скріншоти, зображення з інших додатків.



а



б

Рисунок 3.4 – Процес обрання зображення за допомогою камери:

а – дозвіл на використання камери; б – вікно камери

Для того, щоб обрати зображення з галереї, потрібно натиснути на кнопку “Обрати з галереї” і стануть доступні фото (рис. 3.5).

Після того, як користувач обрав зображення, він має можливість замінити його новим, або почати процес класифікації.

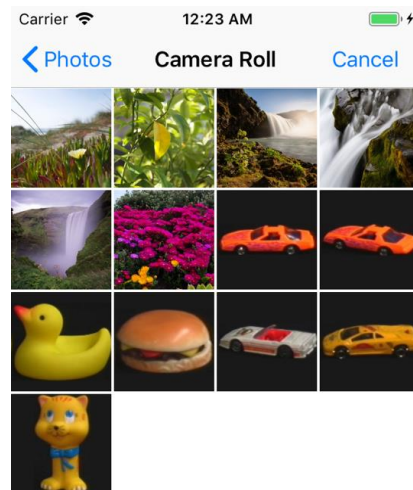


Рисунок 3.5 – Галерея користувача

Для цього потрібно натиснути на кнопку “Виконати метод” і стане доступним індикатор прогресу, який зображен на рисунку 3.6.

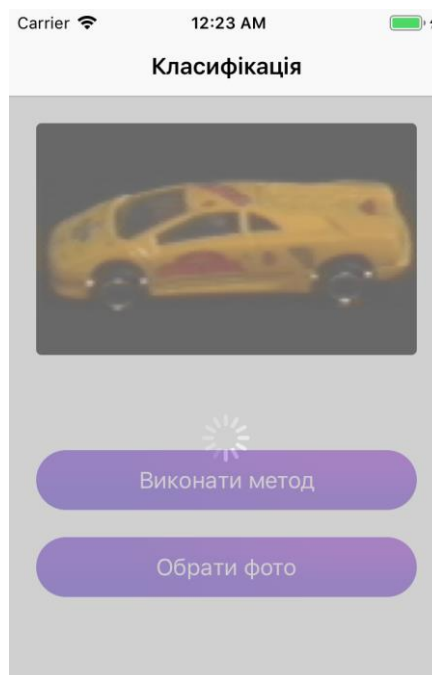


Рисунок 3.6 – Процес класифікації зображення

В цей час додаток згенерує POST запит до нашого серверу, в тіло якого він додасть бінарну інформацію вибраного зображення, і після невеликого очікування отримає повідомлення про закінчення процесу класифікації і результати, або помилку, наприклад, якщо раптом мобільний пристрій користувача втратив зв'язок з інтернетом. Результат класифікації буде показано на екрані, як це зображено на рисунку 3.7.

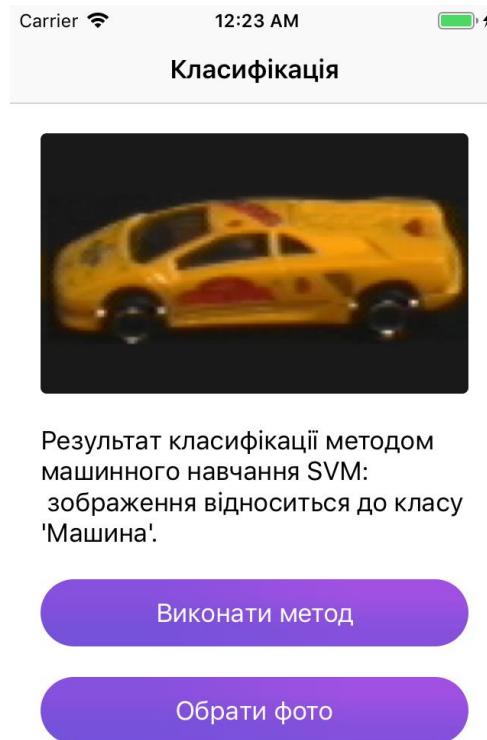


Рисунок 3.7 – Результат виконання запиту класифікації

3.4 Результати роботи системи класифікації зображень

Для експерименту була використана база даних COIL-100, яка широко використовується в дослідженнях 3D розпізнавання об'єктів. Ця база даних складається з зображень 100 різних об'єктів; кожен з них повертається з інтервалом 5 градусів у вертикальній осі. Отже, для кожного об'єкта є 72 зображення, які складають до 7200 зображень для всієї бази даних. Вся база даних COIL-100 розділена на дві групи: одна - навчальна, а друга – тестова (дод. Б). Три різні набори тренувань формуються для трьох різних кутів вибірки (10, 30 і 50 градусів). Приклади зображень наведені у таблиці 3.1. Запропонована система буде експериментуватися з кожним набором навчальних і тестових зображень.

Таблиця 3.1 – Приклади тестових зображень

Оригінальне зображення	Кут 10 градусів	Кут 30 градусів	Кут 50 градусів
			
			
			
			
			

Для того щоб зробити висновок о якості системи потрібно зробити декілька експериментів та порівняти з іншими методами. Було обрано метод зворотнього поширення помилки у якості системи, з якою будемо порівнювати результати. Це нейронна мережа, що складається з щонайменше трьох шарів одиниць: вхідний шар, щонайменше один проміжний прихований шар і вихідний шар. Вихід мережі зворотного розповсюдження розглядається як рішення щодо класифікації. Метою використання

нейромережі в даному дослідженні є прийняття характеристик запам'ятовування і посилення на властивості, які розпізнають тестування 2D-зображення.

Для оцінки точності розпізнавання запропонованих методів і традиційних методів були визначені правильні відсотки розпізнавання (CRP):

$$CRP = \frac{N_p}{T}, \quad (3.1)$$

де N_p – кількість позитивного розпізнавання, а T – загальна кількість проведених тестів.

Після проведення тестів отримано деякі результати для кожного з набору зображень. У таблиці 3.2 наведені результати продуктивності класифікації.

Таблиця 3.2 – Результати продуктивності класифікації

Метод	CRP, %			
	Кут 10 градусів	Кут 30 градусів	Кут 50 градусів	Усі кути (10,30,50)
SVM	86.4	85.5	87.4	88.3
BPN	78	77.5	78.1	78.5

Також варто відзначити результати якості системи, коли зростала кількість зразків, результати наведені у таблиці 3.3.

Таблиця 3.3 – Результати якості системи при різній кількості зразків

Кількість зразків	CRP, %	
	SVM	BPN
25	80	75
50	83.5	76.4
75	84.9	78.4
100	85.4	78.7
125	88.7	79.4

З проведеного аналізу можна зробити висновок, що метод опорних векторів може вдало використовуватися для рішення задачі класифікації зображень. Хоча він і має деякі недоліки, наприклад, повільне навчання, чутливий до шумів, але при цьому він видає дуже якісні результати на цих вибірках і може здобувати ще кращих результатів за допомогою постійного експерименту з ядрами функцій.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи було дослідження питання класифікації інваріантів зображень за допомогою методу машинного навчання SVM (метод опорних векторів) та створення системи класифікації зображень, яку можна буде використовувати у якості мобільного додатку. Так як в процесі проектування використовувалося комп'ютерне обладнання, то аналіз потенційно небезпечних і шкідливих чинників виконується для персонального комп'ютера, на якому буде виконуватися розробка.

4.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [1] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів шиї, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці. Забезпечення цих умов покладається на власника або уповноважений ним орган (далі роботодавець). Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва,

стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Роботодавець повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизмові, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників. Він не має права вимагати від працівника виконання роботи, поєднаної з явною небезпекою для життя, а також в умовах, що не відповідають законодавству про охорону праці. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація, небезпечна для його життя чи здоров'я або людей, які його оточують, і навколишнього середовища.

4.2 Аналіз стану умов праці

Робота над створенням системи класифікації зображень проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	5
Висота, м	2.8
Площа, м ²	25
Об'єм, м ³	70

Згідно з [2] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [3] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Приміщення знаходиться на другому поверсі трьох поверхової будівлі і має об'єм 70 м³, площу – 25 м². Обладнано одне місце праці укомплектоване ПК.

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум знаходиться на рівні 50 дБА. Система вентилявання приміщення — природна неорганізована, а опалення — централізоване.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	750	не менше 600
Ширина простору для ніг, мм	530	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	450	400 ÷ 500
Ширина сидіння, мм	450	не менше 400
Глибина сидіння, мм	470	не менше 400
Висота поверхні спинки, мм	400	не менше 300
Ширина опорної поверхні спинки, мм	400	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	720	700 ÷ 800

4.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту.

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви програм тривалістю 15 хв. через кожну годину роботи.

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-7.15-18 [6] «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Основними робочими характеристиками персонального комп'ютера є робоча напруга $U=+220\text{В} \pm 5\%$; робочий струм $I=2\text{А}$; споживана потужність $P=350\text{ Вт}$.

Робоче місце має відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [3].

Таблиця 4.3 – Аналіз небезпечних і шкідливих факторів

Небезпечні і шкідливі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
виробничі			
фізичні			

Продовження таблиці 4.3

- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	порушення умов праці (вимог до приміщень)	4	[4]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[5]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[5]
Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кіл ькісна оцінка	Норм ативні документи
психофізіологічні:			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів;	4	[6] [3]
	- виконання роботи за темою диплома, тестування; - оформлення роботи		
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці- сидіння користувача,) та організації робочого часу - безпервна робота)	2	[6] [3]

4.3.2 Пожежна безпека

Небезпека розвитку пожежі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому проектом передбачаються засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання.

Згідно ДБН В.2.5-28:2018 [5] таке приміщення, площею 25 м², відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому можливо встановлення автоматичної пожежної сигналізації із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол.

4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник,

який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення мікроклімату для робочого місця відповідають ДСН 3.3.6.042-99 [2] (табл. 4.4):

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С0	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату. Дане приміщення обладнане системою опалення, кондиціонування повітря. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

Рівні позитивних і негативних іонів у повітрі мають відповідати ДСН 3.3.6.042-99 [2].

4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2018 [5]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше $1/8$, в побутових – $1/10$:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10} \right) \cdot S_n, \quad (4.1)$$

де S_b – площа віконних прорізів, м²;

S_n – площа підлоги, м².

$$S_n = a \cdot b = 5 \cdot 5 = 25 \text{ м}^2,$$

$$S = 1/8 \cdot 25 = 3,125 \text{ м}^2.$$

Приймаємо 2 вікна площею $S=1,6$ м² кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, м²; $S = 25$ м²;

Z – поправочний коефіцієнт світильника ($Z=1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним $1,1$;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – $1,5$;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – $0,575$

M – число люмінесцентних ламп в світильнику – 2 ;

F – світловий потік лампи – 5400 лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \cdot 25 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 2,0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.4.3 Шум та вібрація, електромагнітне випромінювання

Рівень шуму, зумовлений як роботою системного блоку, клавіатури, так і друкуванням на принтері, а також зовнішніми чинниками, коливається у межах 50–65 дБА ДСН 3.3.6.042-99 [2].

Віброізоляцію можливо здійснювати за допомогою спеціальної прокладки під системний блок, яка послаблює передачу вібрацій робочого столу. Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам ДСН 3.3.6.042-99 [2].

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання передбачено наступні заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-втяжну вентиляцію або кондиціонування повітря.

Крім того, потрібно дотримуватися правил безпеки під час експлуатації інших електричних приладів та вимоги безпеки при надзвичайних ситуаціях.

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом, приміщення в якому проводиться робота відноситься до першого класу (без підвищеної небезпеки). Коефіцієнт використання вертикальних заземлювачів η_v в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c .

Послідовність розрахунку.

Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (4.3)$$

де $R_{пр.з.}$ – опір природних заземлювачів; R_d – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_d$.

Підставивши числові значення у формулу (4.3), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м. Приблизне значення питомого опору глини приймаємо $\rho = 40$ Ом·м (табличне значення).

Розрахунковий питомий опір ґрунту, $\rho_{розр.}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в.}$, і горизонтальних $\rho_{розр.г.}$, Ом·м за формулою:

$$\rho_{розр.} = \psi \cdot \rho, \quad (4.4)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{розр.в.} = 1,7$ і горизонтальних $\rho_{розр.г.} = 5,5$ Ом·м.

$$\rho_{розр.в.} = 1,7 \cdot 40 = 68 \text{ Ом} \cdot \text{м}$$

$$\rho_{розр.г.} = 5,5 \cdot 40 = 220 \text{ Ом} \cdot \text{м}$$

Розраховується опір розтікання струму вертикального заземлювача R_v , Ом, за формулою (4.5).

$$R_v = \frac{\rho_{розр.в.}}{2 \cdot \pi \cdot l_b} \cdot \left(\ln \frac{2 \cdot l_b}{d_{ст}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_b}{4 \cdot t - l_b} \right), \quad (4.5)$$

де l_b – довжина вертикального заземлювача (для труб - 2–3 м; $l_b=3$ м);

$d_{ст}$ – діаметр стержня (для труб - 0,03–0,05 м; $d_{ст}=0,05$ м);

t – відстань від поверхні землі до середини заземлювача, яка визначається за формулою (4.6):

$$t = h_b + \frac{l_b}{2}, \quad (4.6)$$

де h_b – глибина закладання вертикальних заземлювачів (0,8 м); тоді $t = 0,8 + \frac{3}{2} = 2,3$

М

$$R_b = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання η_b :

$$n = \frac{2 \cdot R_b}{R_d} = \frac{2 \cdot 18,5}{4} = 9,25 \quad (4.7)$$

Γ визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_b = 0,57$ (табличне значення).

Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання η_b , шт:

$$n_b = \frac{2 \cdot R_b}{R_d \cdot \eta_b} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \quad (4.8)$$

Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 \cdot L_b \cdot (n_b - 1), \quad (4.9)$$

де L_b – відстань між вертикальними заземлювачами, (прийняти за $L_b = 3$ м);

n_b – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_g , Ом:

$$R_{\Gamma} = \frac{\rho_{\text{розр.}\Gamma}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_{\Gamma}}, \quad (4.10)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15$ м;

h_{Γ} – глибина закладання горизонтальних заземлювачів (0,5 м);

l_c – довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_{\Gamma} = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

Визначається коефіцієнт використання горизонтального заземлювача η_c . відповідно до необхідної кількості вертикальних заземлювачів пв.

Коефіцієнт використання з'єднувальної смуги $\eta_c = 0,3$ (табличне значення).

Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг}} = \frac{R_B \cdot R_{\Gamma}}{R_B \cdot \eta_c + R_{\Gamma} \cdot n_B \cdot \eta_B} \leq R_d. \quad (4.18)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{заг}} < 4$ Ом, а саме:

$$R_{\text{заг}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_d$$

4.6 Екологія

Діяльність на тему магістерської роботи, а саме: розробка методів кластеризації структурних описів зображень в процесі її виконання, впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [7], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення» [8], Законом України «Про відходи» [9].

В процесі розробки виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки.
- Змінні носії інформації - IV клас небезпеки.
- Відпрацьовані вогнегасники - IV клас небезпеки.
- Макулатура - IV клас небезпеки.

4.7 Висновки до розділу 4

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в дипломній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника.

А також визначені основні екологічні аспекти впливу на навколишнє природне середовище та зазначені заходи щодо поводження з ними.

ВИСНОВКИ

Під час виконання магістерської роботи був розроблений та реалізований метод класифікації зображень SVM. Для цього було доліджено математичні моделі класифікації та метод опорних векторів, розглянуті правила побудови опорних векторів у випадках лінійних та нелінійних даних. Звертаючи увагу на те, що в сучасному світі більше доводиться обробляти нелінійні дані, то в роботі були досліджені функції ядра при побудові опорного вектора. Також були досліджені інваріантні моменти, завдяки яким будується вектор ознак, що використовується безпосередньо SVM.

Для тестування реалізації методу була знайдена тренувальна база зображень, реалізован сам метод, як одна з частин системи, що допомагає класифікувати зображення. Проведено порівняльний аналіз, за результати якого побудували таблиці зі значеннями якості й точності розпізнавання. Результати експерименту виявили, що представлена система виконує свої функції і може бути використана для класифікації зображень.

При написанні роботи придбані навички розробки і практичної реалізації методу класифікації зображень, їх попередньої обробки та оцінки ефективності. В процесі реалізації були використані такі технології як Swift, OpenCV, Python, TensorFlow.

Проблема класифікації зображень є актуальною, бо з розвитком технологій з'являються нові задачі, які потребують якісних методів класифікації зображень, наприклад, блокування контенту у соціальних мережах, який може бути неприємним для інших користувачів, на етапі створення посту. Спираючись на результати роботи програми, SVM показав гарні результати і може бути рекомендован для вирішення таких проблем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Bamieh B. A General Moment Invariants Attributed- Graph Method for Three-Dimensional Object Recognition from a Single Image [Text] / B. Bamieh, R. Figueiredo // IEEE Journal of Robotics and Automation – 1986. – p. 31 – 41.
2. Bartlett P. Generalization performance of support vector machines other pattern classifiers [Text] / P. Bartlett – MIT Press, Cambridge, 1998. – p. 1230.
3. Bishop C.M. Pattern Recognition and Machine Learning [Text] / C.M. Bishop. – NY, Springer, 2006. – p. 738.
4. Burges C. C. Geometry and invariance in kernel-based methods [Text] / C. C. Burges – MIT Press, 1999. – p. 89 – 116.
5. Chapelle O. Training a Support Vector Machine in the Primal [Text] / O. Chapelle – Neural Computation, 2007. – p. 1155 – 1178.
6. Haykin S. Neural networks and learning machines [Text] / S. Haykin – NJ, Pearson, 2009. – p. 938.
7. Hu M. K. Visual Pattern Recognition by Moment Invariants [Text] / M. K. Hu // IEEE Transactions on Information Theory – 1962. – p. 179 – 187.
8. Hu W. Deep convolutional neural networks for hyperspectral image classification [Text] / W. Hu, Y. Huang, L. // Journal of Sensors. – 2015. – p. 1 – 13.
9. Maulik U. Remote Sensing Image Classification: A survey of support-vector-machine-based advanced techniques [Text]: tech.rep. / U. Maulik, D. Chakraborty – 2017. – p. 33–52.
10. Muller K. An introduction to kernel-based learning algorithms [Text]: book / K. Muller, S. Mika – Neural Networks, 2001. – p. 181 – 201.
11. Platt J.C. Fast training support vector machines using sequential minimal optimization [Text] / J.C. Platt – MIT Press, 1999. – p. 185 – 208.
12. Scheinberg K. An efficient implementation of an active set method for SVM [Text] / J. Mach – Learn Press, 2006. – p. 2237 – 2257.
13. Shawe-Taylor J. An introduction to Support Vector Machines and other kernel-based learning methods [Text] / J. Shawe-Taylor – Cambridge University Press, 2000. – 204 p.
14. Shawe-Taylor J. Robust bounds on generalization from the margin distribution [Text]: tech. rep. / J. Shawe-Taylor, N. Cristianini – Royal Holloway, University of London, 1998. – p. 147.

15. Smola A. A tutorial on support vector regression [Text]: tech. rep. / A. Smola, B. Schoelkopf – 1998. – p. 48.
16. Smola, A. J. On a kernel-based method for pattern recognition, regression, approximation and operator inversion [Text] / A. J. Smola – Algorithmica, 1998. – p. 211 – 231.
17. Steinwar I. Support Vector Machines [Text] / I. Steinwar – Springer-Verlag, New York, 2008. – 602p.
18. Vapnik, V. Bounds on error expectation for support vector machines [Text] / V. Vapnik – Neural Computation, 2000. – p. 2013 – 2036.
19. Введение в информационный поиск [Текст]: учеб. пособие / К. Д. Маннинг, П. Рагхаван, Х. Шютце – М.: Изд-во “Вильямс”, 2011. – с. 528.
20. Вьюгин В. Математические основы теории машинного обучения и прогнозирования [Текст] / В. Вьюгин – МЦМНО, 2013 – 390 с.
21. Воронцов К.В. Математические методы обучения по прецедентам (теория обучения машин) [Текст] / – 2012 – с. 67 – 106.
22. Гонсалес Р. Цифровая обработка изображений [Текст] / Р. Гонсалес, Р. Вудс; перевод с англ. – М.: Техносфера, 2005. – 1072 с.
23. Шапиро Л. Компьютерное зрение [Текст] / Л. Шапиро, Дж. Стокман; перевод с англ. – М.: БИНОМ. Лаборатория знаний, 2006. – 752 с.
24. Державні санітарні норми. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» - Режим доступу: <https://zakon.rada.gov.ua/rada/show/va042282-99> - 01.02.1999 р.
25. Державні санітарні правила і норми. ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» - Режим доступу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> - 10.12.1998 р.
26. Закон України «Про охорону праці». Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12> - 10.14.1992 р.
27. Державний стандарт України. ДСТУ Б В.2.5-82:2016 «Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом» - Режим доступу: <http://epicentre.co.ua/dstu/doc28522.html> - 01.07.2016 р.
28. Державні будівельні норми. ДБН В.2.5-28:2018 «Природне і штучне освітлення» - Режим доступу: <http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf> - 03.10.2018

29. Нормативно-правовий акт з охорони праці. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» - Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18> - 14.02.2018 р.

30. Закон України «Про охорону навколишнього природного середовища» - <https://zakon.rada.gov.ua/laws/main/1264-12> - 12.10.2018р.

31. Закон України «Про забезпечення санітарного та епідемічного благополуччя населення» - <https://zakon.rada.gov.ua/laws/main/4004-12>- 04.10.2018р.

32. Закон України «Про відходи» - <https://zakon.rada.gov.ua/laws/main/187/98-%D0%B2%D1%80> - 01.05.2019р.

ДОДАТОК А
Лістинг “svm.py”

```

1     import numpy as np
2     from sklearn.base import BaseEstimator, ClassifierMixin
3     from sklearn.metrics.pairwise import pairwise_kernels
4     from sklearn.utils import check_array, check_random_state
5     from . import _svm
6     class SVM(BaseEstimator, ClassifierMixin):
7         """Support vector machine (SVM).
8         This is a binary SVM and is trained using the SMO algorithm.
9         Reference: "The Simplified SMO Algorithm"
10        (http://math.unt.edu/~hsp0009/smo.pdf)
11        Based on Karpathy's svm.js implementation:
12        https://github.com/karpathy/svmjs
13        Parameters
14        -----
15        C : float, optional (default: 1)
16            Penalty parameter C of the error term.
17        kernel : string, optional (default: 'rbf')
18            Specifies the kernel type to be used in the algorithm.
19            It must be one of 'linear', 'poly', 'rbf', 'sigmoid',
20 'precomputed' or
21            a callable.
22            If none is given, 'rbf' will be used. If a callable is
23 given it is
24            used to pre-compute the kernel matrix from data
25 matrices; that matrix
26            should be an array of shape ``(n_samples, n_samples)``
27        degree : int, optional (default: 3)
28            Degree of the polynomial kernel function ('poly').
29            Ignored by all other kernels.
30        gamma : float, optional (default: 1)
31            Parameter for RBF kernel
32        coef0 : float, optional (default: 0.0)
33            Independent term in kernel function.
34            It is only significant in 'poly' and 'sigmoid'.
35        tol : float, optional (default: 1e-4)
36            Numerical tolerance. Usually this should not be modified.

```

```

37         alphas : float, optional (default: 1e-7)
38         Non-support vectors for space and time efficiency are
39 truncated. To
40         guarantee correct result set this to 0 to do no
41 truncating. If you
42         want to increase efficiency, experiment with setting this
43 little
44         higher, up to maybe 1e-4 or so.
45         max_iter : int, optional (default: 10000)
46         Maximum number of iterations
47         numpasses : int, optional (default: 10)
48         How many passes over data with no change before we halt?
49 Increase for
50         more precision.
51         random_state : int or RandomState instance or None (default)
52         Pseudo Random Number generator seed control. If None, use
53 the
54         numpy.random singleton. Note that different
55 initializations
56         might result in different local minima of the cost
57 function.
58         verbose : int, optional (default: 0)
59         Verbosity level
60 Attributes
61 -----
62         support_vectors_ : array-like, shape = [n_support_vectors,
63 n_features]
64         Support vectors.
65         coef_ : array, shape = [n_features]
66         Weights assigned to the features (coefficients in the
67 primal
68         problem). This is only available in the case of a linear
69 kernel.
70         `coef_` is a readonly property derived from `dual_coef_`
71 and
72         `support_vectors_`.
73         intercept_ : float
74         Constant in decision function.
75 """

```

```

76         def __init__(self, C=1.0, kernel="rbf", degree=3, gamma=1.0,
77 coef0=0.0,
78                             tol=1e-4,          alphasol=1e-7,          maxiter=10000,
79 numpasses=10,
80                             random_state=None, verbose=0):
81             self.C = C
82             self.kernel = kernel
83             self.degree = degree
84             self.gamma = gamma
85             self.coef0 = coef0
86             self.tol = tol
87             self.alphasol = alphasol
88             self.maxiter = maxiter
89             self.numpasses = numpasses
90             self.random_state = random_state
91             self.verbose = verbose
92     def fit(self, X, y):
93         """Fit the model to data matrix X and target y.
94         Parameters
95         -----
96         X : array-like, shape (n_samples, n_features)
97             The input data.
98         y : array-like, shape (n_samples,)
99             The class labels.
100         Returns
101         -----
102         self : returns a trained SVM
103         """
104         self.support_vectors_ = check_array(X)
105         self.y = check_array(y, ensure_2d=False)
106         random_state = check_random_state(self.random_state)
107         self.kernel_args = {}
108         if self.kernel == "rbf" and self.gamma is not None:
109             self.kernel_args["gamma"] = self.gamma
110         elif self.kernel == "poly":
111             self.kernel_args["degree"] = self.degree
112             self.kernel_args["coef0"] = self.coef0
113         elif self.kernel == "sigmoid":
114             self.kernel_args["coef0"] = self.coef0

```

```

115
116         K         =         pairwise_kernels(X,         metric=self.kernel,
117 **self.kernel_args)
118         self.dual_coef_ = np.zeros(X.shape[0])
119         self.intercept_ = _svm.smo(
120             K,     y,     self.dual_coef_,     self.C,     random_state,
121 self.tol,
122             self.numpasses, self.maxiter, self.verbose)
123         # If the user was using a linear kernel, lets also
124 compute and store
125         # the weights. This will speed up evaluations during
126 testing time.
127         if self.kernel == "linear":
128             self.coef_ = np.dot(self.dual_coef_ * self.y,
129 self.support_vectors_)
130             # only samples with nonzero coefficients are relevant for
131 predictions
132             support_vectors = np.nonzero(self.dual_coef_)
133             self.dual_coef_ = self.dual_coef_[support_vectors]
134             self.support_vectors_ = X[support_vectors]
135             self.y = y[support_vectors]
136             return self
137     def decision_function(self, X):
138         """Decision function of the SVM.
139         Parameters
140         -----
141         X : array-like, shape (n_samples, n_features)
142             The input data.
143         Returns
144         -----
145         y : array-like, shape (n_samples,)
146             The values of decision function.
147         """
148         X = check_array(X)
149         if self.kernel == "linear":
150             return self.intercept_ + np.dot(X, self.coef_)
151         else:
152             K         =         pairwise_kernels(X,         self.support_vectors_,
153 metric=self.kernel,

```

```
154                                     **self.kernel_args)
155             return                    (self.intercept_                    +
156 np.sum(self.dual_coef_[np.newaxis, :] *
157                                     self.y * K, axis=1))
158     def predict(self, X):
159         """Predict class labels.
160         Parameters
161         -----
162         X : array-like, shape (n_samples, n_features)
163             The input data.
164         Returns
165         -----
166         y : array-like, shape (n_samples,)
167             The predicted classes.
168         """
169     return np.sign(self.decision_function(X))
```

ДОДАТОК Б

Лістинг “test_svm.py”

```

1     import numpy as np
2     from svm import SVM
3     from sklearn.metrics import accuracy_score
4     from numpy.testing import assert_array_almost_equal
5     from nose.tools import assert_true, assert_greater
6     def test_linear_svm():
7         random_state = np.random.RandomState(0)
8         X = random_state.randn(10, 2)
9         b = -0.2
10        w = np.array([0.5, -0.3])
11        y = np.sign(b + np.dot(X, w))
12        svm = SVM(kernel="linear", C=1.0, random_state=random_state)
13        svm.fit(X, y)
14        y_pred = svm.predict(X)
15        assert_array_almost_equal(y, y_pred)
16        assert_true(hasattr(svm, "coef_"))
17        assert_true(hasattr(svm, "intercept_"))
18    def test_rbf_svm():
19        random_state = np.random.RandomState(0)
20        n_samples = 100
21        X = np.empty((n_samples, 2))
22        X[:, 0] = np.linspace(0, 1, n_samples)
23        X[:, 1] = random_state.randn(n_samples)
24        y = np.sign(X[:, 1] - np.sin(2.0 * np.pi * np.sin(X[:, 0])))
25        svm = SVM(kernel="rbf", C=1.0, gamma=1.0,
26 random_state=random_state)
27        svm.fit(X, y)
28        y_pred = svm.predict(X)
29        assert_greater(accuracy_score(y, y_pred), 0.9)
30    def test_poly_svm():
31        random_state = np.random.RandomState(0)
32        n_samples = 100
33        X = np.empty((n_samples, 2))
34        X[:, 0] = np.linspace(0, 1, n_samples)
35        X[:, 1] = random_state.randn(n_samples)
36        y = np.sign(X[:, 1] - np.sin(2.0 * np.pi * np.sin(X[:, 0])))

```

```
37 svm = SVM(kernel="poly", C=1.0, degree=3, coef0=1.0,  
38           random_state=random_state)  
39 svm.fit(X, y)  
40 y_pred = svm.predict(X)  
41 assert_greater(accuracy_score(y, y_pred), 0.9)
```


ДОДАТОК В

Електронна презентація

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

МАГІСТЕРСЬКА РОБОТА

на тему:

«Дослідження методів класифікації зображень
та реалізація методу опорних векторів»

Студента групи KI-18дм

Білова Владислава Віталійовича

Науковий керівник

Сафонова Світлана Олександрівна

Рисунок В.1 – Слайд 1

Актуальність роботи

- Задача класифікації завжди була важливим завданням машинного навчання. А з появою платформ, де візуальний контент став дуже важливим, ці задачі змусили звернути увагу науковців й дослідників.
- Вони можуть вирішувати різноманітні задачі, наприклад, модерування контенту. Завдяки методам класифікації можна блокувати неприйнятний контент на етапі створення посту.



Рисунок В.2 – Слайд 2

Мета роботи

Незважаючи на досягнуті успіхи у питанні класифікації зображень для систем технічного зору ці питання потребують постійних досліджень у зв'язку з зростанням цифрових зображень.

Метою даної роботи є розробка методу класифікації зображень методом машинного навчання SVM.

Рисунок В.3 – Слайд 3

Постановка задачі

Для цього необхідно виконати наступні завдання:

- дослідити математичну модель класифікації;
- дослідити математичну модель методу опорних векторів;
- дослідити інваріантні моменти зображень та правила виділення ознак з зображення;
- підготувати тренувальну базу зображень;
- розробити систему класифікації зображень;
- провести порівняльний аналіз результатів.

Рисунок В.4 – Слайд 4

Задача класифікації

Існує велика кількість об'єктів, кожен з яких належить до одного з декількох класів. Потрібно побудувати функцію, яка зробе висновок до якого класу належить об'єкт. Ця функція називається класифікатор.

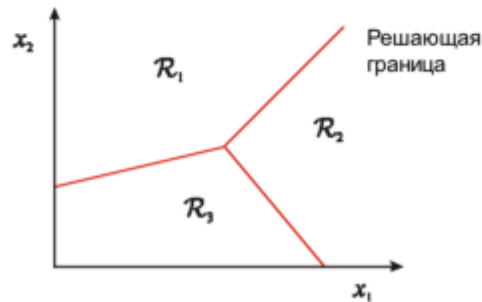


Рисунок В.5 – Слайд 5

Метод опорних векторів

Метод, у якому потрібно знайти гіперплощину, яка буде максимізувати відступ між позитивними й негативними прикладами.

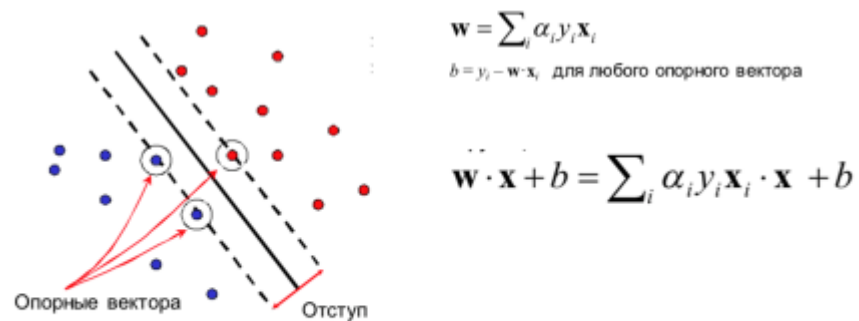


Рисунок В.6 – Слайд 6

Нелінійно роздільні випадки

Відображення вихідного простору параметрів на якийсь багатовимірний простір ознак, де навчальна вибірка лінійно роздільна:

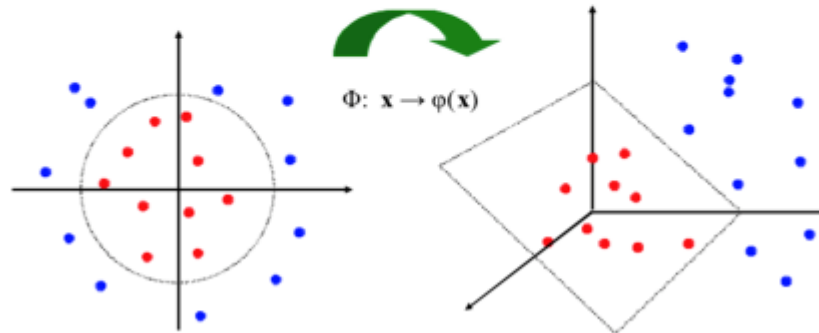


Рисунок В.7 – Слайд 7

- Обчислення скалярних творів в багатовимірному просторі обчислювально складно
- Для спрощення оптимізаційної задачі було розроблено спосіб, що називається kernel trick, в основі якого лежить перехід від скалярних добутків до так званих функцій ядра.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j).$$

- У якості функції ядра можуть використовуватись лише ті функції, які задовольняють теоремі Мерсера.
- Вирішальна функція має вигляд:

$$f(x) = \sum_{i=1}^n \alpha_i y_i \Phi(x) + b = \sum_{i=1}^n \alpha_i y_i K(x_i, x) + b.$$

Рисунок В.8 – Слайд 8

Переваги та недоліки SVM

- це найбільш швидкий метод знаходження вирішальних функцій;
- метод зводиться до вирішення задачі квадратичного програмування в опуклій області, яка завжди має єдине рішення;
- метод знаходить смугу максимальної ширини, що дозволяє в подальшому здійснювати більш впевнену класифікацію;
- метод чутливий до шумів і стандартизації даних;
- не існує загального підходу до автоматичного вибору ядра в разі лінійної нероздільності класів.

Рисунок В.9 – Слайд 9

Результати роботи додатку

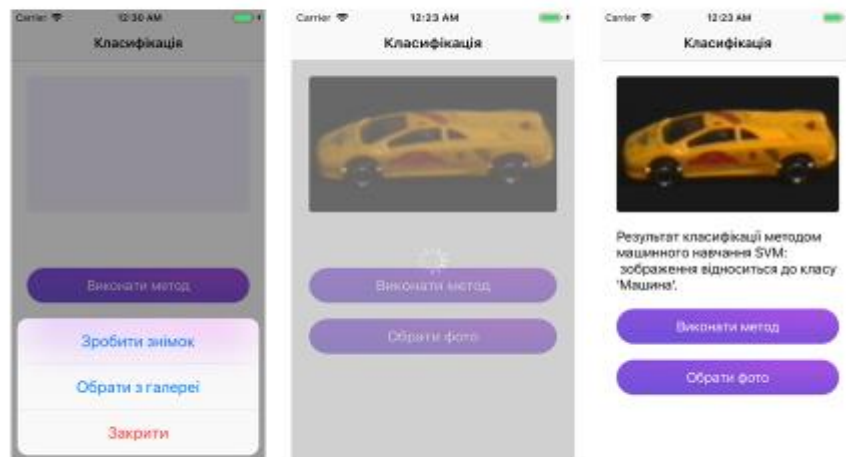


Рисунок В.10 – Слайд 10

Висновки

У результаті виконання магістерської роботи були:

- проведені аналіз сучасних сфер використання класифікації зображень;
- досліджені принципи побудови опорних векторів та функцій ядра;
- вивчені готові рішення і бібліотеки для побудови методу опорних векторів;
- проведені тестування мобільного й серверного додатку.

Рисунок В.11 – Слайд 11

ДЯКУЮ ЗА УВАГУ!

Рисунок В.12 – Слайд 12