

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти магістр

на тему: **«МЕТОДИКА РОЗПІЗНАВАННЯ ВІЗУАЛЬНИХ ЗОБРАЖЕНЬ ЗА
ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ»**

Виконав: студент 6 курсу, групи ПДМ-61
спеціальності 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

Сиротенко І.С.

(прізвище та ініціали)

Керівник

Щербина І.С.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Київ – 2020

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти «Магістр»

Напрямок підготовки 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ О.В.Негоденко

“ _____ ” _____ 2020 року

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Сиротенко Ігор Сергійович

(прізвище, ім'я, по батькові)

1. Тема роботи Методика розпізнавання візуальних зображень за допомогою нейронної мережі

Керівник роботи к.т.н., доцент Щербина І.С.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “13” жовтня 2020 року №230

2. Строк подання студентом роботи 24.12.2020

3. Вихідні дані до роботи:

3.1 Сучасні методи та підходи розпізнавання зображень

3.2 Нейронні мережі, як інструмент розпізнавання зображень

3.3 Розробка моделі для розпізнавання зображень

3.4 Науково-технічна література

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

4.1 Дослідження сучасних підходів розпізнавання зображень

4.2 Аналіз алгоритмів та моделей розпізнавання зображень за допомогою нейронних мереж

4.3 Розробка алгоритму розпізнавання дефектів за допомогою загорткової нейронної мережі

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

5.1 Мета та завдання магістерської роботи;

5.2 Труднощі при розпізнаванні візуальних зображень;

5.3 Нейронна мережа;

5.4 Основні властивості штучних нейронних мереж;

5.6 Навчання нейронних мереж;

5.7 Згорткові нейронні мережі;

5.8 Багатошаровий перцептрон;

5.9 Алгоритм розв'язання задачі розпізнавання дефектів;

5.10 Зображення дефектів;

5.11 Метрики якості обробки зображення;

5.12 Висновки та апробація результатів магістерської роботи;

6. Дата видачі завдання 02.11.2020

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської Роботи	Строк виконання етапів роботи	Примітка
1.	Підбір науково-технічної літератури	02.11.20 - 09.11.20	Виконано
2.	Дослідження сучасних підходів розпізнавання зображень	10.11.20 – 17.11.20	Виконано
3.	Аналіз алгоритмів та моделей розпізнавання зображень за допомогою нейронних мереж	18.11.20 – 25.11.20	Виконано
4.	Розробка прототипу системи розпізнавання зображень дефектів	26.11.20 – 03.12.20	Виконано
5.	Висновки по роботі, оформлення.	04.12.20 – 11.12.20	Виконано
6.	Розробка обов'язкових демонстраційних матеріалів	12.12.20 – 24.12.20	Виконано

Студент

Сиротенко І.С.
(підпис) (прізвище та ініціали)

Керівник роботи

Щербина І.С.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 78 ст., 51 рис., 3 табл., 32 джерела.

НЕЙРОННА МЕРЕЖА, НЕЙРОН, ЗОБРАЖЕННЯ, ФІЛЬТРАЦІЯ, БІНАРИЗАЦІЯ, КЛАСИФІКАЦІЯ, ФІЛЬТР, НАВЧАННЯ, ШАР, АЛГОРИТМ, ВЕКТОР, ВИБІРКА, МАТРИЦЯ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА.

Об'єктом дослідження – є процес обробки графічних зображень за допомогою нейронних мереж..

Мета роботи – покращення процесу розпізнавання візуальних зображень за допомогою розробленого програмного забезпечення на основі нейронної мережі.

Предмет дослідження – методи та засоби обробки графічних зображень.

Методи дослідження. У магістерській роботі використані методи дослідження порівняння, класифікація, математична статистика, математичне імітаційне моделювання, машинне навчання та інтелектуальний аналіз (обробка) даних.

Виконуючи поставлені завдання, магістерської роботи, в першу чергу мною було виконано аналіз сучасних методів розпізнавання зображень та створена класифікація типів існуючих підходів до розпізнавання візуальних зображень.

У другому розділі аналізується ефективність актуальності технології розпізнавання зображень та в якості досліджуваної нейронної мережі, мною обрана згорткова нейронна мережа для подальшої практичної оцінки її ефективності.

У практичній частині роботи проведено експериментальне дослідження різних моделей для розпізнавання дефектів поверхонь металів на графічних зображеннях поверхонь. Розроблено методику експерименту, в ході якого було встановлено оцінки якості роботи алгоритмів.

ЗМІСТ

ВСТУП	8
1 ДОСЛІДЖЕННЯ СУЧАСНИХ ПІДХОДІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ	10
1.1 Огляд проблем розпізнавання візуальних зображень	10
1.2 Архітектура штучних нейронних мереж.....	11
1.3 Огляд існуючих методів розпізнавання графічних зображень.....	15
2 АНАЛІЗ АЛГОРИТМІВ ТА МОДЕЛЕЙ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ	23
2.1 Основні поняття концепції нейронної мережі	23
2.2 Сфера застосувань та види нейронних мереж.....	24
2.3 Штучні нейронні мережі, як інструмент для вирішення задачі	28
2.4 Виявлення локальних ознак. Згорткові нейронні мережі	34
2.5 Багатошаровий перцептрон	40
2.6 Мережа Кохонена.....	44
2.7 Мережа Хеммінга	48
2.8 Нечіткі нейронні мережі. Мережа Ванга-Менделя	50
3. РОЗРОБКА АЛГОРИТМУ РОЗПІЗНАВАННЯ ДЕФЕКТІВ ЗА ДОПОМОГОЮ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ	56
3.1 Постановка завдання розпізнавання	56
3.2 Алгоритм вирішення задачі.....	56
3.3 Побудова моделей допоміжних методів розпізнавання (SVM, Feed Forward MLP)	58
3.4 Фільтри Габора	62
3.5 Архітектура згорткової нейронної мережі	65
3.6 Математична модель згорткової нейронної мережі	69
3.7 Навчання загорткової нейронної мережі	72
3.8 Опис бази зображень	75
3.9 Метрики аналізу моделей і результати окремих моделей	79
ВИСНОВКИ	84

ВСТУП

Актуальність теми дослідження визначається впровадженням комп'ютерних технологій і систем штучного інтелекту в різні сфери діяльності людини. Ці дослідження спираються на фундаментальні основи такі, як розпізнавання образів, нейронні мережі, генетичні алгоритми.

З кожним роком зростає зацікавленість вирішення більш складних задач розпізнавання зображень, що обумовлена автоматизацією, необхідністю образних процесів комунікації в інтелектуальних системах. Тому методика розпізнавання візуальних зображень за допомогою нейронних мереж є актуальною. Один з перспективних напрямків вирішення даної проблеми ґрунтується на застосуванні штучних нейронних мереж і нейрокомп'ютерів, як найбільш прогресивних по відношенню проблем задач розпізнавання візуальних зображень. У наш час запропоновано велику кількість архітектур нейромерж для застосування у розпізнаванні об'єктів. Аналіз запропонованих рішень показує, що й досі не існує єдиної моделі, яка б була кращою серед усіх результуючих показників роботи. Перспективу в удосконаленні архітектур вбачають у згорткових нейронних мережах. Переваги згорткових мереж над багатошаровими полягають у використанні спільної ваги у загорткових шарах, що означає, що для кожного пікселя шару використовується один і той же фільтр (банк ваги).

Метою магістерської роботи є покращення процесу розпізнавання візуальних зображень за допомогою розробленого програмного забезпечення на основі нейронної мережі.

Джерела дослідження:

- <http://itcm.comp-sc.if.ua/2017/Sineglazov.pdf>
- <https://evergreens.com.ua/ua/development-services/neural-network.html>
- <https://center2m.ru/ai-recognition>
- <https://habr.com/ru/company/yandex/blog/314508/>
- <http://masters.donntu.org/2006/fvti/merzlenko/library/art7.htm>

Завдання магістерської роботи: дослідження існуючих способів вирішення проблеми розпізнавання зображень; аналіз моделей і алгоритмів розпізнавання зображень за допомогою нейронних мереж; практичне дослідження ефективності використання нейронних мереж.

Об'єктом дослідження є процес обробки графічних зображень за допомогою нейронних мереж.

Предметом дослідження є методи та засоби обробки графічних зображень.

Методи дослідження. У магістерській роботі використані методи дослідження порівняння, класифікація, математична статистика, математичне імітаційне моделювання, машинне навчання та інтелектуальний аналіз (обробка) даних.

Наукова новизна одержаних результатів. Вдосконалено методіку розпізнавання візуальних зображень використовуючи нейронну мережу, що дозволило розробити алгоритм, який вирішує завдання розпізнавання дефектів поверхонь металів на зображеннях.

Практичне значення одержаних результатів. Отримані результати і розроблений алгоритм рекомендується використовувати як на металургійних та металопрокатних підприємствах так і при розміщенні мережевого обладнання в місцях що перебувають під постійним впливом зовнішніх факторів.

Апробація результатів магістерської роботи. Основні положення і результати магістерської роботи доповідались і обговорювались на одній науково-практичній конференції.

Публікації. За матеріалами роботи опубліковано одну статтю у науковому журналі.

1 ДОСЛІДЖЕННЯ СУЧАСНИХ ПІДХОДІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

1.1 Огляд проблем розпізнавання візуальних зображень

За останні кілька років розпізнавання візуальних зображень отримало важливу роль, також було запропоновано багато алгоритмів виявлення осіб на зображеннях, які застосовують різні підходи. Зображення дуже важливо для ідентифікаційних людей. Мета розпізнавання візуальних зображень визначати, чи є особи на зображенні чи ні, і якщо є, виявити їх будь-яким способом. Метою процесу виявлення осіб є локалізація всіх ділянок зображення, на яких є особа, незалежно від умов освітлення. Незважаючи на існування типових функцій в структурі особи, це важке завдання, оскільки риси сильно варіюються в залежності від віку, кольору шкіри і виразу обличчя. Алгоритми розпізнавання візуальних зображень використовують в різних практичних цілях, наприклад, в інтерфейсах взаємодії людина - комп'ютер, системах відеоспостереження і контролю доступу, в системах технічного зору, робототехніці. Виявлення особи представили ще на початку 1970-х рр., де були використані прості методи.

Виявлення осіб - це визначення кількості осіб, присутніх на зображенні, і виявлення їх положення. Загалом, процес виявлення осіб складається з двох кроків, на першому з яких зображення сканується цілком для виявлення області (функцій), які можуть бути ідентифіковані як обличчя; найбільш поширеним ознакою є колір шкіри. Далі слідує локалізація, яка дає більш точну оцінку розмірів і положення осіб. Виявлення осіб виражається в здатності виявляти і локалізувати особи на зображенні або сцені, виявлення осіб є першим етапом в побудові системи ідентифікації і розпізнавання особи людини.

Багато напрямів науки, виробництва і техніки в значній мірі орієнтуються на розвиток систем, в яких інформація носить візуальний характер. Розпізнавання облич є одним з практичних застосувань алгоритмів розпізнавання візуальних зображень, основним завданням якого є автоматичне виділення особи на

зображенні і, при необхідності, ідентифікація людини по обличчю. У контексті будівництва натурального інтерфейсу людина-комп'ютер завдання виділення особи виявляє своє використання в підсистемі відстеження особи і його характерних рис у відео-потоці для визначення виразу обличчя.

Завдання виявлення особи на зображенні є більш ніж простим для людського зору, однак при спробі побудови автоматичної системи виявлення осіб доводиться зіткнутися з такими труднощами:

- Сильно варіюється зовнішній вигляд особи у різних людей;
- Навіть відносно невелика зміна орієнтації особи щодо камери тягне за собою серйозну зміну зображення особи;
- Можлива присутність індивідуальних особливостей (вуса, бороди, окуляри, зморшки і т.д.) істотно ускладнює автоматичне розпізнавання;
- Зміна виразу обличчя може сильно позначитися на тому, як особа виглядає на зображенні;
- Частина особи може бути невидима (закрита іншими предметами) на зображенні;
- Умови зйомки (освітлення, баланс кольору камери, спотворення зображення, що вноситься оптикою системи, якість зображення) в значній мірі впливають на зображення візуального зображення яке отримали[2].

1.2 Архітектура штучних нейронних мереж

На жаль, сьогодні не існує такого загальновизнаного визначення, яке б задовольняло всіх. Причиною є той факт, що проблемою нейронних мереж займаються спеціалісти в різних галузях науки, і взаємному розумінню заважають методологічні та термінологічні бар'єри. Якщо розглядати штучну нейронну мережу як деяке середовище для обробки інформації, тоді її можна задати шляхом визначення елементів даного середовища та правил їх взаємодії. В цьому випадку говорять, що штучна нейронна мережа є структурою, яка складається з великої кількості процесорних елементів, кожен з яких має локальну пам'ять і може

взаємодіяти з іншими процесорними елементами за допомогою комунікаційних каналів з метою передачі даних, що можуть бути інтерпретовані довільним чином. Процесорні елементи незалежно в часі обробляють локальні дані, що поступають до них через вхідні канали. Зміна параметрів алгоритмів такої обробки залежить тільки від характеристик даних. Іншими словами, штучні нейронні мережі – це обчислювальні парадигми, які реалізують спрощені моделі біологічних нейронних мереж (БНМ). Під БНМ будемо розуміти локальні ансамблі нейронів, які об'єднані синаптичними зв'язками. Сукупність таких ансамблів формує мозок із його різноманітними функціональними можливостями. Сьогодні відома велика кількість нейронних структур та їх модифікацій, що орієнтовані на вирішення конкретного типу задач. Найбільш відомі типи таких структур показані на рис.1.1.

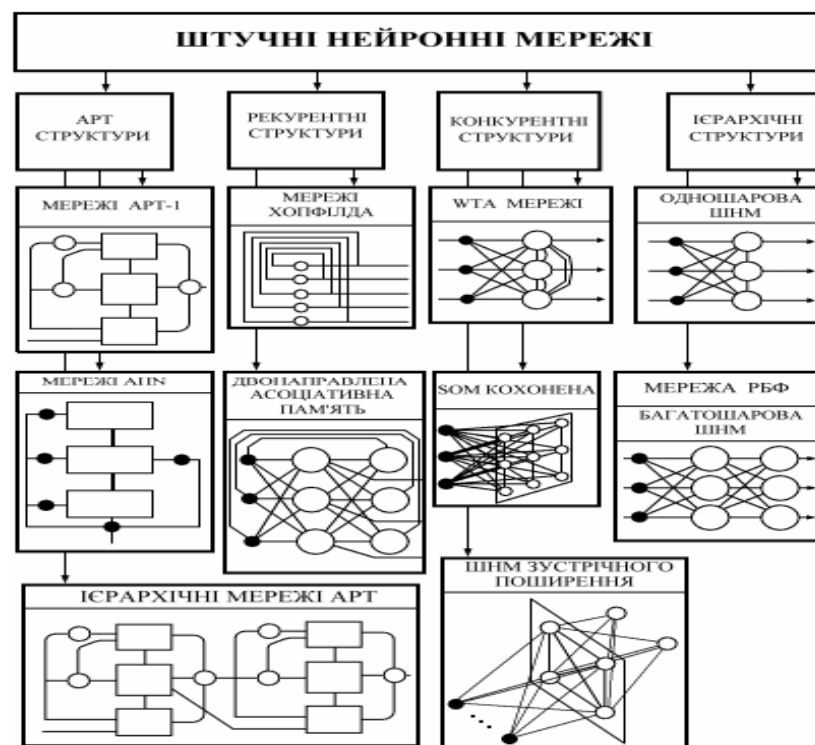


Рис. 1.1 - Нейронні структури

Розглянемо детальніше основні властивості ШНМ:

- локальна обробка інформації в штучному нейроні, який є базовою структурною одиницею мережі;

- паралелізм, результатом якого є вирішення глобальної задачі шляхом представлення її у вигляді множини локальних задач, що тісно взаємодіють між собою;

- здатність до навчання, яке підвищує ефективність роботи мережі;

- здатність до розподіленого зберігання знань, які були одержані в ході навчання.

ШНМ задають у вигляді направлених графів, вершинами яких є нейрони, а ребрами позначені міжнейронні зв'язки. Архітектури сучасних нейронних мереж найчастіше поділяють на три категорії: – мережі з повним набором міжнейронних зв'язків; – мережі з фіксованим індексом оточення; – мережі з пошаровою структурою. У ШНМ із повним набором міжнейронних зв'язків забезпечується можливість взаємодії кожного нейрона мережі з будь-яким іншим. На рис.1.2 наведений приклад повного з'єднання чотирьох нейронів[7].

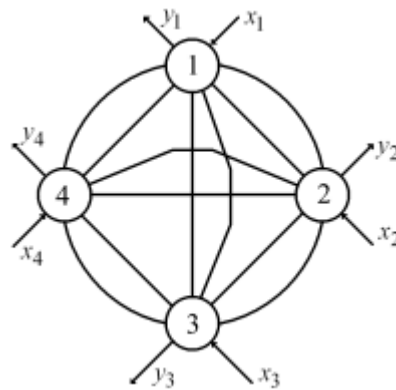


Рис. 1.2 - Повне з'єднання.

Структури з повним з'єднанням є узагальненими структурами, оскільки всі інші довільні об'єднання нейронів можуть розглядатися як підмножини даної структури. Тому ШНМ із повним з'єднанням є універсальним середовищем для реалізації мережних алгоритмів. Широке використання таких структур обмежується недоліком, який полягає в значному зростанні кількості міжнейронних зв'язків при збільшенні кількості нейронів. У випадку, коли необхідно використовувати структури з великою кількістю нейронів, застосовують кліткові структури з фіксованим індексом оточення.

Ще однією важливою характеристикою нейронних мереж із постійним індексом оточення є модульність. Потужність структури у цьому випадку може нарощуватись простим додаванням елементів без зміни ідеології алгоритму, що на ній працює. Найбільш поширеними є структуровані за шарами нейронні мережі, які, в залежності від свого функціонального призначення, можуть містити однотипні або різнотипні нейрони. Виходячи з шарової структури ШНМ, характер міжнейронних зв'язків має свої міжшарові та внутрішньшарові особливості. На рис.1.3 показана класифікація видів міжнейронних зв'язків.

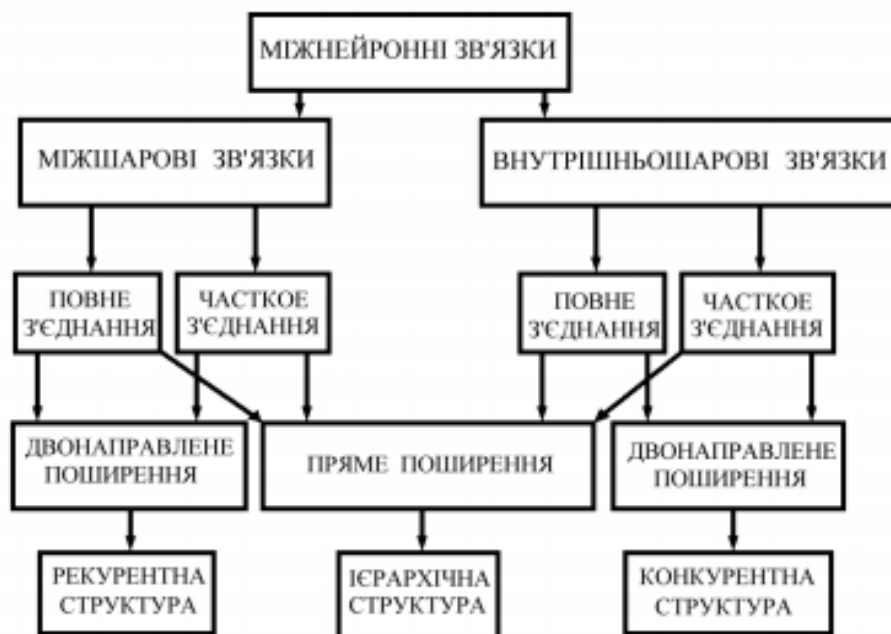


Рис. 1.3 - Види міжнейронних зв'язків.

Нейронні структури з повним з'єднанням можуть бути як одношаровими, так і багатшаровими. У одношаровій структурі з повним з'єднанням всі вхідні сигнали можуть поступати на всі нейрони. Класичною структурою даного типу є персептрон Розенблатта. Багатшарові мережі з повним з'єднанням забезпечують можливість передачі інформації з кожного нейрона попереднього шару на будь-який нейрон наступного. Найчастіше це — багатшарові персептрони. У випадку прямого міжшарового поширення інформація передається однонаправлено у напрямку зростання номера шару. Пряме поширення в межах одного шару

використовують у випадку, коли група нейронів даного шару з'єднана з попереднім шаром опосередковано через виділені нейрони. Двонаправлене поширення допускає також зворотну передачу, що дозволяє створювати алгоритми, за якими враховувався б взаємний міжнейронний обмін. Таким чином, однонаправленість зв'язків призводить до побудови виключно ієрархічних структур, у яких обробка інформації розподіляється по рівнях. За кожний рівень ієрархічної обробки інформації відповідає свій шар нейронів. Вихідна інформація більш високого рівня попереднього шару є вхідною для нейронів наступного шару, який забезпечує глибший рівень обробки.

Двонаправлені міжшарові зв'язки необхідні для реалізації рекурентних структур, які дають можливість застосування ітераційних алгоритмів. Загальною рисою таких структур є те, що подальша передача інформації відбувається тільки у випадку завершення ітераційного процесу. Двонаправлені зв'язки у межах одного шару використовуються для створення конкуруючих груп нейронів. При активації сигналом з попереднього шару кожен з нейронів передає сигнал активації нейронам своєї групи та сигнал гальмування всім іншим нейронам. В результаті конкурентоздатною стає та група нейронів, що одержала найбільше збудження[3].

1.3 Огляд існуючих методів розпізнавання графічних зображень

Методи розпізнавання графічних образів можна розділити на три групи: попередня фільтрація і підготовка зображення, логічна обробка результатів фільтрації, алгоритми прийняття рішень на основі логічної обробки. Межі між групами умовні. Для вирішення завдання далеко не завжди потрібно застосовувати методи з усіх груп, буває достатньо двох, а іноді навіть одного. Фільтрація. До цієї групи відносяться методи, які дозволяють виділити на зображеннях області, без їх аналізу. Велика частина цих методів застосовує єдине перетворення на всі точки зображення. На рівні фільтрації аналіз зображення не

проводиться, але точки, які проходять фільтрацію, можна розглядати як області з особливими характеристиками.

Саме простіше перетворення - це бінаризація зображення по порогу. Для RGB зображення і зображення в градаціях сірого порогом є значення кольору. Зустрічаються ідеальні завдання, в яких такого перетворення досить. Припустимо, потрібно автоматично виділити предмети на білому аркуші паперу: вибір порога, за яким відбувається бінаризація, багато в чому визначає процес самої бінаризації. В даному випадку, зображення було бінаризованим за середнім кольором. Зазвичай бінаризація здійснюється за допомогою алгоритму, який адаптивно вибирає поріг.

Бінаризація може дати дуже цікаві результати при роботі з гістограмами, в тому числі в ситуації, якщо ми розглядаємо зображення не в RGB, а в HSV. Наприклад, сегментувати кольори. На цьому принципі можна побудувати як детектор мітки так і детектор шкіри людини.

Класична фільтрація: Фур'є, ФНЧ, ФВЧ. Класичні методи фільтрації з радіолокації і обробки сигналів можна з успіхом застосовувати в безлічі завдань Pattern Recognition. Традиційним методом в радіолокації, який майже не використовується в зображеннях в чистому вигляді, є перетворення Фур'є (конкретніше - БПФ). Одне з небагатьох винятків, при яких використовується одновимірне перетворення Фур'є, - компресія зображень. Для аналізу зображень одновимірного перетворення зазвичай не вистачає, потрібно використовувати куди більш ресурсоємних двовимірне перетворення[5].

$$G_{uw} = \frac{1}{NM} \sum_{n=1}^{N-1} \sum_{m=1}^{M-1} x_{mne^{-2\pi j}} \quad (1.1)$$

Мало хто його насправді розраховує, зазвичай, куди швидше і простіше використовувати згортку області з уже готовим фільтром, заточеним на високі (ФВЧ) або низькі (ФНЧ) частоти. Такий метод, звичайно, не дозволяє зробити аналіз спектра, але в конкретному завданні відеоспостереження зазвичай потрібен не аналіз, а результат.

Найпростіші приклади фільтрів, що реалізують підкреслення низьких частот (фільтр Калмана (рис. 1.4)) і високих частот (Фільтр Габора (рис. 1.5)).

Для кожної точки зображення вибирається вікно і перемножується з фільтром того ж розміру. Результатом такої згортки є нове значення точки. При реалізації ФНЧ і ФВЧ виходять зображення такого типу:

Але що якщо використовувати для згортки з сигналом якусь довільну характеристическую функцію? Тоді це буде називатися "Вейвлет- перетворення". Це визначення вейвлетов не є коректним, але традиційно склалося, що в багатьох командах вейвлет-аналізом називається пошук довільного патерну на зображенні за допомогою згортки з моделлю цього патерну. Існує набір класичних функцій, використовуваних в вейвлет-аналізі. До них відносяться вейвлет Хаара, вейвлет Морлі, вейвлет мексиканський капелюх, і.т.д.

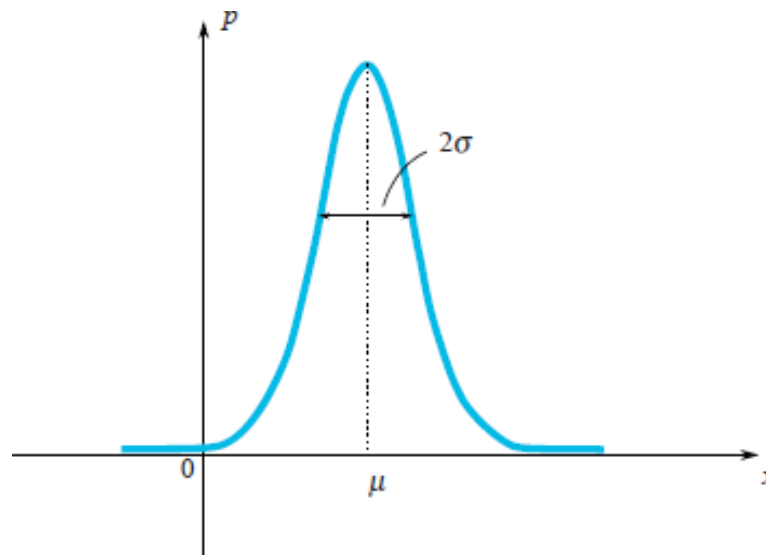


Рис. 1.4 – Фільтр Калмана

Вище наведено 4 приклади класичних вейвлетов. 3x-мірний вейвлет Хаара, 2x-мірні вейвлет Мейсера, вейвлет Мексиканська Капелюх, вейвлет Добеши. Хорошим прикладом використання розширеної трактування вейвлетов є завдання пошуку відблиску в оці, для якої вейвлетом є сам відблеск. Класичні вейвлети зазвичай використовуються для стиснення зображень, або для їх класифікації (буде описано нижче).

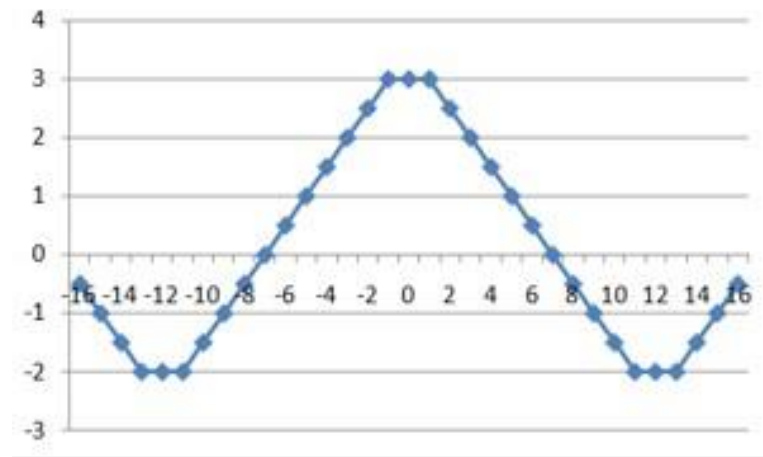


Рис. 1.5 – Фільтр Гарбора

Після такої вільного трактування вейвлетів з мого боку варто згадати власне кореляцію, що лежить в їх основі. При фільтрації зображень це незамінний інструмент. Класичне застосування - кореляція відеопотоку для знаходження зрушень або оптичних потоків. Найпростіший детектор зсуву - теж в якомусь сенсі різницевий корелятор. Там де зображення не корелюють - був рух.

Цікавим класом фільтрів є фільтрація функцій. Це чисто математичні фільтри, які дозволяють виявити просту математичну функцію на зображенні (пряму, параболу, коло). Будується зображення, в якому для кожної точки вихідного зображення промальовується безліч функцій, які її породжують. Найбільш класичним перетворенням є перетворення Хафа для прямих (рис. 1.6). У цьому перетворенні для кожної точки $(x; y)$ виділяється безліч точок $(a; b)$ прямих $y = ax + b$, для яких вірно рівність.

Перетворення Хафа дозволяє знаходити будь-які параметризовані функції. Наприклад окружності. Є модифіковане перетворення, яке дозволяє шукати будь-які фігури. Це перетворення страшенно люблять математики. Але ось при обробці зображень, воно, на жаль, працює далеко не завжди. Дуже повільна швидкість роботи, дуже висока чутливість до якості бінаризації. Навіть в ідеальних ситуаціях я вважав за краще обходитися іншими методами[8].

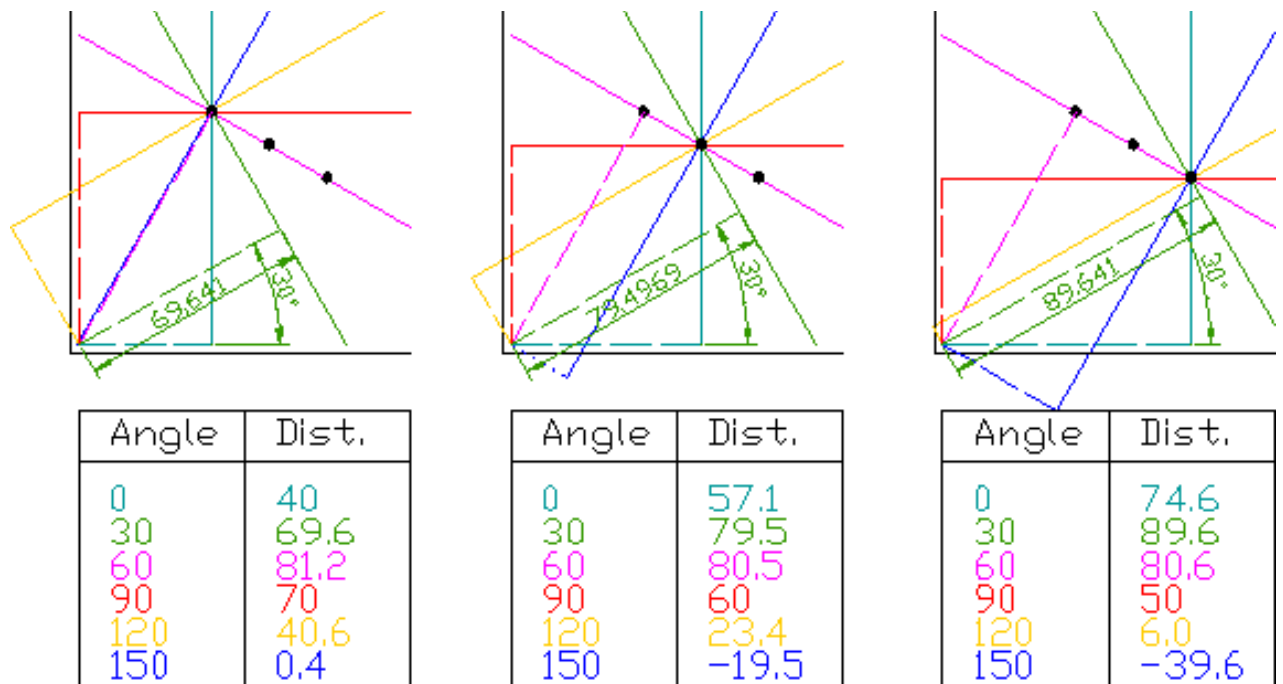


Рис. 1.6 – Перетворення Хафа для прямих

Аналогом перетворення Хафа для прямих є перетворення Радону. Воно обчислюється через БПФ, що дає вигреш продуктивності в ситуації, коли точок дуже багато. До того ж його можна використовувати до не бінаризованими зображенню.

Окремий клас фільтрів - фільтрація кордонів і контурів. Контури дуже корисні, коли ми хочемо перейти від роботи з зображенням до роботи з об'єктами на цьому зображенні. Коли об'єкт досить складний, але добре виділяється, то часто єдиним способом роботи з ним є виділення його контурів. Існує цілий ряд алгоритмів, які вирішують задачу фільтрації контурів:

Найчастіше використовується саме Кенні, який добре працює і реалізація якого є в OpenCV (Собель там теж є, але він гірше шукає контури).

Логічна обробка результатів фільтрації. Фільтрація дає набір придатних для обробки даних. Але найчастіше можна просто взяти і використовувати ці дані без їх обробки. У цьому розділі буде кілька класичних методів, що дозволяють перейти від зображення до властивостей об'єктів, або до самих об'єктів.

Переходом від фільтрації до логіки, є методи математичної морфології. По суті, це найпростіші операції нарощування і ерозії бінарних зображень. Ці методи

дозволяють прибрати шуми з бінарного зображення, збільшивши або зменшивши наявні елементи. На базі математичної морфології існують алгоритми оконтурювання, але зазвичай користуються якимись гібридними алгоритмами або алгоритмами в зв'язці.

У підозділі про фільтрації вже згадувалися алгоритми отримання кордонів. Отримані кордону досить просто перетворюються в контури. Для алгоритму Кенні це відбувається автоматично, для інших алгоритмів потрібна додаткова бінаризація.

Контур є унікальною характеристикою об'єкта. Часто це дозволяє ідентифікувати об'єкт по контуру. Існує потужний математичний апарат, що дозволяє це зробити. Апарат називається контурним аналізом.

Особливі точки це унікальні характеристики об'єкта, які дозволяють зіставляти об'єкт сам з собою або зі схожими класами об'єктів. Існує кілька десятків способів дозволяють виділити такі точки. Деякі способи виділяють особливі точки в сусідніх кадрах, деякі через великий проміжок часу і при зміні освітлення, деякі дозволяють знайти особливі точки, які залишаються такими навіть при поворотах об'єкта. Почнемо з методів, що дозволяють знайти особливі точки, що не такі стабільні, зате швидко розраховуються, а потім підемо по зростанню складності:

Перший клас. Особливі точки, які є стабільними протягом секунд. Такі точки служать для того, щоб вести об'єкт між сусідніми кадрами відео, або для відомості зображення з сусідніх камер. До таких точок можна віднести локальні максимуми зображення, кути на зображенні (рис. 1.7), точки в яких досягається максимуми дисперсії, певні градієнти і.т.д.

Другий клас. Особливі точки, які є стабільними при зміні освітлення і невеликих рухах об'єкта. Такі точки служать в першу чергу для навчання і подальшої класифікації типів об'єктів.

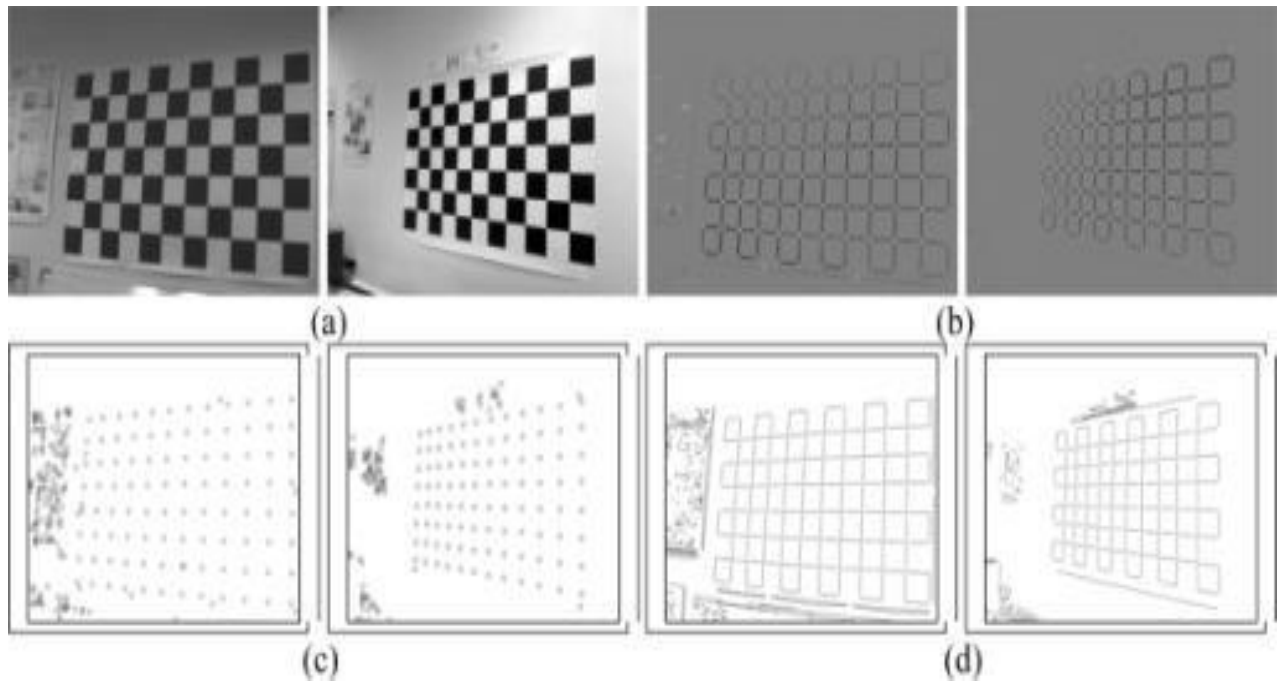


Рис 1.7 – Розпізнавання кутів на зображеннях

Наприклад, класифікатор пішохода або класифікатор особи - це продукт системи, побудованої саме на таких точках. Деякі з раніше згаданих вейвлетов можуть є базою для таких точок. Наприклад, примітиви Хаара, пошук відблисків, пошук інших специфічних функцій[13]. До таких точок відносяться точки, знайдені методом гістограм спрямованих градієнтів (рис 1.8).

Третій клас. Стабільні точки. Мені відомо лише про два методи, які дають повну стабільність і про їх модифікації. Це SURF і SIFT. Вони дозволяють знаходити особливі точки навіть при повороті зображення. Розрахунок таких точок здійснюється довше в порівнянні з іншими методами, але досить обмежений час. Нажаль ці методи запатентовані.

Висновки за розділом. Таким чином можна зробити висновок, що існує багато методів та способів роботи із зображенням. Розпізнавання візуальних зображень та їх обробка це важкий і кропіткий процес, але перспектива таких технологій дуже висока.

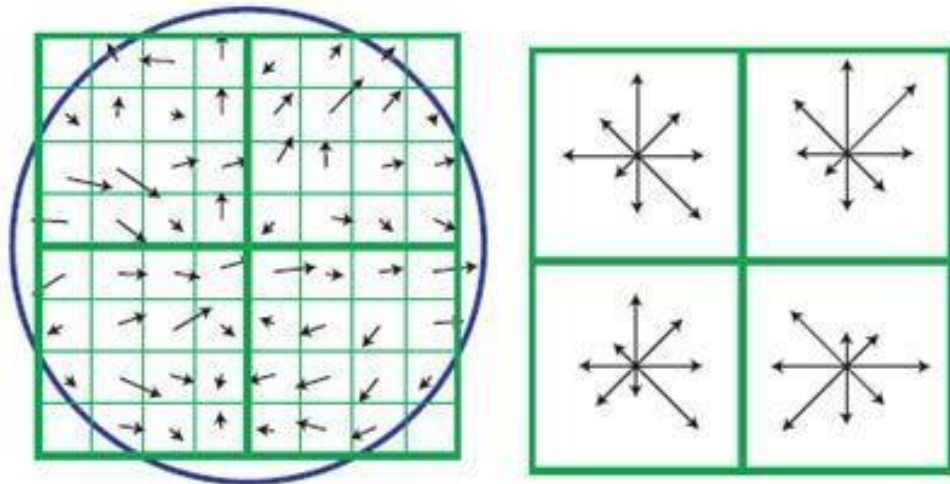


Рис 1.8 – Гістограм спрямованих градієнтів

Інструменти розпізнавання зображень мають в собі реалізації складних алгоритмів, що забезпечують швидку обробку пікселів зображень і виведення результатів. За допомогою різних інструментів є можливість реалізовувати методи розпізнавання візуальних зображень. Нами було розглянуто популярність та структуру нейронних мереж і їх можливості, а ще кожен крок обробки вхідних даних. Тому можна сказати, що розпізнавання візуальних зображень як напрямок комп'ютерних технологій розвивається дуже швидко і відкриває все нові можливості, які широко застосовуються в нашому повсякденному житті. Та особливо цей напрямок корисний та ефективний для бізнесу.

2 АНАЛІЗ АЛГОРИТМІВ ТА МОДЕЛЕЙ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

2.1 Основні поняття концепції нейронної мережі

Штучна нейронна мережа побудована за принципом біологічної нейронної мережі. Це математична інтерпретація нейронів головного мозку людини, яка виникла внаслідок вивчення і спроби змоделювати процеси, які протікають в мозку. Для того щоб домогтися високої продуктивності, нейронні мережі використовують множини взаємозв'язків між нейронами. На рис.2.1 представлена базова модель структури нейронної мережі.

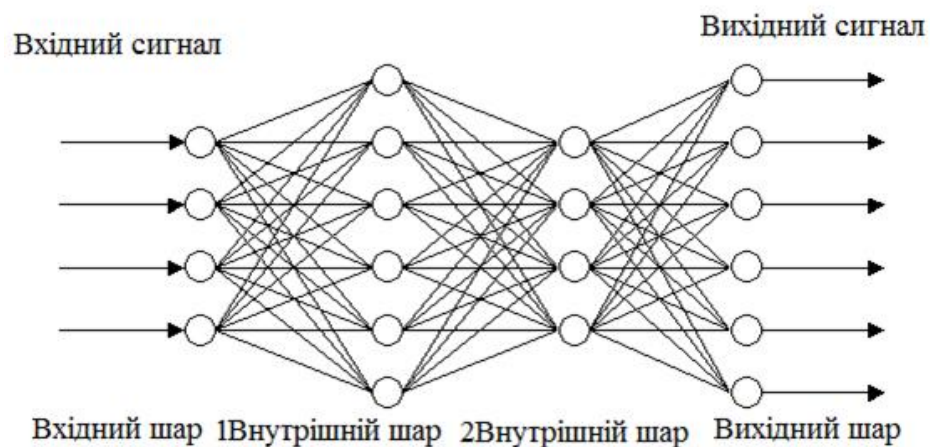


Рис 2.1 - Базова структура нейронної мережі

Штучний нейрон в нейромережі - це проста функція, яка має свій вхід і вихід. Ця функція полягає в тому, що кожен вхід множиться на ваги, після цього все підсумовується, проходить через функцію активації і на вихід видається результат, як показано на рис. 2.2.

Між вхідним і вихідним шарами є певна кількість прихованих шарів, які виконують основні обчислювальні операції.

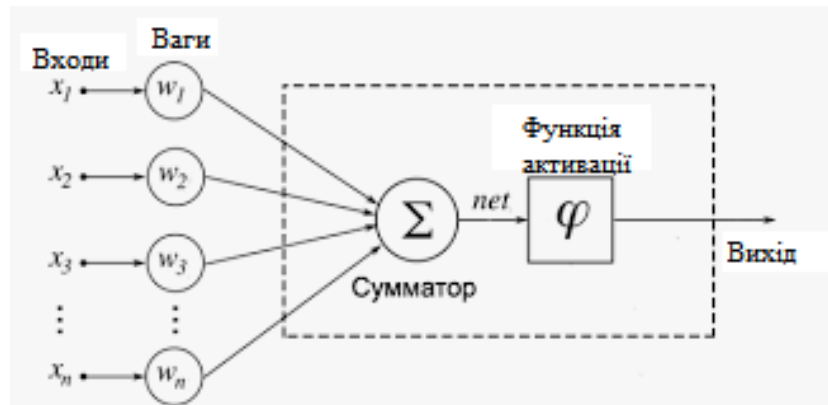


Рис. 2.2 - Структура штучного нейрона

Якщо число прихованих шарів дорівнює 1, то мережа вважається неглибокою, якщо більше – глибокою, рис.2.3.

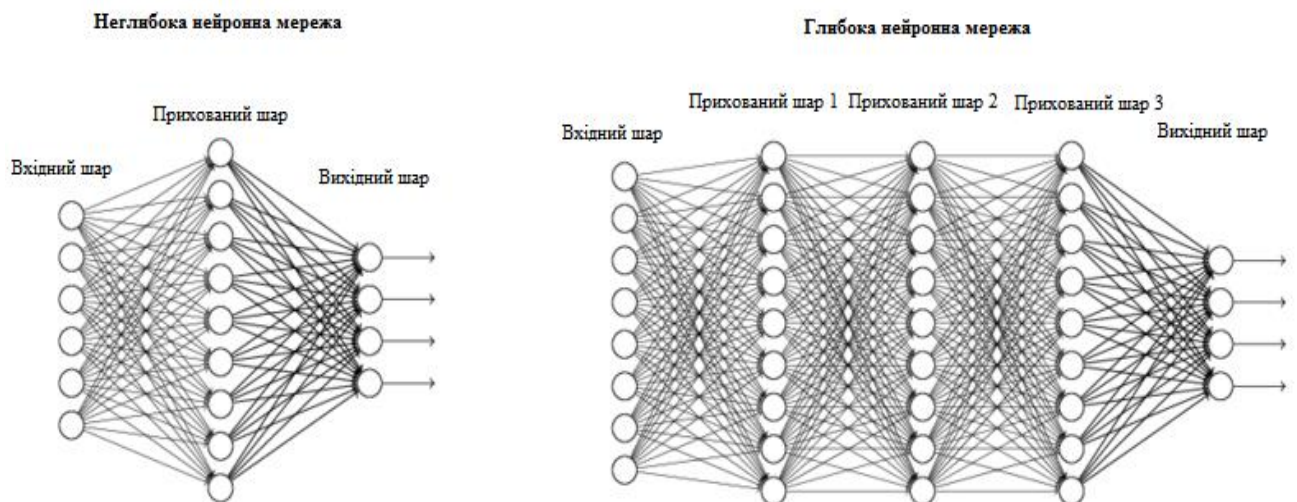


Рис. 2.3 – Приклад багатошарових нейронних мереж

2.2 Сфера застосувань та види нейронних мереж

Сьогодні компанії будь-якого профілю стикаються з необхідністю обробки великої кількості даних. Від якості і швидкості їх аналізу залежить ефективність прийняття рішень і рівень підтримки клієнтів. Впоратися зі зростанням інформаційними потоками допомагають технології штучного інтелекту. В їх основі лежать алгоритми глибокого машинного навчання, що поєднують в собі

нейромережі певних типів. В цьому розділі ми розглянемо, яке саме застосування вони знайшли в різних галузях.

Може скластися враження, що штучні нейронні мережі (ANN) - це якась проривна технологія, однак на рівні алгоритмів вони були описані ще в другій половині минулого століття. Просто зараз з'явилася технічна можливість реалізувати їх в кінцевих продуктах – людство за весь період розвитку накопичило достатні обсяги інформації і створило засоби для її швидкого аналізу. До теперішнього часу найбільшого поширення набули такі види нейромереж:

- згорткові нейронні мережі (CNN) імітують роботу зорової кори головного мозку і частково виконують функцію абстрактного мислення. Вони прекрасно справляються із завданням розпізнавання зображень, а їх обчислення легко розпаралелити на графічних процесорах, що дозволяє створювати відносно дешеві апаратні платформи з елементами штучного інтелекту (ШІ).

CNN застосовуються в системах машинного зору безпілотних автомобілів, комерційних дронів, роботів, а також в охоронному відеоспостереженні. Ви теж кожен день використовуєте CNN, якщо включили в налаштуваннях смартфона розблокування за допомогою розпізнавання особи (Face ID)

- рекурентні нейронні мережі (RNN) мають короткочасною пам'яттю, за рахунок чого легко аналізують послідовності довільної довжини. RNN розбивають потік даних на елементарні частини і оцінюють взаємозв'язки між ними. Ці алгоритми знайшли основне застосування в розпізнаванні рукописного тексту й мови. Коли ви шукаєте мелодію на слух в Shazam, розмовляєте з Siri, Google Now або Алісою від «Яндекса», залишаєте замітки від руки для Cortana - на хмарних платформах беруться за справу рекурентні нейромережі.

- мережі з довготривалою і короткочасною пам'яттю (LSTM) стали подальшим розвитком RNN. Вони гарні для прогнозування змін будь-якої величини (наприклад, біржових курсів або купівельного попиту) шляхом екстраполяції. Також їх застосовують для глибокого аналізу природної мови. Наприклад, Google використовує LSTM в персональному помічника і системі машинного перекладу Google Translate. Без LSTM якість перекладів так і

залишалося б на рівні програм з дев'яностих, з якими часом було простіше перевести текст самому, ніж виправляти численні ляпи.

- керовані рекурентні блоки (GRU) - порівняно недавня модифікація RNN, що з'явилася тільки в 2014 році. Їх використовують для синтезу мови, яка володіє емоційним забарвленням і звучить як справжня. Наприклад, в тестах сервісів Google Duplex і Microsoft Xiaoice люди не змогли відрізнити як говорять боти від живих співрозмовників. Примітно, що Xiaoice дозволила Microsoft зміцнитися на азіатському ринку, де розвиток компанії завжди стримувалося мовним бар'єром.

- глибокі нейронні мережі (DNN) - будь-яка мережа більш ніж з трьома шарами. Вони лежать в основі механізмів глибокого машинного навчання, знаходячи неявні взаємозв'язку між різнорідними даними. Яскравий приклад - пошук кореляцій між розвитком захворювань і різними потенційними факторами у величезних масивах наукових статей за допомогою IBM Watson[17].

- генеративно-змагальні мережі (GAN). Це комбінація нейромереж, одна з яких генерує варіанти, а інша відсіває їх (виступає в якості арбітра). Таке поєднання дозволяє реалізувати машинне навчання без учителя, що підвищує автономність її. Наприклад, PixelDTGAN генерує окремі зображення одягу, взуття та аксесуарів для каталогів онлайн-магазинів. В якості вхідних даних використовуються фотографії, на яких ці предмети гардероба демонструють фотомоделі. Поки зйомка для каталогів одягу вважається досить витратною частиною електронної комерції, але цілком можливо, що в найближчі роки нейромережі дозволять швидко проілюструвати каталог, навіть не залучаючи фотографа. Обробка фотографій теж забирає багато часу. Ви можете спробувати зробити це набагато швидше, додаючи і прибираючи графічні об'єкти за допомогою іншої нейромережі - IBM GANPaint. Подібна їй нейросеть DRAGAN вже застосовується для автоматичної відтворення персонажів аніме і мультфільмів. Вона дозволяє прискорити вихід нових серій і утримати аудиторію розважальних каналів, не перевантажуючи аніматорів колосальним об'ємом роботи. GAN дозволяють анімувати тривимірну модель людини, переносючи на неї руху іншого в реальному часі. Перші результати виглядають не дуже

переконливо, однак особливість будь нейромережі - в тому, що вона покращується з кожною новою порцією даних.

Google використовує нейронні мережі в усіх своїх сервісах - від онлайн-пошуку і фільтрації пошти до підбору роликів на YouTube і перекладу текстів. На відміну від старих моделей машинного перекладу, нейромережі Google оперують практично необмеженими об'ємами даних для розпізнавання мовних шаблонів. Вони можуть виявляти більше семантичних зв'язків і покращувати результат з кожним разом за допомогою постійного навчання.

Ще з 2001 року в Google став можливий пошук по зображеннях, проте на перших порах для цього застосовувався простий аналіз метаданих. Тобто фактично пошук відбувався по текстовому опису картинки, яке часто було неточним. Сучасна версія цього сервісу використовує систему автоматичного опису графічних файлів. Вона побудована на комбінації рекурентних і згортальних нейромереж, тому в ній можна шукати подібні зображення, сортувати їх по типам і навіть знаходити графічні файли без текстової анотації. Google Cloud Speech-to-Text став справжньою магією для великих компаній, що працюють з клієнтами з різних країн. Вихід за межі рідного регіону завжди висував високі вимоги до організації технічної підтримки. Якщо раніше для цього доводилося відкривати регіональні філії або наймати перекладачів, то тепер в рішення для колл-центрів може бути інтегрований цей сервіс. Він підтримує інтерфейс прикладного програмування і допомагає перетворювати запис розмов в текст, застосовуючи моделі нейронних мереж. В даний час він розпізнає понад 120 мов і продовжує освоювати нові.

LinkedIn використовує нейронні мережі спільно з лінійними текстовими класифікаторами для аналізу всіх записів, які розповсюджуються через свою платформу. Як пояснює віце-президент напрямку II LinkedIn Діпак Агарвал, таке поєднання допомагає ефективніше виявляти спам і контент образливого змісту, а також підбирати кращі персональні рекомендації.

Facebook застосовує подібний мережевий алгоритм аналізу тексту. Він називається DeepText і переважно використовується для таргетування рекламних

оголошень. Додатково Facebook використовує згорткові нейронні мережі для розпізнавання осіб і генеративно-змагальні мережі для обробки зображень. Наприклад, якщо на знімку людина моргнула, то GAN від Facebook знайде його зображення з відкритими очима і відретушує невдале фото по аналогії. Чим більше портретів користувача доступно для аналізу, тим точніше вийде результат. Tesla використовує набір DNN у фірмовій системі автопілота. Вони розпізнають всі об'єкти навколо машини в реальному часі, класифікують їх і визначають характеристики. Дев'ята версія автопілота стала в чотири рази складніше восьмої за рахунок збільшення потоку даних. Одних камер стало вісім штук, причому їх дозвіл теж збільшилася. Робота Tesla Autopilot v.9 представлено на рис.2.4



Рис. 2.4 - Приклад роботи Tesla Autopilot v.9.

Антивірусні розробники застосовують різні нейронні мережі для пошуку нових загроз інформаційній безпеці. Вони допомагають розпізнавати невідомі модифікації шкідливих файлів і нові види мережових атак.

2.3 Штучні нейронні мережі, як інструмент для вирішення задачі

Дослідження штучних нейронних мереж пов'язано з тим, що вони дозволяють наблизитися до можливостей обробки інформації людським мозком, який являє собою надзвичайно складний, нелінійний, паралельний комп'ютер

(систему обробки інформації). Мозок має здатність організувати свої структурні компоненти, так звані нейрони, так, щоб вони могли виконувати конкретні задачі (такі як розпізнавання образів, обробку сигналів органів почуттів, моторні функції) в багато разів швидше, ніж можуть дозволити найшвидкодіючі сучасні комп'ютери.

На сьогоднішній день існує багато прикладів використання штучних нейронних мереж для прогнозів, класифікації, оптимізації, розпізнавання образів та багато інших.

Нейронні мережі – обчислювальні структури, які моделюють прості біологічні процеси, що асоціюються з процесами людського мозку. Вони представляють собою системи, здатні до навчання шляхом аналізу позитивних і негативних впливів. Елементарним перетворювачем в даних мережах є штучний нейрон або просто нейрон, названий так за аналогією з біологічним прототипом.

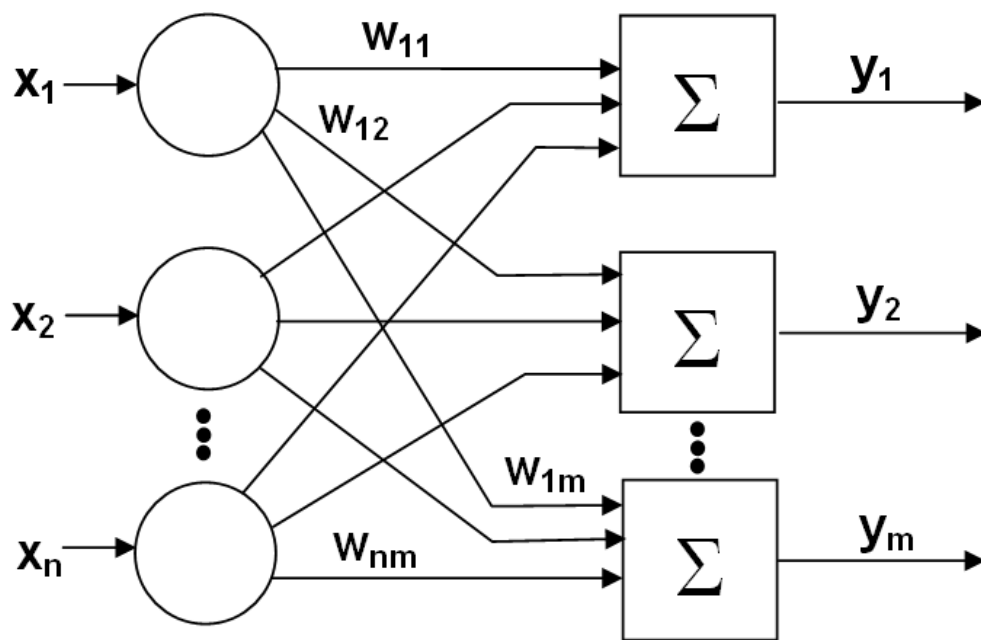


Рис 2.5 – Одношарова нейронна мережа

Прототипом для створення нейрона став біологічний нейрон головного мозку. Біологічний нейрон має тіло, сукупність відростків – дендритів, за якими в нейрон надходять вхідні сигнали, і аксонів, що передають вихідні сигнали нейронів іншим клітинам. Точка з'єднання дендрита і аксона називається

синапсом. Спрощено функціонування нейрона можна представити наступним чином[11]:

Нейрон отримує від дендритів набір (вектор) вхідних сигналів.

У тілі нейрона оцінюється сумарне значення вхідних сигналів. Однак входи нейрона нерівнозначні. Кожен вхід характеризується деяким ваговим коефіцієнтом, що визначає важливість інформації переданої ним. Таким чином, нейрон не просто підсумовує значення вхідних сигналів, а обчислює скалярний добуток вектора вхідних сигналів і вектора вагових коефіцієнтів.

Нейрон формує вихідний сигнал, інтенсивність якого залежить від значення обчисленого скалярного перемноження. Якщо воно не перевищує деякого заданого порогу, то вихідний сигнал не формується зовсім – нейрон «не спрацьовує».

Вихідний сигнал надходить на аксон і передається дендриту інших нейронів.

Найважливішою властивістю нейронних мереж є їх здатність навчатися на основі даних навколишнього середовища і в результаті навчання підвищувати свою продуктивність. Підвищення продуктивності відбувається з часом у відповідності з певними правилами. Навчання нейронної мережі відбувається за допомогою інтерактивного процесу корегування синаптичних ваг і порогів. В ідеальному випадку нейронна мережа отримує знання про навколишнє середовище на кожній ітерації процесу навчання.

З поняттям навчання асоціюється досить багато видів діяльності, тому складно дати цьому процесу однозначне визначення. Більше того, процес навчання залежить від точки зору на нього. Саме це робить практично неможливим появу будь-якого точного визначення цього поняття. Наприклад, процес навчання з точки зору психолога в корені відрізняється від навчання з точки зору шкільного вчителя. З позицій нейронної мережі, ймовірно, можна використовувати наступне визначення: Навчання – це процес, у якому вільні параметри нейронної мережі настроюються за допомогою моделювання середовища, у яке ця мережа вбудована.

Тип навчання визначається способом підстроювання цих параметрів. Це визначення процесу навчання нейронної мережі передбачає наступну послідовність подій:

- У нейронну мережу надходять стимули із зовнішнього середовища.
- У результаті реалізації першого пункту змінюються вільні параметри нейронної мережі.
- Після зміни внутрішньої структури нейронна мережа відповідає на штрафи вже іншим чином.

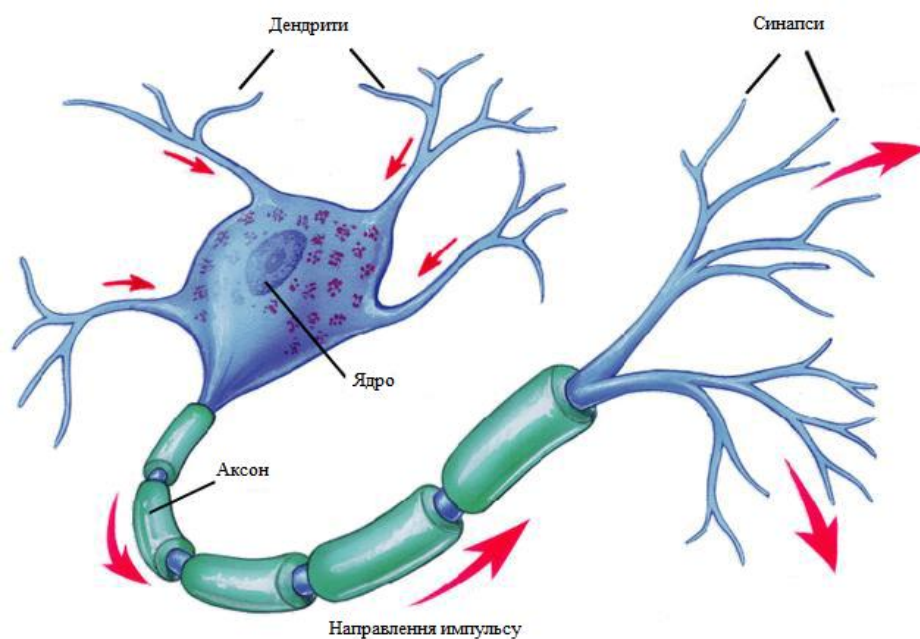


Рис 2.6 – Біологічний нейрон

Вищевказаний список чітких правил вирішення проблеми навчання нейронної мережі називається алгоритмом навчання. Не існує універсального алгоритму навчання, відповідного для всіх архітектур нейронних мереж. Існує лише набір засобів, представлений безліччю алгоритмів навчання, кожен з яких має свої переваги. Алгоритми навчання відрізняються один від одного способом налаштування синаптичних ваг нейронів. Ще однією відмінною характеристикою є спосіб зв'язку навченою нейронної мережі із зовнішнім світом. У цьому контексті говорять про парадигму навчання, пов'язану з моделлю навколишнього

середовища, в якій функціонує дана нейронна мережа. Існують три способу навчання: з вчителем; без вчителя; змішана[6].

Навчання нейронної мережі з учителем припускає, що для кожного вхідного вектора з навчальної множини існує необхідне значення вихідного вектора, так званого цільовим. Ці вектори утворюють навчальну пару. Ваги мережі змінюють до тих пір, поки для кожного вхідного вектора не буде отриманий прийнятний рівень відхилення вихідного вектора від цільового. Нейронна мережа має у своєму розпорядженні правильними відповідями (виходами мережі) на кожен вхідний приклад. Ваги налаштовуються так, щоб мережа виробляла відповіді, як можна більш близькі до відомих правильних відповідей. Посилений варіант навчання з учителем припускає, що відома тільки критична оцінка правильності виходу нейронної мережі, але не самі правильні значення виходу.

Навчання нейронної мережі без вчителя є набагато більш правдоподібною моделлю навчання з точки зору біологічних коренів штучних нейронних мереж. Навчальна множина складається лише з вхідних векторів. Алгоритм навчання нейронної мережі підлаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред'явлення досить близьких вхідних векторів давало однакові виходи. Навчання без вчителя не вимагає знання правильних відповідей на кожний приклад навчальної вибірки. У цьому випадку розкривається внутрішня структура даних, або кореляції між зразками в системі даних, що дозволяє розподілити зразки за категоріями. При змішаному навчанні частина ваг визначається за допомогою навчання з учителем, у той час як інша виходить за допомогою самонавчання.

Класифікація образів. Завдання полягає у вказівці приналежності вхідного образу (наприклад, мовного сигналу чи рукописного символу), представленого вектором ознак, одному або декільком попередньо визначеним класам. Наприклад розпізнавання букв, розпізнавання мови, класифікація сигналу електрокардіограми, класифікація клітин крові.

Кластеризація (категоризація). При вирішенні задачі кластеризації, яка відома також як класифікація образів «без вчителя», відсутня навчальна вибірка з

мітками класів. Алгоритм кластеризації заснований на подібності образів і розміщує близькі образи в один кластер. Відомі випадки застосування кластеризації для стиснення даних і дослідження властивостей даних.

Апроксимація функцій. Припустимо, що є навчальна вибірка (пари даних вхід-вихід), яка генерується невідомою функцією $F(x)$, спотвореної шумом. Завдання апроксимації полягає в знаходженні оцінки невідомої функції $F(x)$. Апроксимація функцій необхідна при рішенні численних інженерних і наукових задач моделювання.

Прогнозування. Завдання полягає в передбаченні деякого значення у деякий майбутній момент часу. Передбачення мають значний вплив на прийняття рішень у бізнесі, науці і техніці. Передбачення цін на фондовій біржі і прогноз погоди є типовими програмами прогнозування.

Оптимізація. Численні проблеми в математиці, статистиці, техніці, науці, медицині та економіці можуть розглядатися як проблеми оптимізації. Завданням алгоритму оптимізації є знаходження такого рішення, яке задовольнить обмеженням системи і максимізує чи мінімізує цільову функцію. Задача комівояжера є класичним прикладом задачі оптимізації.

Пам'ять, що адресується за змістом. У моделі обчислень фон Неймана звертання до пам'яті доступно тільки за допомогою адреси, який не залежить від утримання пам'яті. Більш того, якщо допущена помилка в обчисленні адреси, то може бути знайдена зовсім інша інформація. Асоціативна пам'ять, чи пам'ять, що адресується за змістом, доступна за вказівкою заданого змісту. Зміст пам'яті може бути викликаний навіть по частковому входу. Асоціативна пам'ять надзвичайно ефективна при створенні мультимедійних інформаційних баз даних.

Програмні продукти на базі нейронних мереж використовують для контролю якості води та можуть знаходити пластикові бомби в багажі авіапасажирів. Фахівці інвестиційного банку за допомогою програмного нейропакета роблять короткострокові прогнози коливань курсів валют.

Адаптація до змін навколишнього середовища. Нейронні мережі мають здатність адаптуватися до змін навколишнього середовища. Зокрема, нейронні

мережі, навчені діяти в певному середовищі, можуть бути легко перевчені для роботи в умовах незначних коливань параметрів середовища. Більш того, для роботи в нестационарній середовищі (де статистика змінюється з плином часу, наприклад як котирування акцій на біржі) можуть бути створені нейронні мережі, вони перенавчаються в реальному часі. Чим вище адаптивні здатності системи, тим більш стійкою буде її робота в нестационарній середовищі. При цьому слід зауважити, що адаптивність не завжди веде до стійкості; іноді вона призводить до зовсім протилежного результату. Наприклад, адаптивна система з параметрами, що швидко змінюються в часі, може також швидко реагувати і на сторонні порушення, що викличе втрату продуктивності. Для того, щоб використовувати всі переваги адаптивності, основні параметри системи повинні бути досить стабільними, щоб можна було не враховувати зовнішні перешкоди, і досить гнучкими, щоб забезпечити реакцію на істотні зміни середовища.

Висока швидкість проведення обчислювань. Нейронні мережі мають потенційну надвисоку швидкодію за рахунок використання масового паралелізму обробки інформації.

Нейронні мережі потенційно відмовостійкі при апаратній реалізації. Це означає, що за несприятливих умов їх продуктивність падає незначно. Наприклад, якщо пошкоджений якийсь нейрон або його зв'язки, вилучення запам'ятовані інформації ускладнюється. Проте, враховуючи розподілений характер зберігання інформації в нейронній мережі, можна стверджувати, що тільки серйозні пошкодження структури нейронної мережі суттєво вплинуть на її працездатність. Тому нейронні мережі чудово підходять для розпізнавання образів на зображеннях.

2.4 Виявлення локальних ознак. Згорткові нейронні мережі

Проблеми, що виникли в процесі використання моделей, які формують цілісні репрезентації, сприяли розвитку нової групи алгоритмів, що використовують локальні ознаки зображень. Необхідність такого походу була

продиктована властивістю стаціонарності природних зображень - об'єкти, присутні на зображенні, могли вільно переміщатися в межах поля зору, при цьому бажаним результатом розпізнає алгоритму залишалось співвіднесення безлічі таких інваріантних репрезентацій об'єкта до одного класу.

Крім іншого, використання локальних ознак при розпізнаванні зображень було підкріплено свідченнями з області нейробіології. У класичній праці Д. Хьюбела і Т. Візела, що візуальна кора головного мозку являє собою складний комплекс клітин, кожна з яких чутлива тільки до обмеженого ділянці поля зору. Такі ділянки, інакше звані рецептивних полями, стикуються разом, забезпечуючи перекриття всього поля зору. Відповідні клітини при цьому виконують роль локальних фільтрів вхідних даних, реагуючи на присутність у власному рецептивної поле деяких примітивних структур, таких як краю і кордони. Було виявлено також існування так званих «складних клітин», мають більш широкі рецептивні поля, і демонстрували інваріантність по відношенню до точного розташування об'єкта в поле зору. С учетом того, что визуальная кора головного мозга представляет собой наиболее мощную и гибкую зрительную систему из существующих на данный момент, появление моделей, эмулирующих ее поведение, выглядело естественным шагом. Одной из наиболее успешных моделей, считающейся признанным лидером в области распознавания изображений, является сверточная нейронная сеть.

Згорткові мережі являють собою варіацію архітектури багат шарового перцептрона, і включають в себе згорткові шари, шари підвибірки (Субдіскретізація), і повнозв'язні шари. Архітектура сверточное мережі використовує переваги двовимірної структури вхідних даних - зображень за допомогою методу локальної зв'язності, обмежуючи кількість зв'язків між нейронами прихованого сверточного шару і вхідними даними. Конкретно, кожен нейрон прихованого шару пов'язаний тільки з обмеженим локальним (які не мають розривів) ділянкою зображення.

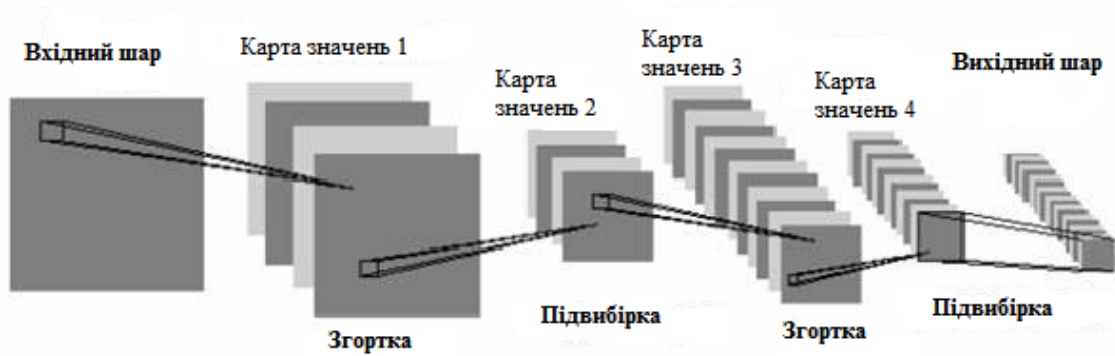


Рис. 2.7 – Архітектура згорткової нейронної мережі

Крім цього, нейронна мережа використовує загальні, або розділяються ваги, накладаючи штучне обмеження на алгоритм навчання зворотним поширенням помилки, так, щоб кожен нейрон прихованого шару мав набір ваг, спільний з іншими нейронами цього шару. При прямому поширенні така мережа здійснює математичну операцію згортки вхідного зображення набором фільтрів, які подаються вагами нейронів прихованого шару. Проміжними результатами мережі є так звані «карти ознак» - двовимірні матриці, що представляють собою результат згортки окремим фільтром.

Шар підвибірки, або Субдіскретизація, виконує операцію угруповання карт ознак, розглядаючи регіони розміру $\rho * \rho$ і агрегує значення, отримані в результаті згортки. Основне призначення Субдіскретизація - знизити варіативність даних, забезпечуючи стійкість до трансляції локального ознаки в межах окремого регіону. Таким чином, в умовах, коли один і той самий ознака виявляється зміщений на деяке значення $(\Delta x, \Delta y)$ в межах, що не, що перевищують ρ , відповідний нейрон, що інкапсулює локальний субдіскретизований ознака зображення, буде як і раніше активний. Таким чином мережа забезпечує інваріантність до просторових спотворень. Як агрегує функції шару Субдіскретизація зазвичай розглядається функція обчислення середнього або максимального значення.

Таким чином, значення рахунок використання декількох поперемінних шарів згортки і Субдіскретизація сверточної мережу дозволяє отримувати уявлення, незалежні від конкретного розташування локального ознаки в

зображенні, і однаковим чином реагувати на об'єкти, що цікавлять (наприклад, людські обличчя), присутні на будь-якій ділянці фотографії.

Згорткові мережі - один з найбільш успішних існуючих на сьогоднішній день алгоритмів розпізнавання зображень. Моделям, які реалізують відповідну архітектуру, належать перші місця в змаганнях алгоритмів розпізнавання, таких як ImageNet: згорткові мережі розпізнають рукописні цифри вибірки MNIST з 0.23% помилок, людські обличчя з 2.4% помилок (використовувалася вибірка розміром в 5600 фотографій осіб більш ніж 10 осіб); результат змагання ImageNet для переможця 2014 року - сверточное мережі GoogLeNet компанії Google становить 0.06656% помилкових розпознавань, що представляє собою найкращий результат на поточний момент і можна порівняти з помилкою, що допускається людиною на вибірці ImageNet. Серед недоліків виділяють труднощі при обробці маленьких об'єктів, і нездатність справлятися з спотвореннями, такими як розмиває фільтр або сильний шум (такі спотворення присутні в навколишньому світі, наприклад, при погляді через товсте скло). При цьому згорткові мережі порівняно легко справляються з проблемами високоточного розпізнавання, які викликають труднощі у людей - наприклад, розпізнавання окремих моделей машин або порід собак, і інші завдання, що вимагають виділення узкоспеціфічних ознак. Крім того, без знання параметрів локалізації, сверточное мережу вразлива до узагальненого класу проблем цілісності, тобто модель, навчена деяким локальним ознаками зображеного об'єкта, буде позитивно класифікувати штучні зображення, де відповідні елементи розташовані в хаотичному порядку - помилка, яку з легкістю уникає людина[2].

Операція згортки забезпечує інваріантність по відношенню до трансляції локальних ознак (зрушенню по осях x і y). Архітектура згортальних мереж не передбачає стійкості до інших афінних перетворень, таким як обертання, дзеркальне відображення і масштабування. Для вирішення цієї проблеми, як правило, використовуються евристичні методи (вирівнювання зображення по лінії горизонту, використання просторових пірамід і різних відображених копій оригіналу). Перетворення, які не є афінними по відношенню до плоского

зображення, такі як обертання об'єкта в тривимірній площині, аналогічним чином не можуть бути оброблені за допомогою операції згортки. Для об'єктів, представлених в різних орієнтаціях, моделі доводиться навчатися різним, непересічним наборам ознак. Згорткові нейронні мережі та інші підходи, які використовують локальні інваріантні ознаки, в процесі розпізнавання періодично вимушено навчаються ознаками, максимізує інваріантність в рамках представленої вибірки, але не забезпечують дискримінативне уявлення об'єкту в цілому.

Альтернативні підходи до виділення локальних ознак включають в себе методи класичного комп'ютерного зору, які не використовують навчальні моделі. Ці методи здійснюють пошук на зображенні характерних ділянок, що відповідають алгоритмічно явно заданим умовам. Серед них виділяються:

1. Виявлення країв / кордонів. Краєм називається ділянку зображення, представляє собою кордон між двома контрастними регіонами, помітну людським оком. Математично точки, складові такі ділянки, визначаються як точки, де градієнт зображення має локальний максимум. Дослідження як в області функціонування біологічних зорових систем, так і теорії інформації, показують, що репрезентація об'єктів в полі зору за допомогою кордонів є ефективний з точки зору мінімізації ентропії спосіб зберігання і обробки інформації і може використовуватися для компактної репрезентації зображення. Крім цього, виділення кордонів дозволяє знизити вплив деяких факторів, які не впливають на розпізнавання, таких як освітлення і тіні. Виділення меж (за допомогою фільтрів Кенні, Собеля, або згортки вейвлетами Габора) часто використовується як попередній етап обробки зображень в інших алгоритмах розпізнавання, в тому числі - згортальних мережах.

2. Виявлення кутів або «точок інтересу». До цієї групи належать алгоритми, які виділяють локальні ділянки зображення, максимально чутливі до змін. Відповідне умова формулюється наступним чином: якщо розглянути функцію ковзного вікна $\omega(x, y)$, то для будь-яких комбінацій значень зсуву (u, v) в двовимірній площині зображення знайдемо такі x та y , які максимізують $E(x, y) =$

$\sum_{u,v} \omega(x, y) [I(x + u, y + v) - I(x, y)]^2$ де I — інтенсивність зображення в межах, заданих функцією вікна. Традиційно ця група алгоритмів (що включає в себе детектор Харріса, детектор Ши-Томасі і інші) використовувалася для відшукування кутів між прямими лініями, але в даний момент розглядаються також будь-які точки з високим значенням кривизни.

3. Виявлення ділянок неоднорідності. Під ділянками неоднорідності, на відміну від кутів, розуміються деякі безперервні регіони зображення, які відрізняються за значеннями кольору або інтенсивності від навколишнього фону, і при цьому схожі між собою. Як правило, такі ділянки відповідають локальним екстремумам зображення. Перераховані локальні ознаки широко використовуються в задачах візуального трекінгу і стеження за об'єктом, але в чистому вигляді непридатні для задачі розпізнавання в силу своєї недіскримінаційної природи - такі методи не надає можливості відрізнити один кут (або ділянку неоднорідності) від іншого і висловити відмінність або схожість в числовому еквіваленті. Цим вимогам, проте, задовольняють підходи, які використовують ідею відшукування точок інтересу з використанням локальних дескрипторів, і представлені такими алгоритмами як SIFT, SURF і ORB.

Дескриптор є композицією ділянок зображення, локалізованих спільно, де для кожної ділянки або блоку розраховуються параметри орієнтації, масштабу, і деякі інші, що дозволяють вероятностно ідентифікувати місце розташування ознаки, яке відповідає даному дескриптору. З урахуванням використання в дескрипторах параметрів орієнтації та масштабу, такі ознаки виявляються інваріантними по відношенню до обертання зображення, зміни масштабу і яскравості / контрасту. Такі алгоритми як SIFT і ORB, крім того, забезпечують можливість зіставлення зображень, співвідносячи однакові локальні ознаки один з одним.

Ознаки, що використовують локальні дескриптори, можуть ефективно використовуватися для розпізнавання зображень одного і того ж об'єкта під впливом афінних перетворень в тривимірному просторі.



Рис. 2.8 – Об'єкти, представлені локальними ознаками SIFT, демонструють високу стійкість до просторовим перетворенням, і оклюзії

Локальність ознак дає можливість справлятися з проблемою оклюзії (див. Рис. 1.5), забезпечуючи можливість зіставляти об'єкти по частинах. Основні проблеми методів сімейства SIFT - слабка стійкість до варіативності, що не дозволяє алгоритму відносити до одного класу об'єкти, візуально відрізняються формою текстурою. Методи виділення локальних ознак дозволяють справлятися з деякими класами проблем розпізнавання зображень, забезпечуючи стійкість до оклюзії, знижуючи обчислювальну навантаження при обробці зображень високої розмірності і дозволяючи формувати інваріантні ознаки для виявлення об'єктів під дією інваріантних перетворень. При цьому нездатність виробляти локалізацію об'єктів на зображенні і вразливість до хаотичного розташування локальних ознак привела до появи методів, що враховують просторові відносини між локальними ознаками при розпізнаванні.

2.5 Багатошаровий перцептрон

Багатошаровим перцептроном називають штучну нейронну мережу, що складається з декількох вхідних вузлів, що утворюють вхідний шар, одного або декількох обчислювальних шарів нейронів і одного вихідного шару (рис. 2.9). У

таких мережах сигнал, що подається на вхідний шар, передається послідовно в прямому напрямку від шару до шару. Даний тип ШНМ успішно застосовується для вирішення різноманітних завдань, зокрема для задачі розпізнавання образів.

Нейронна мережа зворотного поширення помилки складається з декількох шарів нейронів, причому кожен нейрон попереднього шару пов'язаний з кожним нейроном наступного шару. У таких мережах, після визначення кількості шарів і кількості елементів кожного шару, потрібно обчислити значення ваг і порогів мережі таким чином, що б мінімізувати помилку прогнозу. Дане завдання вирішується за допомогою різноманітних алгоритмів навчання. Суть цих алгоритмів полягає в підгонці мережі до навчальних даних. Помилка реалізованої мережі буде визначатися шляхом прогону всіх вхідних даних і порівнянням реально отриманих значень на виході мережі з цільовими значеннями. Потім полеченої різниці підсумовуються в загальну, так звану, функцію помилок, яка характеризує загальну помилку мережі. Але частіше, як функції помилок, береться сума квадратів помилок.

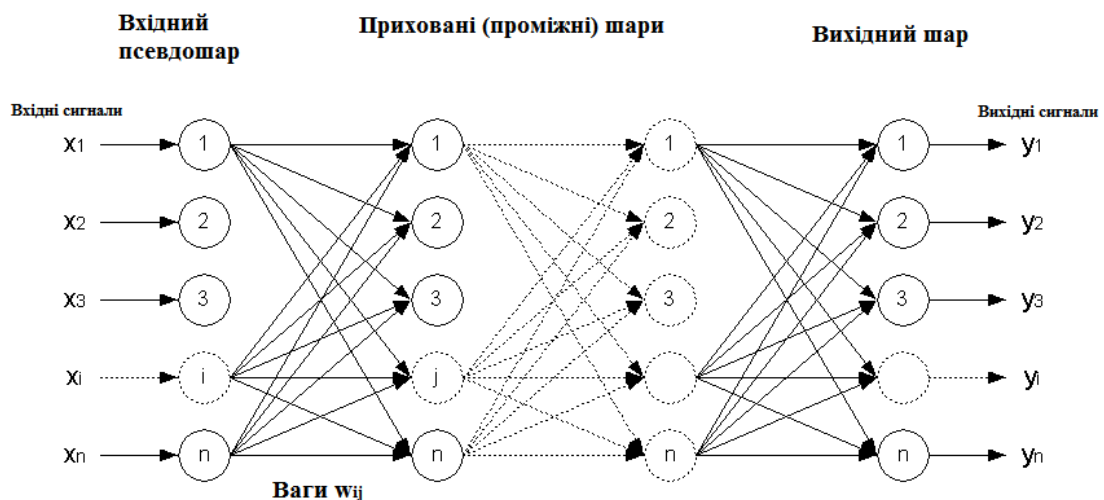


Рис. 2.9 Багатошаровий перцептрон

Одним з найбільш поширених алгоритмів навчання багатошарових нейронних мереж є алгоритм зворотного поширення помилки. У цьому алгоритмі обчислюється вектор градієнта поверхні помилок. Потім просуваємося на деяку

величину в напрямку вектора (він буде вказувати нам напрямок найшвидшого спуску), де значення помилки буде вже менше. Таке послідовне просування поступово призведе до мінімізації помилки. Тут виникають труднощі з визначенням величини, на яку слід просуватися. Якщо величина кроку буде відносно великий, це призведе до найшвидшому спуску, проте є ймовірність «перестрибнути» шукану точку або піти в неправильному напрямку, якщо поверхня має досить складну форму. Наприклад, в разі, якщо поверхня являє собою вузький яр з крутими схилами, алгоритм буде дуже повільно просуватися, стрибаючи з одного схилу на інший. Якщо ж величина кроку буде маленькою, це призведе до знаходження найбільш оптимального напрямку, однак може значно збільшити число ітерацій. Для досягнення найбільш оптимального результату величина кроку береться пропорційно крутизні схилу з деякою константою-швидкістю навчання. Вибір цієї константи здійснюється експериментальним шляхом і залежить від умов конкретного завдання.

Введемо наступні позначення. Матрицю вагових коефіцієнтів від входів до прихованого шару позначимо W , а матрицю ваг, що з'єднують прихований і вихідний шар - V . Для індексів приймемо такі позначення: входи будемо нумерувати тільки індексом z , елементи прихованого шару - індексом u , а виходи - індексом k . Число входів мережі одно p , число нейронів в прихованому шарі - t , число нейронів у вихідному шарі - r . Нехай мережа навчається на вибірці (X^t, D^t) , $t = \overline{1T}$. Тоді алгоритм навчання багат шарового персептрона буде виглядати наступним чином :

Крок 1. Ініціалізація мережі.

Ваговим коефіцієнтам привласнюються малі випадкові значення, наприклад, з діапазону $(-0.3, 0.3)$; задаються ε - параметр точності навчання, α - параметр швидкості навчання (як правило, ≈ 0.1 і може ще зменшуватися в процесі навчання), N_{max} - максимально допустиму кількість ітерацій.

Крок 2. Обчислення поточного вихідного сигналу.

На вхід мережі подається один з образів навчальної вибірки, і визначаються значення виходів усіх нейронів нейромережі.

Крок 3. Налаштування синаптичних ваг.

Розрахувати зміну ваг для вихідного шару нейронної мережі за формулами:

$$V_{jk}^{N+1} = V_{jk}^N - \alpha \frac{\partial E}{\partial V_{jk}}, \quad (2.1)$$

де $\frac{\partial E}{\partial V_{jk}} = \delta_k y_j^c$, $\delta_k = (y_k - d_k) y_k (1 - y_k)$.

Розрахувати зміну ваг для вихідного шару нейронної мережі за формулами::

$$W_{ij}^{N+1} = W_{ij}^N - \alpha \frac{\partial E}{\partial W_{ij}}, \text{ де } \frac{\partial E}{\partial W_{ij}} = \left(\sum_{k=1}^3 \delta_k V_{jk}^{N+1} \right) y_j^c (1 - y_j^c) X_i \quad (2.2)$$

Крок 4. Кроки 2-3 повторюються для всіх навчальних векторів. Навчання завершується після досягнення для кожного з навчальних образів значення функції помилки, які не перевершує ϵ або після максимально допустимого числа ітерацій.

На кроці 2 вектори з навчальної послідовності краще пред'являти на вхід у випадковому порядку.



Рис 2.10. Схема навчання мережі зворотного поширення

Кількість входів і виходів мережі, як правило, диктується умовами задачі, а розмір прихованого шару знаходять експериментально. Зазвичай число нейронів в ньому становить 30-50% від числа входів. Занадто велика кількість нейронів прихованого шару призводить до того, що мережа втрачає здатність до узагальнення (вона просто досконально запам'ятовує елементи навчальної вибірки і не реагує на подібні зразки, що є неприйнятним для завдань розпізнавання). Якщо число нейронів в прихованому шарі занадто мало, мережа виявляється просто не в змозі навчитися[19].

2.6 Мережа Кохонена

Розглянемо наступний тип нейронних мереж - мережа Кохонена. Даний вид мережі застосовується при вирішенні задач класифікації, які складаються у разі необхідності розділення об'єктів на класи, причому в якості основи для розбиття береться вектор параметрів, а кількість класів, як правило, заздалегідь не відомо і формується динамічно. Прототипом будемо називати об'єкт найбільш підходящий для свого класу. Один з методів вирішення задачі класифікації передбачає визначення декількох класів і вибір випадковим чином координат прототипів. Далі, кожен вектор зв'язується з найближчим прототипом, а з центроїдів кожного вектора, пов'язаних з вихідним зразком, формуються нові прототипи. Близькість векторів, в цьому випадку, визначається як евклідова відстань:

$$d(x, y) = \sum_i (x_i - y_i)^2 \quad (2.3)$$

Це основний метод побудови мережі Кохонена, яка навчається без учителя на принципах самоорганізації. Таке навчання є формування класів із прототипів, отриманих з векторів ваг нейронів. У процесі ж рішення реальних завдань, мережа відносить вхідний образ до одного з сформованих раніше класів. Розглянемо структуру мережі Кохонена більш детально.

І так, мережа Кохонена складається з m нейронів, кількість яких збігається з необхідною кількістю класів. У процесі виконання завдання, кількість нейронів може змінюватися, тобто кількість класів може бути визначено динамічно. Кожен

нейрон має по n входів. Число входів в нейроні визначає розмірність вектора параметрів об'єкта. Процес навчання починається з ініціалізації ваговій матриці W , стовпці якої є параметри нейронів-класифікаторів. Потім, навчання зводиться до підстроювання ваг, при подачі на вхід векторів з навчальною виборкою. Для кожного j -го нейрона ($1 \leq j \leq m$) визначається відстань від нього до вхідного вектора X :

$$d_j = \sum_{i=1}^n (x_i - w_{ij})^2 \quad (2.4)$$

Потім, знаходимо k -й нейрон, до якого мінімальна відстань, тобто мережу віднесла вхідний вектор до класу k . Таким чином, на N -й ітерації, будуть змінюватися тільки ваги з околиці нейрона до:

$$W_{ij}^{N+1} = W_{ij}^N + \alpha_N (x_i - w_{ij}^N) \quad (2.5)$$

Спочатку, в околиці кожного нейрона знаходиться вся нейронна мережа, однак з процесом навчання, ця околиця звужується і в кінцевому підсумку, на останніх кроках навчання, підлаштовуються тільки ваги k -го нейрона. швидкість навчання α_N є величиною динамічною і як правило, на початковому етапі навчання, вважаю, що вона дорівнює 0,9, а з кожним наступним кроком, її значення зменшують на 0,001.

Розглянемо алгоритм навчання мережі Кохонена:

Крок 1. Ініціалізація мережі.

Ваговим коефіцієнтам мережі $W_{ij} \quad i = \overline{1, n}, j = \overline{1, m}$ присвоюються малі випадкові значення. Задаються значення α_0 - початковий темп навчання і D_0 - максимальна відстань між ваговими векторами (стовпцями матриці W).

Крок 2. Пред'явлення мережі нового вхідного сигналу X .

Крок 3. Обчислення відстані від входу X до всіх нейронів мережі:

$$d_j = \sum_{i=1}^n (X_i - W_{ij}^N)^2, j = \overline{1, m} \quad (2.6)$$

Крок 4. Вибір нейрона k , $1 \leq k \leq m$ з найменшою відстанню d_k .

Крок 5. Налаштування ваг нейрона k і всіх нейронів, що знаходяться від нього на відстані, що не перевищує D_N :

$$W_{ij}^{N+1} = W_{ij}^N + \alpha_N (X_i - W_{ij}^N) \quad (2.7)$$

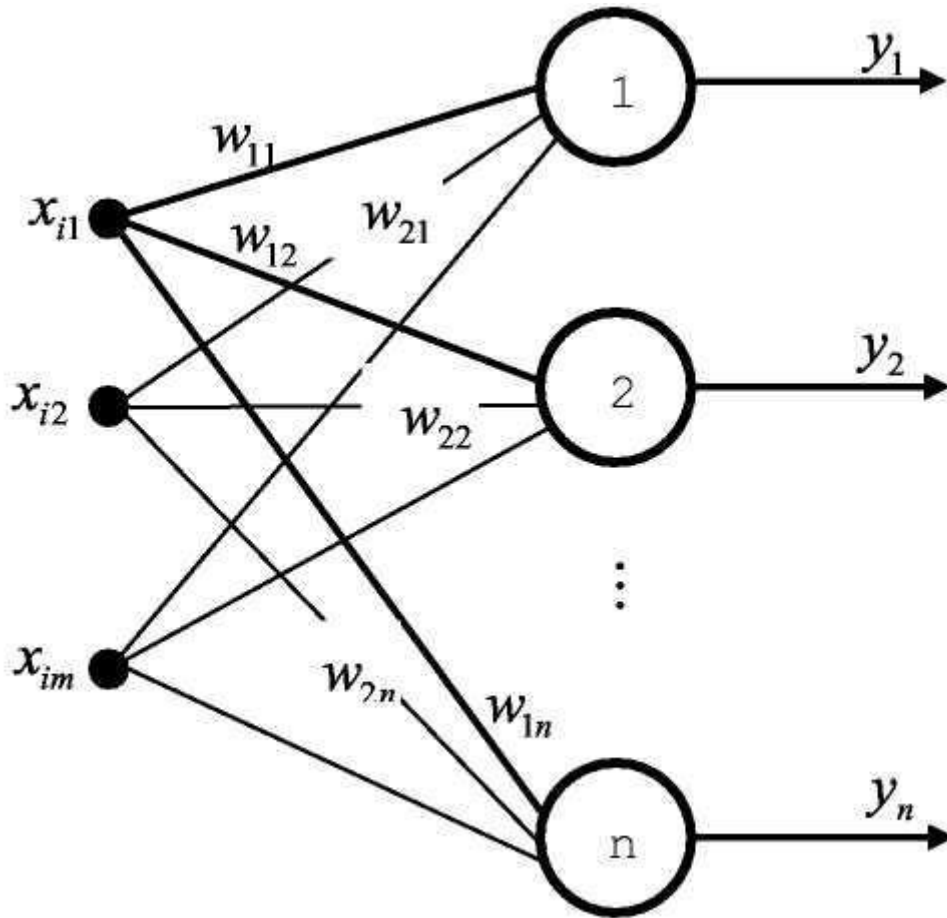


Рис. 2.11. Структура мережі Кохонена

Крок 6. Зменшення значень α_N, D_N .

Крок 7. Кроки 2-6 повторюються до тих пір, поки ваги не перестануть змінюватися (або поки сумарна зміна всіх ваг стане дуже мало). Після навчання класифікація виконується за допомогою подачі на вхід мережі випробуваного вектора, обчислення відстані від нього до кожного нейрона з подальшим вибором нейрона з найменшою відстанню як індикатора правильної класифікації.

Зауваження: якщо попередньо провести одиничну нормування всіх вхідних векторів, тобто подавати на вхід мережі образи X' , компоненти якого пов'язані з компонентами вектора X за формулами:

$$X'_i = \frac{X_i}{\sqrt{X_1^2 + X_2^2 + \dots + X_n^2}} \quad (2.8)$$

а також якщо після кожної ітерації процесу навчання здійснювати нормування ваг кожного нейрона (стовпців матриці W), то в якості міри близькості вхідних векторів і вагових векторів нейронів мережі можна розглядати скалярний твір між ними. Дійсно, в цьому випадку:

$$d_j = \sum_{i=1}^n (X_i - W_{ij}^N)^2 = \sum_{i=1}^n X_i^2 - 2 \sum_{i=1}^n X_i W_{ij}^N + \sum_{i=1}^n W_{ij}^2 = 2 - 2 \sum_{i=1}^n X_i W_{ij}^N \quad (2.9)$$

Таким чином, найменшим буде відстань до того нейрона, скалярний твір з вагами якого у вхідного вектора максимально. В цьому випадку можна вважати, що кожен нейрон Кохонена реалізує тотожну активаційну функцію $f(s) = s$, где $s = \sum_{i=1}^n W_{ij} * X_i$. Нейрон з максимальним значенням активаційної функції оголошується "переможцем" і його ваги (а також ваги нейронів з його оточення) перераховуються.



Рисунок 2.12 Схема навчання мережі Кохонена

2.7 Мережа Хеммінга

Серед безлічі різних видів штучних нейронних мереж, існують такі мережі, які неможливо віднести до жодної з категорій за типом навчання: ні до мереж, які навчаються без учителя, ні до мереж, які навчаються з учителем. У таких мережах настройка вагів проводиться тільки один раз-перед початком роботи на реальному безлічі. Налаштування проводиться на основі аналізу даних про вхідному безлічі, а власне процес навчання зводиться до одноразового розрахунку вагових коефіцієнтів. Таким чином, з одного боку ми отримуємо свідомо справжню інформацію на вході і мережу як би проходить одну епоху навчання, з іншого боку, дана зміна вагових коефіцієнтів не є навчанням, тому що змінивши вагові коефіцієнти один раз, мережа більше не звертається до навчального безлічі. Прикладом мережі з такою архітектурою є мережа Хеммінга [25].

Нехай m - число образів, які може запам'ятовувати мережу. Мережа Хеммінга складається з двох шарів, в кожному з яких по m нейронів. В мережі знаходиться n входів, кожен з яких з'єднаний з усіма нейронами першого шару. Величини, що подаються на вхід мережі, є біполярними, тобто належать множині $\{-1;1\}$. Структура другого шару являє собою шар нейронів, пов'язаних один з одним негативними зворотними зв'язками. Єдиною позитивним зворотним зв'язком кожен нейрон пов'язаний з власним виходом.

Розглянемо принцип функціонування даної мережі. Для цього ведемо поняття відстані Хеммінга, під яким будемо розуміти число відрізняються бітів в двох бінарних векторах. Основна ідея роботи мережі Хеммінга полягає в оцінці величини, зворотної відстані Хеммінга від тестованого образу до всіх еталонних зразків. При аналізі вхідного образу, мережа повинна знайти еталонний зразок з мінімальною відстанню Хеммінга до невідомого вхідного сигналу і форматувати єдиний вихід мережі, що відповідає одному з еталонних зразків. Як активационной функції для нейронів першого шару є лінійна функція виду:

$$f(s) = s + \frac{n}{2} = \sum_{i=1}^n W_{ij} X_i + \frac{n}{2} \quad (2.10)$$

де n - кількість входів мережі Хеммінга.

Активаційна функція для нейронів другого шару являє собою граничну функцію виду:

$$y = f(s) \begin{cases} 0, s < 0 \\ s, 0 \leq s < T \\ T, s \geq T \end{cases} \quad (2.11)$$

де s - зважена група входів, T - величина порогу.

Значення порога T береться відносно великим, для виключення ситуації, в якій при певних значеннях аргументу відбувається насичення за одну ітерацію мережі.

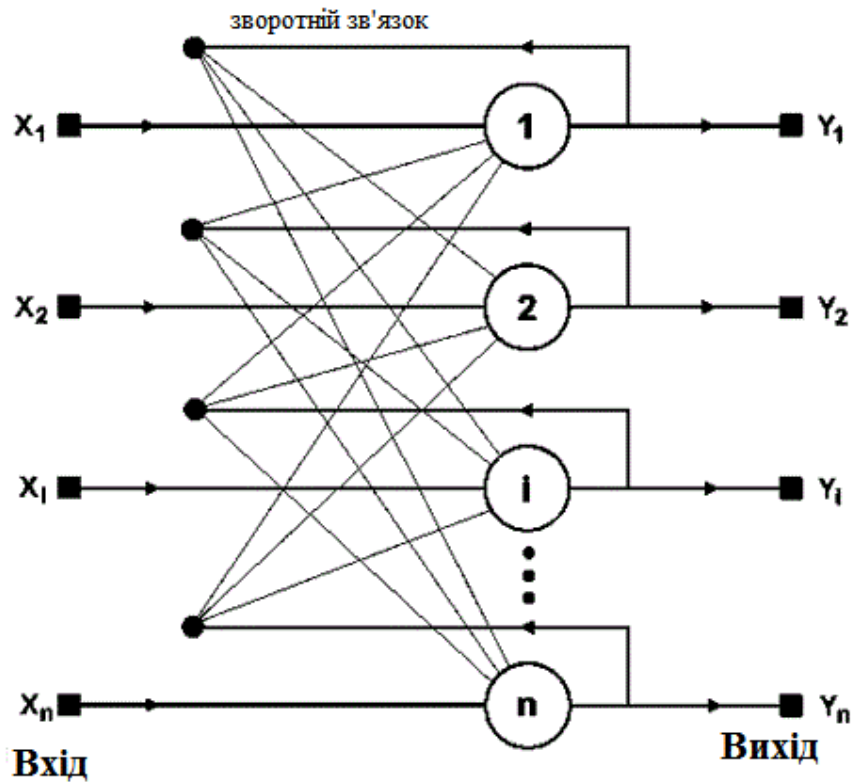


Рис. 2.13 Структура Мережі Хеммінга

В процесі ініціалізації мережі, вагові коефіцієнти обчислюються наступним чином:

$$w_{ik} = \frac{X_i^k}{2}; \in \overline{1, m}, k \in \overline{1, n} \quad (2.12)$$

де $X_i^k \in \{-1, 1\}$ - i -й елемент k -го вхідного образу.

Як значення вагових коефіцієнтів другого шару беруть не великі величини $\varepsilon \in (0, \frac{1}{m})$. Вага нейрона, пов'язаний з його ж виходом приймає значення +1.

Алгоритм функціонування мережі Хеммінга

На входи мережі подається невідомий вектор X , виходячи з якого, розраховуються стани (виходи) нейронів першого шару, які передаються на вхід другого шару.

Обчислюються виходи нейронів другого шару.

Перевіряється, чи змінилися виходи нейронів другого шару за останню ітерацію. Якщо так - виходи другого шару множаться на відповідні вагові коефіцієнти і передаються на входи цього ж шару, а потім перехід до кроку 2. Якщо немає - виходи стабілізувалися, і робота мережі завершена.

2.8 Нечіткі нейронні мережі. Мережа Ванга-Менделя

Принцип нечіткої логіки досить давно використовується для вирішення завдань, в яких вихідні дані є слабо формалізованими або ж ненадійними. Основними перевагами мереж з такою структурою є:

- зручність подання інформації: опис постановки задачі і умов проводиться на мові близькому до природного;
- універсальність: згідно теоремі нечіткої апроксимації, будь-яка математична модель може бути апроксимована системою, побудованою на нечіткій логіці;
- ефективність: ряд теорем, подібних теорем про повноту для штучних нейронних мереж, показують високу ефективність роботи таких мереж.

Однак, такої організації нейронних мереж притаманний і ряд недоліків:

- вихідний набір нечітких правил формується людиною, що не завжди об'єктивним, а іноді неповним або навіть суперечливим;
- вид і параметри даних, що зв'язують вхід і вихід, також визначаються суб'єктивно і не завжди відображають дійсність

Використовуваний в різних видах систем, модель на основі нечітко логіки є базою знань, побудовану фахівцям предметної області, як безліч нечітких правил виду:

$$P_1 \text{ якщо } x \in A_1, \text{ то } y \in B_1$$

$$P_2 \text{ якщо } x \in A_2, \text{ то } y \in B_2$$

...

$$P_n \text{ якщо } x \in A_n, \text{ то } y \in B_n$$

де x і y - вхідна і вихідна змінна відповідно, а A і B - функції приналежності.

Нечіткий логічний висновок формується в кілька кроків:

- введення нечіткості: на цьому етапі, функції приналежності застосовуються до фактичних значень вхідних змінних;
- логічний висновок: обчислюється значення істинності для передумов кожного правила і застосовується до висновків кожного правила. Це призводить до одного непевному підмножині, яке буде призначено кожної змінної виводу для кожного правила;
- композиція: нечіткі підмножини, призначені кожною змінною виведення об'єднують в одне безліч для всіх змінних виведення;
- приведення до чіткості: використовується у випадках, коли необхідний перетворити нечіткий набір висновків в чітке число

На цих принципах побудована велика кількість мереж, дозволить обговорити докладніше одну з них- мережу Ванга-Менделя [12]. Структура такої мережі є чотиришарову нейронну мережу, в якій перший шар виконує фазифікація вхідних змінних, другий - агрегування значень активації умови, третій (лінійний) - агрегування M правил виведення (перший нейрон) і генерацію нормалізує сигналу (друго нейрон), тоді як що складається з одного нейрона вихідний шар здійснює нормалізацію, формуючи вихідний сигнал[24].

В этой сети первый и третий слой являются параметрическими: первый слой содержит $M*N*2$ У цієї мережі перший і третій шар є параметричними: перший шар містить - M параметрів w_i .

Вихідний сигнал мережі Ванга-Менделя розраховується за формулою:

$$y(x) = \frac{\sum_{i=1}^M w_i \prod_{j=1}^N \mu_{ij}(x_j)}{\sum_{i=1}^M \prod_{j=1}^N \mu_{ij}(x_j)} \quad (2.13)$$

де w_1 - ваговий коефіцієнт, $\mu_y()$ - функція Гаусса з параметрами математичного очікування, яке визначає центр c_y і параметрами розкиду, які визначаються середнім квадратичним відхиленням.

$$\mu_{ij}(x_j) = e^{-\frac{(x_j - c_{ij})^2}{d_{ij}^2}} - \text{функція Гаусса.}$$

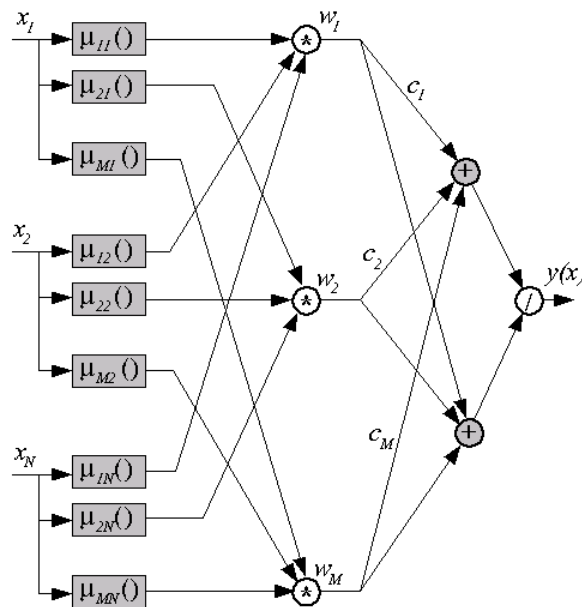


Рис. 2.14. Структура мережі Ванга-Менделя

Завдання мережі полягає в побудові такого відображення пар даних (x, d) , щоб очікуване значення, відповідне вхідному вектору x , формувалося вихідний функцією $y(x)$. Навчання нечітких мереж, так само як і класичних мереж, може проводитися за алгоритмом з учителем, заснованому на мінімізації цільової функції, що задається з використанням евклідової норми як $E = \frac{1}{2} \sum_{i=1}^p \left(y(x^{(l)} - d^{(l)}) \right)^2$, де p - кількість навчальних пар (x, d) [1].

Для навчання нечіткої нейронної мережі, застосовують алгоритм, що включає послідовне чергування наступних кроків:

- для фіксованих значень параметрів c_{ij} та d_{ij} першого шару

обчислюються значення параметрів w_i третього шару мережі;

- при зафіксованих значеннях параметрів w_i третього шару уточнюються параметри c_{ij} і d_{ij} першого шару мережі.

На другому етапі, фіксуються значення коефіцієнтів поліномів третього шару, і здійснюється уточнення (зазвичай багаторазове) коефіцієнтів функції Гаусса для першого шару мережі стандартним методом градієнта: $c_{ij}^{k+1} = c_{ij}^k - v_c \frac{\partial E^k}{\partial c_{ij}^k}$, $d_{ij}^{k+1} = d_{ij}^k - v_d \frac{\partial E^k}{\partial d_{ij}^k}$, де k – номер чергового циклу навчання, v_c - швидкість навчання для коефіцієнтів c_{ij} , v_d - швидкість навчання для коефіцієнтів d_{ij} , $E = \frac{1}{2} \sum_{l=1}^L (y_l - y_l^e)^2$ - помилка мережі, де L - загальна кількість навчальних вибірок, y_l - вихід мережі Ванга-Менделя для даної вибірки, y_l^e - еталонне значення виходу мережі Ванга-Менделя.

Похідні $\frac{\partial E}{\partial c_{ij}}$ та $\frac{\partial E}{\partial d_{ij}}$ обчислюються за формулами:

$$\frac{\partial E}{\partial c_{ij}} = \sum_{l=1}^L (y_l - y_l^e) * \frac{\partial y_l}{\partial c_{ij}}, \quad \frac{\partial E}{\partial d_{ij}} = \sum_{l=1}^L (y_l - y_l^e) * \frac{\partial y_l}{\partial d_{ij}} \quad (2.13)$$

Похідні $\frac{\partial y}{\partial c_{ij}}$ и $\frac{\partial y}{\partial d_{ij}}$ можна знайти за формулами:

$$\frac{\partial y}{\partial c_{ij}} = \frac{2 * (x_i - c_{ij}) * \prod_i \mu_{ij}(x_i) * \sum_j ((w_j - w_j) * \prod_i \mu_{ij}(x_i))}{d_{ij}^2 (\sum_j \prod_i \mu_{ij}(x_i))^2},$$

$$\frac{\partial y}{\partial d_{ij}} = \frac{2 * (x_i - d_{ij})^2 * \prod_i \mu_{ij}(x_i) * \sum_j ((w_j - w_j) * \prod_i \mu_{ij}(x_i))}{d_{ij}^3 (\sum_j \prod_i \mu_{ij}(x_i))^2}$$

де $\mu_{ij}(x_j) = e^{-\frac{(x_j - c_{ij})^2}{d_{ij}^2}}$ - Функція Гаусса.

Оскільки в низці етапів етап уточнення нелінійних параметрів функції Гаусса має багато меншу швидкість збіжності, то в ході навчання реалізацію етапу 1, як правило, супроводжує реалізація декількох етапів 2 [62, 85]. Часто потрібно знайти «рішення» системи, яка рішень (в звичайному сенсі) не має. Виходом із ситуації є знаходження таких значень невідомих параметрів, що всі умови системи виконуються «в деякій мірі».

Матриця A^+ називається псевдообернена матриця для матриці A , якщо $AA^+A = A$. Отсюда сразу вытекает, что если матрица A имеет размер $m \times n$, то псевдообратная матрица A^+ имеет размер $n \times m$.

Наведемо й інший, часто зустрічається в літературі, підхід до визначенню цього поняття. Спочатку введемо поняття псевдорішення системи рівнянь. Нехай нам дана система рівнянь

$$Ax=b, \quad (2.14)$$

де A - матриця розмірів $m \times n$, b - вектор з m елементів.

Будь-яке рішення цієї системи є також і рішенням системи

$$A^T Ax = A^T b, \quad (2.15)$$

До універсальних способів знаходження псевдообернена матриця відносяться рекурентні алгоритми Гревіль і Фадєєва. У даній роботі наведемо алгоритм Гревіль для псевдообернення матриць.

Нехай дана матриця $A \in R^{m \times n}$ и a_k - k -й стовбец, $k = 1, \dots, n$.

Нехай A_k - матриця, составленная из k -перших стовбців матриці A :

$$A_k = [a_1 \ a_2 \ \dots \ a_k]$$

При $k = 1$: $A_1 = a_1$, а при $k = 2, \dots, n$: $A_k = [A_{k-1} \ a_k]$; $A_n = A$

Матриця $A^+ \in R^{n \times m}$ Матриця може бути обчислена за допомогою рекурентного алгоритму:

Ініціалізація

$$A_1^+ = \begin{cases} 0, \text{ якщо } a_1 \\ a_1^T \\ \frac{a_1^T}{\|a_1\|^2}, \text{ інакше} \end{cases}$$

Цикл по $k = 2, \dots, n$.

$$A_k^+ = \begin{bmatrix} A_{k-1}^+ (I - a_k f_k^T) \\ f_k \end{bmatrix}, \text{ де } I - \text{одинична матриця порядку } m,$$

$$f_k = \begin{cases} \frac{c_k^T}{\|c_k\|^2}, c_k = (I - A_{k-1} A_{k-1}^+) a_k, c_k \neq 0, \\ \frac{a_k^T (A_{k-1}^+)^T A_{k-1}^+}{1 + \|A_{k-1}^+ a_k\|^2}, c_k = 0 \end{cases}$$

Отримана на останньому кроці матриця A_n^+ і є шукана псевдообернена матриця. Для адаптації даної мережі до вирішення поставленого завдання модифікуємо структуру мережі Ванга-Менделя таким чином, щоб мережа мала кілька виходів. Структура такої мережі аналогічна описаній вище, за винятком організації третього і четвертого шару (рис. 2.13). Так третій шар буде являти собою набір з декількох пар нейронів сумматоров, а четвертий шар - кілька нейронів-нормалізаторов, кількість яких буде відповідати кількості пар третього шару. Крім того, в такій структурі набір вагових коефіцієнтів буде являти собою не вектор, а матрицю.

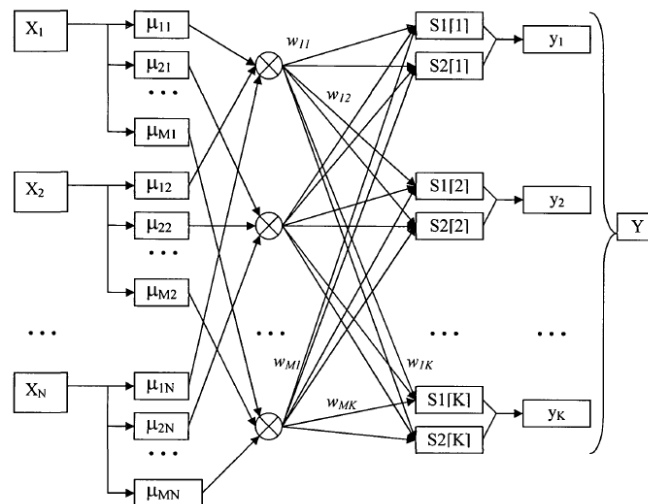


Рис. 2.15 Структура мережі Ванга-Менделя з декількома виходами

Виходи мережі з такою структурою обчислюються за формулою

$$y_l = S1[l]/S2[l], \quad (2.16)$$

де $S1[l] = \sum_{i=1}^M w_{il} \prod_{j=1}^N \mu_{ij}(x_j)$, $S2[l] = \sum_{i=1}^M w_{il} \prod_{j=1}^N \mu_{ij}(x_j)$, $l = 1 \dots K$

Висновки за розділом. Як показало наше дослідження, згорткові нейронні мережі досить точно виконують розпізнавання образів на статичних візуальних зображеннях. Тому за основу для подальшого практичного дослідження буде взятий саме даний тип нейронних мереж. В наступному розділі ставиться завдання довести їх ефективність.

3. РОЗРОБКА АЛГОРИТМУ РОЗПІЗНАВАННЯ ДЕФЕКТІВ ЗА ДОПОМОГОЮ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ

3.1 Постановка завдання розпізнавання

Нехай X - i -множина зображень, Y - i -множина непересічних класів (класів можливих дефектів), f^i - i -цільова функція, що відображає множину X на множину Y , $f^i: X \rightarrow Y$. Значення цільової функції y^i відомі тільки на кінцевій множині пар прецедентів (x_i, y_i) - навчальній вибірці. Запис $f^i(x_k) = y_k$ означатиме, що на зображенні x_k присутній дефект y_k . Необхідно відновити функціональну залежність між зображеннями і відповідями, побудувати алгоритм $A: X \rightarrow Y$, що володіє наступними властивостями:

- відображення A має допускати фіктивну чисельну реалізацію;
- на навчальній вибірці A повинен виробляти правильні відповіді;

A повинен володіти узагальнюючою здатністю, тобто наближати цільову функцію не тільки на об'єктах навчальної вибірки, а й на всій множині X .

Розглянуті в попередньому розділі методики розпізнавання не відповідають встановленим вимогам, тому потрібно розробити алгоритм, що вирішує завдання розпізнавання, який буде задовольняти цим вимогам.

3.2 Алгоритм вирішення задачі

Попередня обробка зображення. Отримане від користувача зображення перетворюється в півтонове, видаляється шум з зображення. Зображення готується для пошуку дефектів на ньому.

Сегментація зображення, пошук можливих дефектів. Виявлення можливих місць розташування дефектів на зображенні з метою подальшого їх розпізнавання нейронною мережею обробка зображення. Видалення ефекту нерівномірної освітленості зображення за допомогою адаптивної еквалізації гістограми.

Розпізнавання дефекту. Використання навченої згорткової нейронної мережі для розпізнавання дефектів.

Алгоритм представлений на рис. 3.1.

Так як нейронна мережа приймає на вхід зображення розмірністю 32×32 , то потрібно витягти з вихідного зображення підвікна потрібного розміру, для цього потрібно сегментувати вихідне зображення таким чином, щоб дефект максимально повно займав частину зображення такої розмірності. Іншими словами, потрібно знайти прямокутник, в який вписується дефект. Ця процедура називається детектуванням. Детектований дефект при необхідності масштабується, сегментується і подається на вхід нейронної мережі.

Детектування можливого місця розташування дефекту відбувається в кілька етапів. Передобробка заснована на припущенні про крос-кореляцію підвікон вихідного зображення і шаблонних дефектів. Тому спочатку проводиться згортка вихідного зображення з шаблонним дефектом, а потім на результаті згортки шукаються центри локальних максимумів, які будуть відповідати можливим місцезнаходженням дефектів. Алгоритм детекції см. На рис. 3.2.



Рисунок 3.1 - Алгоритм розв'язання задачі розпізнавання

3.3 Побудова моделей допоміжних методів розпізнавання (SVM, Feed Forward MLP)

Для того, щоб проаналізувати якість роботи моделей, заснованих на згорткових нейронних мережах, необхідно розробити і оцінити якість роботи моделей, заснованих на інших алгоритмах і методах розпізнавання. В якості допоміжних моделей виберемо метод опорних векторів з «мішком слів» (SVM with bag of features) і багатопшарову нейронну мережу прямого поширення (Feed Forward MLP).

Так як перед нами стоїть завдання багатокласової класифікації (маються 3 класу: поря (непровар), тріщина і не дефект), то при побудові мульти класового SVM будемо використовувати мішок ознак [31], який є модифікацією мішка слів, використовуваного при аналізі та класифікації текстової інформації[19].

В якості ознак виступають особливі (ключові) точки зображення, для пошуку і опису яких використовується алгоритм SURF (Speed Up Robust Features). SURF вирішує два завдання - пошук особливих точок зображення і створення їх дескрипторів, які інваріанти до масштабу і обертання, тобто опис особливої точки буде однаково, навіть якщо зразок змінить розмір і буде повернений. Також і сам пошук особливих точок повинен володіти інваріантністю для того, щоб повернений об'єкт сцени мав той же набір ключових точок, що і зразок. Метод шукає особливі точки за допомогою матриці Гессе. Детермінант матриці Гессе досягає свого екстремуму в точках, в яких максимально зміна градієнта яскравості (рисунок 2.3). З малюнка видно, що особливі точки (окреслені кольоровими колами) представляють собою локальні екстремуми яскравості зображення. Дрібні точки не розпізнано як особливі, через порогового відсікання по величині гессіан. Даний метод добре детектує плями, кути і краю ліній. Однак є суттєва проблема: гессіан інваріантний щодо обертання, але не інваріантний щодо масштабу [25]. Для вирішення даної проблеми алгоритм SURF використовує різномасштабні фільтри для знаходження гессіанів.

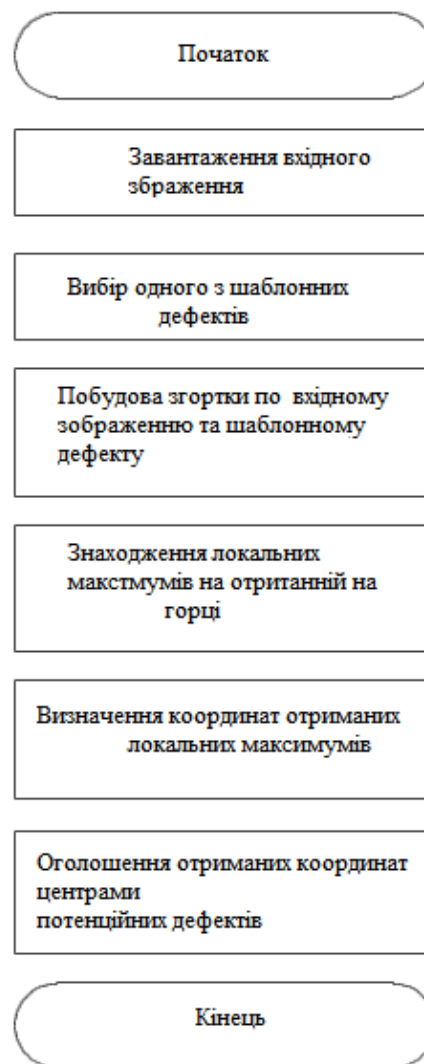


Рисунок 3.2 - Детекція можливих місць розташування дефектів

Для кожної ключової точки вважається напрямок градієнта яскравості і масштаб, взятий з scale-коефіцієнтів матриці Гессе. Градієнт в ключовій точці обчислюється за допомогою фільтрів Хаара [20].

Далі, коли алгоритм SURF знайшов ключові точки, він формує їх дескриптори [17]. Дескриптори являють собою набір з 64 або 128 чисел для кожної особливої точки. Ці числа відображають напрямок градієнта навколо ключової точки. Оскільки ключові точки являють собою максимум гессіан, то це гарантує, що в околиці точки повинні бути ділянки з різними градієнтами. Напрямки градієнтів також вважаються за допомогою фільтрів Хаара.

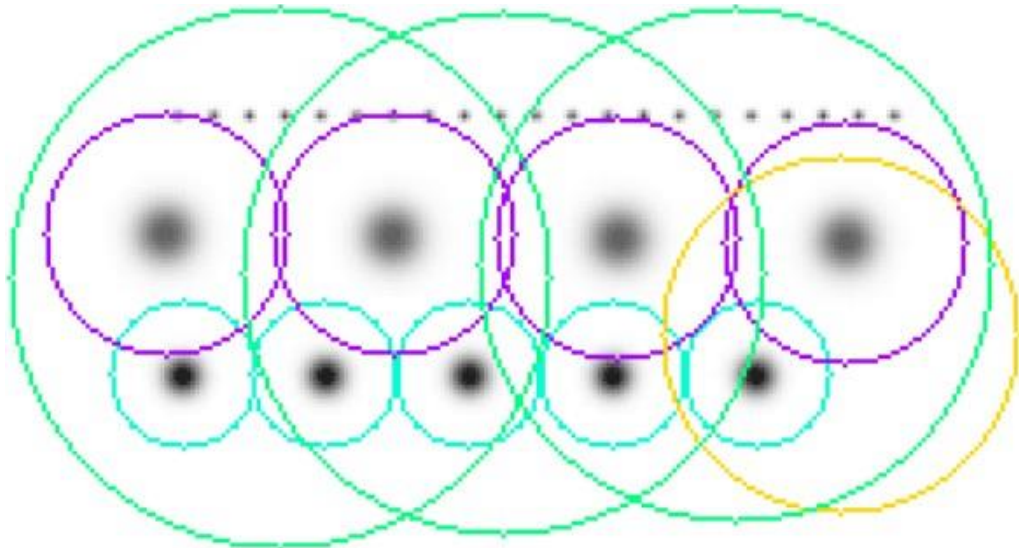


Рисунок 2.3 - Локальні екстремуми яскравості

Розмір фільтра Хаара береться рівним $4s$, де s - масштаб особливої точки. Приклад фільтрів Хаара наведено на малюнку нижче (рисунок 2.4).



Рисунок 2.4 - Фільтри Хаара

Чорні області мають значення -1 , білі $+1$. Фільтри Хаара дають точкове значення перепаду яскравості по осі X і Y відповідно. Оскільки фільтри Хаара мають прямокутну форму, їх значення легко вважаються.

Інший досить поширений метод розпізнавання зображень дефектів - багатосаровий перцептрон прямого поширення (Feed Forward MLP). Можна використовувати три різних алгоритму: алгоритм Левенберга-Марквардта, метод Байєсова, метод масштабованих сполучених градієнтів (scaled conjugate gradient). Будемо використовувати метод масштабованих сполучених градієнтів, так як він показує найкращу швидкість роботи і споживає мало пам'яті.

Нехай w - матриця ваг мережі, $F(w)$ - помилка навчання мережі (як і в попередніх методах - середньоквадратична помилка).

Опишемо алгоритм мінімізації $F(w)$.

Вибирається початкове наближення w_1 і встановимо $k = 1$.

Визначається напрямок пошуку p_k і величина кроку a_k так, щоб виконувалася нерівність:

$$F(w_k + a_k p_k) < F(w_k) \quad (3.1)$$

Змінюється вектор $w_{k+1} = w_k + a_k p_k$.

Якщо $F'(w_k) \neq 0$, тоді $k = k + 1$ і здійснюється перехід до кроку 2; інакше w_{k+1} - шуканий мінімум.

Метод сполучених градієнтів в своїй основі містить вищеописаний алгоритм мінімізації, змінюється лише спосіб вибору величини кроку. Цей метод передбачає, що задана система сполучених векторів. Але немає необхідності знати вектора p_1, p_2, \dots, p_k заздалегідь, так як вони можуть бути визначені рекурсивно.

Наведемо алгоритм сполучених градієнтів, так як метод масштабованих сполучених градієнтів буде ґрунтуватися на ньому.

Вибирається початкове наближення w_1 . встановлюється

$$p_1 = r_1 = -F'(w_k), \quad k = 1. \quad (3.1)$$

Обчислюються s_k і δ_k за формулами:

$$s_k = F''(w_k) p_k \quad (3.2)$$

$$\delta = p^T s$$

Обчислюється величина кроку a_k за допомогою формул:

$$\mu_k = p_{kk}^T r \quad (3.3)$$

$$a_k = \frac{\mu_k}{\delta_k}$$

Здійснюється крок в напрямку p_k :

$$w_{k+1} = w_k + a_k p_k$$

$$r_{k+1} = -F'(w_{k+1})$$

Якщо $k \bmod N = 0$, то алгоритм перезапускається (8), інакше за формулами (9 - 11) знаходиться нове поєднане напрямком.

$$p_{k+1} = r_{k+1}$$

$$\beta_k = \frac{|r_{k+1}|^2 - p_{k+1}^T r_k}{\mu_k}, \quad (3.3)$$

$$p_{k+1} = r_{k+1} + \beta_k p_k$$

Якщо напрямок найшвидшого спуску $r_k \neq 0$, тоді $k = k + 1$ і здійснюється перехід до кроку 2. Інакше повертається w_{k+1} як досягнутий мінімум.

Структура нейронної мережі показана на малюнку (рис 2.5). Мережа має один прихований шар і один вихідний шар, що складається з 3-х нейронів. Кожен з виходів мережі показує оцінку впевненості класифікатора в тому, до якого класу належить зображення.

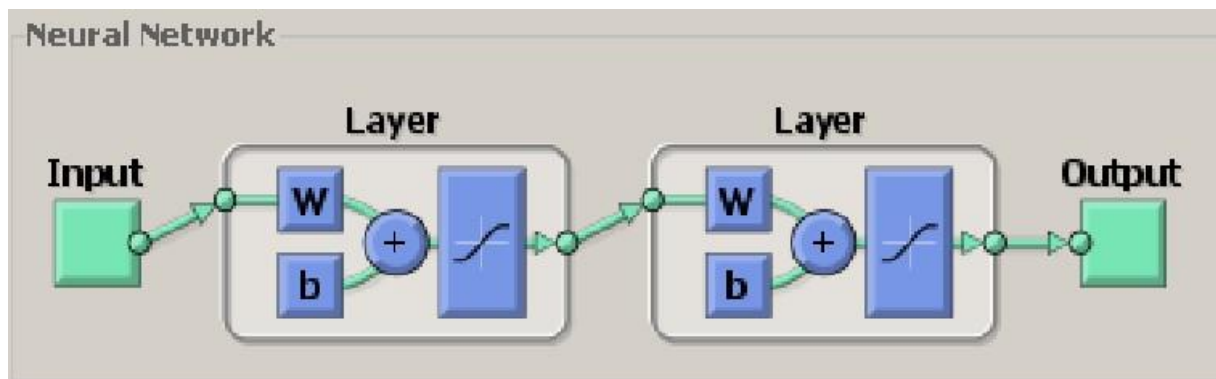


Рисунок 3.5 - структура багатозарового персептрона

На вхід мережі подаються дані, які зазнали передобробці і операції згортки в частотній області з фільтрами Габора, про які піде мова в наступному підрозділі.

3.4 Фільтри Габора

Фільтр Габора - лінійний електронний фільтр, імпульсна перехідна характеристика якого визначається у вигляді гармонійної функції, помноженої на Гауссіан. При цифровій обробці зображень цей фільтр застосовується для розпізнавання меж об'єктів [26].

Через властивості відповідності згортки в частотній області множенню у часовій області, перетворення Фур'є імпульсної передавальної характеристики фільтра Габора є сверткою перетворень Фур'є гармонійної функції і гауссіана:

$$g(x, y; \lambda, \theta, \varphi, \sigma, \gamma) = e^{\frac{-x^2 + y^2 \gamma^2}{2\sigma^2}} \cdot \cos\left(2\pi \frac{x}{\lambda} + \varphi\right) \quad (3.4)$$

де $x' = x \cdot \cos \theta + y \cdot \sin \theta$ та $y' = -x \cdot \sin \theta + y \cdot \cos \theta$, λ - довжина хвилі множника косинуса, θ - визначає орієнтацію нормалі паралельних смуг функції Габора в градусах, φ - зсув фаз в градусах, γ - коефіцієнт стиснення[14].

Для побудови двовимірного фільтра Габора застосовується формула (13):

$$G(x, y) = e^{\frac{-1}{2} \left(\frac{x_\varphi^2}{\sigma_x^2} + \frac{y_\varphi^2}{\sigma_y^2} \right)} \cdot \cos(2\pi \theta x_\varphi) \quad (3.5)$$

де $x_\varphi = x \cdot \cos(\varphi) + y \cdot \sin(\varphi)$ та $y_\varphi = -x \cdot \sin(\varphi) + y \cdot \cos(\varphi)$, σ_x, σ_y - стандартні відхилення гаусового ядра, по осях x і y , що визначають розтягнутість фільтра по осях, θ - частотна модуляція фільтра, φ - просторова спрямованість фільтра, що визначає його орієнтацію щодо головних осей.

Фільтр Габора ефективний для обробки зображень із структурною надмірністю [30]. До них відносяться дактилоскопічні зображення, кристаллограмми і знімки поверхонь металів. Оскільки, в цілому, розпізнавання дефектів не є складним завданням для людини, вибір біологічно подібних із зоровою корою людини фільтрів Габора, добре підходить для вирішення цієї проблеми.

Загалом, ядра вейвлетів Габора є самоподібними, так як вони можуть бути виведені з одного фільтра - материнського вейвлета, за допомогою масштабування і обертання за допомогою хвильового вектора $k_{\mu, \nu}$. Кожне ядро - добуток гауссової згортки і комплексної хвильової площини, тоді як перший терм в квадратних дужках в (9) визначає коливальну частина ядра, а другий терм компенсує значення постійного струму. Параметр сигма - стандартне відхилення ширини гаусової згортки від довжини хвилі. У більшості випадків, дослідники використовують вейвлети Габора з п'ятьма різними масштабами, $\nu \in \{0, \dots, 4\}$, і вісьмома орієнтаціями, $\mu \in \{0, \dots, 7\}$ [16-18]. На рисунку 3.6 показана реальна

частина ядер Габора на 5 масштабних рівнях і в 8 напрямках, а також їх величини для наступних параметрів:

$$\sigma = 2\pi, k_{max} = \frac{\pi}{2}, f = \sqrt{2}.$$

Обробка зображення фільтром Габора досягається шляхом усереднення значень оброблюваного зображення за деякою області в кожній точці. Відповідно, накладення фільтра Габора на зображення має вигляд (16):

$$I'(x, y) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n I\left(x - \frac{n}{2} + i, y - \frac{n}{2} + j\right) \cdot G(i, j), \quad (3.5)$$

де $I(x, y)$ - інтенсивність вихідного зображення в точці (x, y) , $I'(x, y)$ - інтенсивність нового зображення в точці (x, y) , $G(i, j)$ - значення функції Габора $i \in [0, n], j \in [0, n]$

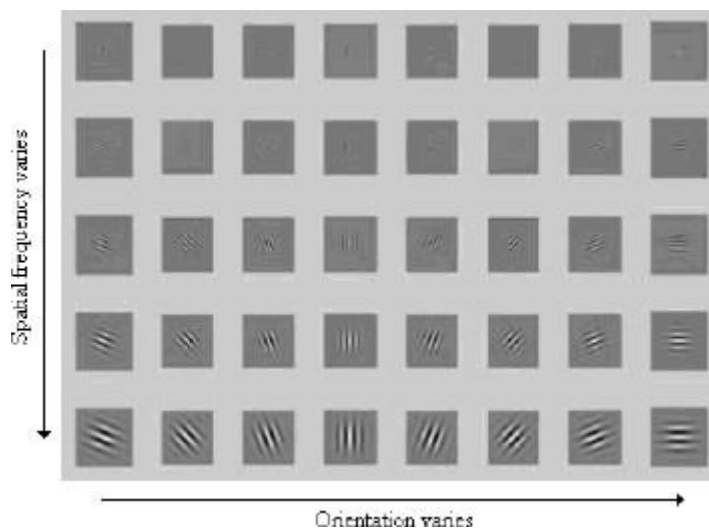


Рисунок 3.6 - Представлення вейвлетів Габора

Подання вейвлетів Габора для зображення - це згортка зображення з сімейством ядер Габора, використовуючи рівняння (8). Нехай $I(z)$ - розподіл рівнів сірого зображення. Результат конволюції зображення I і $\Psi_{\mu, \nu}(z)$ визначається як (3.6):

$$O_{\mu, \nu} = I(z) * \Psi_{\mu, \nu}(z) \quad (3.6)$$

де $z = (x, y)$ і $*$ позначає оператор конволюції (згортки). Застосовуючи теорему згортки, швидке перетворення Фур'є (ШПФ) використовується для отримання результату операції згортки. Рівняння (18) - визначення конволюції через ШПФ.

$$I\{O_{\mu,v}(z)\}I\{I(z)\}I\{\Psi_{\mu,v}(z)\} \quad (3.7)$$

$$O_{\mu,v}(z) = I^{-1}\{I\{I(z)\}I\{\Psi_{\mu,v}(z)\}\} \quad (3.8)$$

де I і I^{-1} позначають перетворення Фур'є і зворотне перетворення Фур'є відповідно.

Приклад результату згортки зображення з фільтрами Габора представлений на малюнку 3.7. Вихідні значення згорток виявляють здібності просторової орієнтації і селективності. Такі характеристики роблять стійкі локальні особливості, які підходять для візуального розпізнавання. Надалі, ми позначимо $O_{\mu,v}(z)$ величину виходів операції згортки.

3.5 Архітектура згорткової нейронної мережі

В роботі була реалізована і використана модель згорткової нейронної мережі з наступною архітектурою.

Що містить 9 шарів (таблиця 2.1), з яких один згортковий і один субдискретизуючий шар («проста» архітектура). Фільтри першого загорткового шару самонавчання. Навчання мережі вироблялося протягом 400, 1000 і 10000 епох.

Що містить 9 шарів (таблиця 2.1), з яких один сверточних і один субдискретизуючий шар («проста» архітектура). Фільтри першого сверточного шару - фільтри Габора. Навчання мережі вироблялося протягом 400, 1000 і 10000 епох.

Що містить 14 шарів (таблиця 2.2), з яких два згортальних шару і два субдискретизуючих шару («ускладнена» архітектура). Фільтри першого сверточного шару самонавчання. Навчання мережі вироблялося протягом 400, 1000 і 10000 епох.

«Складна» архітектура - за основу взята мережа AlexNet

Розглянемо докладніше характеристики шарів сверточное мережі, наведених в таблиці 3.1.

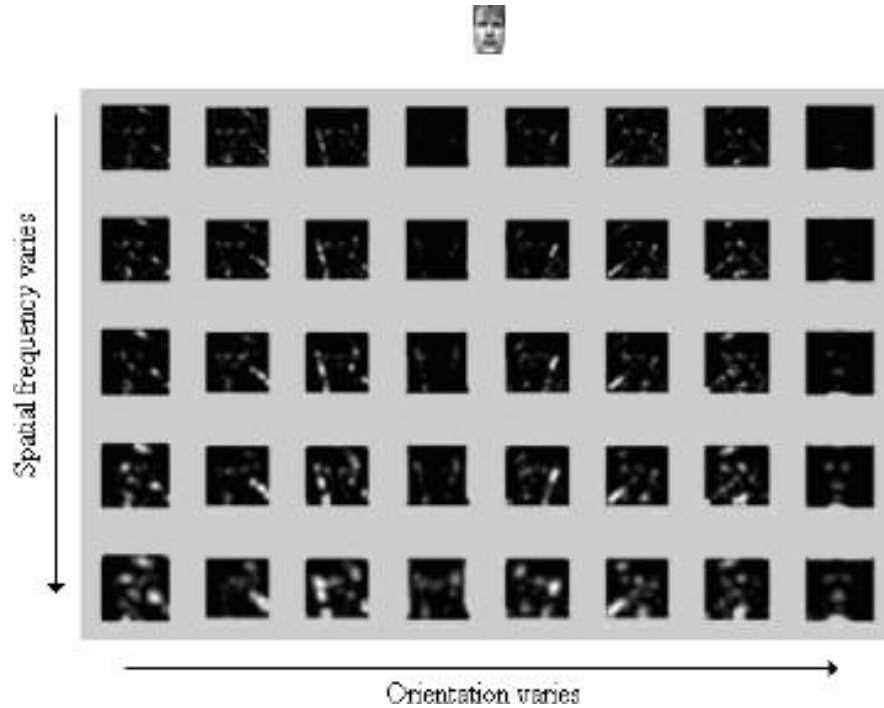


Рисунок 3.7 - Одне із зображень вибірки і 40 виходів операції згортки

Перший шар використаної нейронної мережі - Image Input. Йому на вхід подається зображення розміром 32×32 пікселя, про якого йдеться функція, що виконує попередню обробку зображення.

Другий шар використаної нейронної мережі - сверточних. Йому на вхід подається зображення розміром 32×32 пікселя, про якого йдеться в згортка розміром 32×32 . Кількість використаних фільтрів - 40. В якості фільтрів даного шару були використані фільтри Габора в одному з варіантів мережі з «простий» архітектурою. Розмір вихідних карт ознак даного шару складає 32×32 пікселів. Третій шар - ReLU. Завдання третього шару - застосування нелінійної функції. Четвертий шар - шар Субдискретизація. У даній архітектурі мережі використовується MaxPool шар, вибір здійснюється між 3-ма сусідніми пікселями.

П'ятий шар розглянутої нейронної мережі - повнозв'язну шар, що містить 100 нейронів. Як функції активації даного шару, для класифікації більш ніж на два класи, була обрана softmax-функція виду (3.9):

$$P(c_\tau \vee x, \theta) = \frac{P(x, \theta \vee c_\tau)P(c_\tau)}{\sum_{j=1}^k P(x, \theta \vee c_j)P(c_j)} = \frac{e^{a_\tau(x, \theta)}}{\sum_{j=1}^k e^{a_\tau(x, \theta)}} \quad (3.9)$$

де $0 \leq P(c_\tau \vee x, \theta) \leq 1, \sum_{j=1}^k P(c_j \vee x, \theta) = 1, a_\tau = \ln(P(x, \theta \vee c_\tau)P(c_\tau)), P(x, \theta \vee c_\tau)$ - умовна ймовірність даного класу τ , а $P(c_\tau)$ - апіорна ймовірність класу.

Таблиця 3.1 - Шари CNN з «простою» архітектурою

№ сло я	Название слоя в программе	Тип слоя	Характеристики
1	'imageinput'	Image Input	32x32x1 images with 'zerocenter' normalization
2	'conv'	Convolution	40 32x32x1 convolutions with stride [1 1] and padding [0 0]
3	'relu'	ReLU	ReLU
4	'maxpool'	Max Pooling	3x1 max pooling with stride [2 2] and padding [1 0]
5	'fc'	Fully Connected	100 fully connected layer
6	'softmax'	Softmax	Softmax
7	'fc_1'	Fully Connected	3 fully connected layer
8	'softmax_1'	Softmax	Softmax
9	'classoutput'	Classification Output	cross-entropy with 'flaw', 'not-defect', and 1 other classes

Функція softmax також відома як нормалізована експонента і може розглядатися як мультикласове узагальнення для логістичної сігмоїдної функції [30].

Шостий шар - Softmax.

Сьомий шар - повнозв'язну шар, що містить 3 нейрона.

Восьмий шар - Softmax.

Дев'ятий шар - повнозв'язний шар класифікації. Для завдань багатокласової класифікації MATLAB присвоює кожному входу один з k взаємовиключних класів [29]. Функція втрат (помилки) для цього випадку є функцією взаємної ентропії для схеми кодування 1 до k [30] і обчислюється за формулою (10):

$$E(\theta) = - \sum_{i=1}^n \sum_{j=1}^k t_{ij} \ln y_i(x_i, \theta), \quad (3.10)$$

де θ - вектор параметрів, t_{ij} - індикатор того, що i -ий зразок належить j -му класу, $y_i(x_i, \theta)$ - вихідний вектор для i -го зразка. Вихід $y_i(x_i, \theta)$ може бути інтерпретований як ймовірність того, що нейронна мережа асоціює i -ий вхід з j -им класом, тобто $P(t_j = 1 \vee x_j)$.

Функція активації вихідного шару - softmax (21):

$$y_\tau(x) = \frac{e^{\alpha_\tau(x)}}{\sum_{j=1}^k e^{\alpha_j(x)}} \quad (3.11)$$

де $0 \leq y_\tau \leq 1$, $\sum_{j=1}^k y_j = 1$.

Для кожного елемента x вхідний вибірки MATLAB нормалізує x , використовуючи формулу (3.11):

$$x = \frac{x}{\left(K + \frac{\alpha + ss}{windowChannelSize}\right)^\beta} \quad (3.11)$$

де K , α та β - гіперпараметри, ss - сума квадратів елементів у вікні нормалізації. Ця формула трохи відрізняється від представленої в [30]. Ми можемо отримати аналогічну формулу, помноживши значення α на $windowChannelSize$.

Структура мережі AlexNet приводиться на рис 3.8. Мережа AlexNet містить всі розглянуті раніше види шарів, але в набагато більшій кількості.

Pretrained AlexNet convolutional neural network architect:

25x1 Layer array with layers:|

1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0]
6	'conv2'	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2]
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0]
10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1]
11	'relu3'	ReLU	ReLU
12	'conv4'	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1]
13	'relu4'	ReLU	ReLU
14	'conv5'	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1]
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50% dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50% dropout
23	'fc8'	Fully Connected	3 fully connected layer
24	'prob'	Softmax	softmax
25	'output'	Classification Output	crossentropyex with 'flaw', 'pore', and 'not-defect' classes

Рисунок 3.8 – Архітектура модифікованої мережі AlexNet

3.6 Математична модель згорткової нейронної мережі

Для опису математичної моделі даної нейронної мережі будемо використовувати такі позначення.

Під $l \in [1; L]$ будемо розуміти розглянутий в даний момент шар нейронної мережі, де $L = 2\alpha + 2, \alpha \in \mathbb{Z}$ - кількість шарів в мережі. За N^l позначимо кількість карт ознак на шарі l , а за f_l - функцію активації розглянутого шару l . Також під змінної y_n^l будемо розуміти n -у карту ознак на шарі l [7].

Введемо в розгляд згортковий шар l . У подібній архітектурі нейронної мережі l приймається непарним числом, тобто $l = 1, 3, \dots, 2\alpha + 1$. Тоді для карти ознак n матиме місце наступне:

$w_{mn}^l = \{w_{mn}^l(i, j)\}$ - згортка, що застосовується до карти ознак m шару $(l-1)$ на шарі l з картою ознак n ;

b_n^l - порогові значення, що приєднуються до карти ознак n на шарі l ;

V_n^l - список всіх рівнів шару $(l-1)$, які з'єднуються з картою ознак n шару l .

Таким чином, карта ознак n згорткового шару l буде обчислюватися так:

$$y_n^l = f_l \left(\sum_{m \in V_n^l} y_m^{l-1} \otimes w_{mn}^l + b_n^l \right), \quad (3.12)$$

де під оператором \otimes розуміється математична операція двовимірної згортки.

Припустимо, що розмір вхідних карт ознак y_m^{l-1} дорівнює $H^{l-1} \times W^{l-1}$, а розмір застосовуваної до них згортки w_{mn}^l дорівнює $r^l \times c^l$, тоді розмір вихідний карти ознак y_n^l обчислюється як див. Рисунок 3.8

Введемо в розгляд субдискретизуючий шар l . У згорткової нейронної мережі l прийнято приймати парним числом, тобто $l = 2, 4, \dots, 2\alpha$. Для карти ознак n введемо такі позначення: w_{mn}^l - фільтр, який застосовується до n на шарі l , і b_n^l - додаткове порогове значення.

Далі будемо діяти наступним чином: розділимо карту ознак n $(l-1)$ -ого шару на непересічні блоки розміром 2×2 пікселя. Потім підсумуємо значення чотирьох пікселів в кожному блоці і в результаті отримаємо матрицю $z_n^{l-1} = \{z_n^{l-1}(i, j)\}$, елементами якої будуть відповідні значення сум. Таким чином, формула для обчислення значень елементів матриці матиме такий вигляд:

$$z_n^{l-1} = y_n^{l-1}(2l-1, 2j-1) + y_n^{l-1}(2l-1, 2j) + y_n^{l-1}(2l, 2j-1) + y_n^{l-1}(2l, 2j)$$

Карта ознак n субдискретизуючого шару l обчислюється як (25):

$$y_n^l = f_l(z_n^{l-1} \times w_{m,n}^l + b_n^l) \quad (3.13)$$

Завдяки представленим вище міркуванням стає можливим порахувати розмір $H^l \times W^l$ карти ознак y_n^l субдискретизуючого шару l (див. рисунок 3.9 і формулу (3.14)):

$$H = \frac{H^{l-1}}{2}, W = \frac{W^{l-1}}{2} \quad (3.14)$$

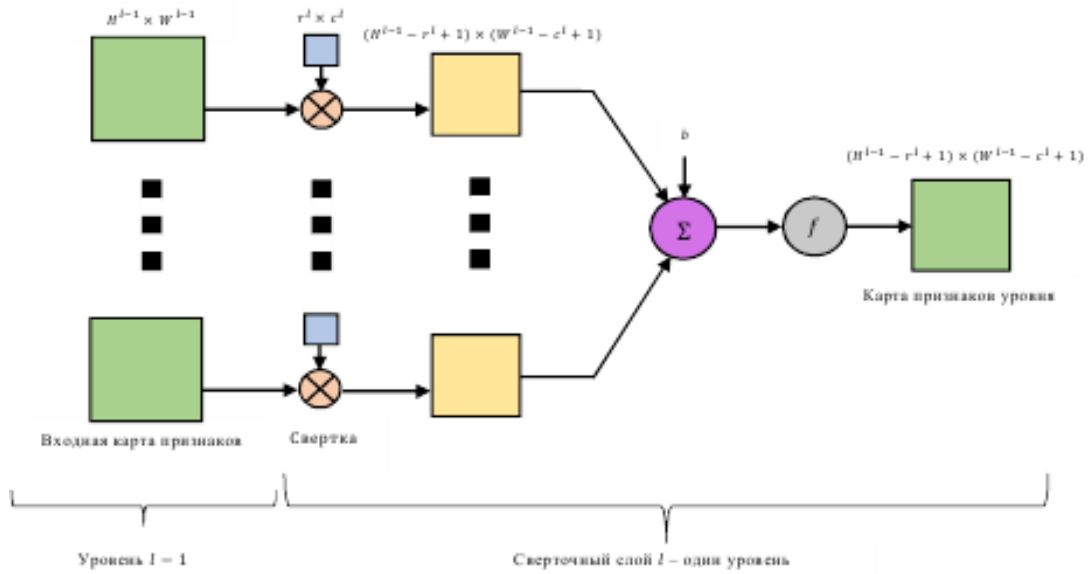


Рисунок 3.9 - Схема згорткового шару l

Введемо в розгляд вихідний шар L, що складається з одиничних нейронів. Прийmemo за N^L - кількість нейронів на даному шарі. Як і при розгляді попередніх шарів, позначимо за $w_{m,n}^L$ - фільтр, який застосовується до карті ознак m останнього сверточного шару для отримання переходу до нейрона n вихідного шару. Нехай b_n^L - порогове значення, яке додається до нейрона n .

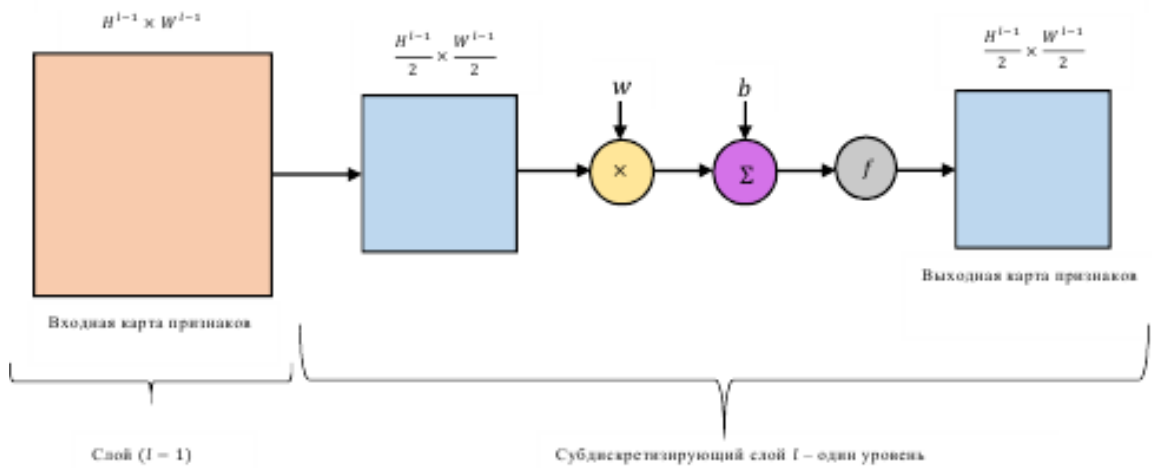


Рисунок 3.10 – Схема субдискретизирующего шару l

Таким чином, виходом згорткової нейронної мережі є вектор такого вигляду: $y = \{y_1^L, y_2^L, \dots, y_{N^L}^L\}$.

3.7 Навчання загорткової нейронної мережі

Основною метою навчання більшості нейронних мереж, призначених для вирішення завдань розпізнавання є мінімізація функціоналу помилки в перебігу декількох епох [32]. Як і класичні архітектури нейронних мереж, згорткові нейронні мережі «навчаються» за допомогою методу зворотного поширення помилки, модифікації стохастичного градієнтного спуску.

Нехай навчальна множина містить P зображень. Позначимо як X^P p -е зображення з навчальної множини, а за T_n^P - бажаний p -ий вихід системи. Тоді функціонал помилки визначається як (3.15):

$$E(w) = -T_n^P \cdot \log Y_n^P \quad (3.15)$$

де Y_n^P - вихід нейронної мережі.

Для мінімізації розглянутого вище функціоналу помилки (28) необхідно обчислити його градієнт, для цього буде потрібно знайти приватні похідні по основним параметрам мережі.

Для кожного нейрона (i, j) карти ознак n шару l , $l = 1, \dots, L$, приватна похідна по зваженої сумі $s^l, P_{n(i,j)}(i, j)$, де $n = 1, \dots, N^l$, $l = 1, \dots, L$, $j = 1, \dots, H^l$, має вигляд (3.16):

$$\delta_{n,p}^l(i, j) = \frac{\partial E}{\partial s_{n,p}^l(i, j)} \quad (3.16)$$

Можна висловити похідну (2) через функцію активації шару. Для вихідного шару L (2) буде мати вигляд (30):

$$\delta_{n,p}^L(i, j) = \frac{1}{P \times N^L} \cdot e_n^p \cdot f'(s_{n,p}^L) \quad (3.17)$$

де $e_n^p = Y_n^{L,p} - T_n^p$

Для згорткового шару $l = 2$ а $l = 1$ (2) буде мати вигляд (3.18):

$$\delta_{n,p}^l(i, j) = f_i'[s_{n,p}^l(i, j)] \times \delta_{n',p'}^{l+1} \times w_{m,n}^{l+1,p}, \quad (3.18)$$

Використовуючи обчислені похідні (3), (4), (5), можна обчислити градієнт функції помилки мережі (1) для кожного шару за параметрами $w_{m,n}^l$ і b_n^l

Градієнт для вихідного шару L (3,18):

$$\frac{\partial E}{\partial w_{m,n}^l} = \sum_{p=1}^P \delta_n^l \cdot p_{m-1}^l \cdot p \cdot \frac{\partial E}{\partial b_n^l} - \sum_{p=1}^P \delta_n^l \cdot p \quad (3.18)$$

де $m = 1, 2, \dots, N^{l-1}$, $n = 1, 2, \dots, N^l$.

Градiєнт для згорткового шару $l = 2a + 1$ (3.19):

$$\frac{\partial E}{\partial w_{m,n}^l} = \sum_{p=1}^P \delta_n^l \cdot p(i, j) \cdot y_m^{l-1} \cdot p \cdot \frac{\partial E}{\partial b_n^l} - \sum_{p=1}^P \sum \delta_n^l \cdot p(i, j) \quad (3.19)$$

де $m = 1, 2, \dots, N^{l-1}$, $n = 1, 2, \dots, N^l$

В основі методу зворотного поширення помилки лежить стохастический градиєнтний спуск: настройка параметрів мережі для досягнення мінімуму функції помилки проводиться після одного випадково обраного навчального зразка, пред'явленого мережі і мінімізація здійснюється в напрямку антиградиєнта. Таким чином, оновлення ваг здійснюється відразу після обчислення градиєнта для поточного елемента навчальної вибірки (3.20).

$$\Delta w = -\alpha \cdot \nabla E^p \quad (3.20)$$

де $\alpha > 0$ - швидкість навчання мережі.

На першому кроці задаємо довільні значення вагових коефіцієнтів. Потім передаємо сигнал в мережі в прямому напрямку. На цьому ж етапі обчислюємо значення функції помилки для вихідного шару.

Другим кроком навчання мережі є поширення помилки, отриманої на попередньому етапі, від вихідного шару до попередніх. При цьому відбувається обчислення нових значень ваг і порогів для кожного входу нейрона з використанням градиєнтного спуску (3.21):

$$w' = w + \alpha \cdot \frac{\partial E}{\partial w}; \quad b' = b + \alpha \cdot \frac{\partial E}{\partial b} \quad (3.21)$$

де $\frac{\partial E}{\partial w}$, $\frac{\partial E}{\partial b}$ розраховуються за формулами (33), (34), (35) і (36).

Таким чином, при побудові згорткової нейронної мережі в даній роботі, стояла мета - мінімізувати функцію втрат $E(w)$ за рахунок вибору параметрів, що настроюються навченою системи w . $E(w)$ являє собою арифметичне середнє функцій виду (3.22):

$$E^p = T(T^p, F(w, Z^p)) \quad (3.22)$$

де Z^p - p -ий вхід системи, T^p - невідповідність між бажаним виходом системи для Z^p і значенням, яке видає нейронна мережа.

При навчанні згорткової нейронної мережі в MATLAB можна налаштувати багато параметрів процесу навчання. Розглянемо ці параметри більш докладно. Перший і найважливіший параметр - метод навчання. В якості методу навчання прийнято використовувати метод стохастичного градієнтного спуску з імпульсом (Stochastic Gradient Descent with Momentum, SGDM). Необхідно вказати 'sgdm' як SolverName у виклику trainingOptions. SGDM оновлює ваги і зміщення, роблячи невеликі кроки в напрямку антиградієнта функції втрат, щоб мінімізувати втрати. Він оновлює параметри за допомогою підмножини даних на кожній ітерації. Ці підмножини називаються міні пакетом (mini-batch). Розмір mini-batch можна вказати, використовуючи параметр MiniBatchSize в trainingOptions. Чим більше розмір mini-batch, тим швидше відбувається навчання, але нижче якість. За замовчуванням в MATLAB значення mini-batch одно 128. Для кращої якості навчання воно було зменшено до 16.

Кожна оцінка градієнта з використанням mini-batch називається ітерацією. А повний прохід по всьому набору даних називається епохою навчання. Максимальна кількість епох навчання також є параметром, який можна налаштувати. Оптимальним виявилось кількість епох навчання рівне 400 для ускладненою структури сверточное мережі і 1000 для простої структури сверточное мережі[27].

Параметр імпульсу - це спосіб запобігти коливання алгоритму стохастичного градієнтного спуску уздовж шляху найшвидшого спуску. Як правило, значення за замовчуванням для цього параметра дорівнює 0.9.

Можна вказати глобальну швидкість навчання (global learning rate) за допомогою аргументу 'InitialLearnRate' в trainingOptions. Значення за замовчуванням цього параметра дорівнює 0.1, але можна вибрати і менше значення. За замовчуванням, trainNetwork використовує це значення протягом усього процесу навчання, однак можна змінити його, примножуючи його через певного кількість епох на деякий коефіцієнт. Замість того, щоб використовувати

невелику фіксовану швидкість навчання протягом усього навчання, можна вибрати більш високу швидкість навчання на початку навчання і поступово зменшувати це значення під час оптимізації, що може допомогти скоротити час навчання, однак більш дрібні кроки до оптимального значення в міру проходження навчання дадуть кращу якість в кінці навчання. Якщо ми хочемо, зменшують швидкість навчання, варто використовувати аргумент 'LearnRateSchedule' зі значенням 'piecewise'. Після вибору цієї опції trainNetwork за замовчуванням примножує початкову швидкість навчання на 0.1 кожні 10 епох. Проте, можна вказати коефіцієнт, за допомогою якого можна зменшити початкову швидкість навчання і кількість епох, через яке відбувається множення за допомогою параметрів 'LearnRateDropFactor' і 'LearnRateDropPeriod'.

3.8 Опис бази зображень

Для тестування розроблених алгоритмів розпізнавання була використана згенерована база даних, що містить зображення дефектів і поверхонь без дефектів, а також відповідні їм мітки. Для навчання класифікаторів було використано 1500 зображень, з яких 900 зображень використовувалися в якості навчальної вибірки, 600 зображень використовувалися в якості тестової вибірки. Розмір кожного зображення становить 32×32 пікселя. Кожен піксель зображення кодується числом в інтервалі $[0; 255]$, де 0 відповідає чорному кольору, а 255 - білому.

Попередня обробка зображень необхідна для досягнення максимальної точності розпізнавання системи. З урахуванням предобработки вхідних зображень нейронна мережа буде інваріантна до незначних спотворень, таким як шуми, поворот і масштабування. Предобработка зображення здійснювалася в 3 етапи.

- Попередня обробка: переклад зображення в відтінки сірого, згладжування зображення за допомогою медіанного фільтру і видалення шуму.

- Детекція ймовірних місць розташування дефектів.

- Додаткова обробка подокна зображення: адаптивна еквалізація гістограми, нормалізація даних. Перед подачею зображення на Feed-Forward MLP також додатково проводилася згортка зображення з фільтрами Габора.

Так як зручніше працювати з напівтоновим зображенням (менше розмірність, менший обсяг вхідних даних), то вхідне зображення (якщо воно RGB) перетворюється в відтінки сірого за допомогою функції `rgb2gray (I)`, де I - вихідне зображення.

Для згладжування зображення використовувався медіанний фільтр представлений в пакеті Image Processing Toolbox функцією `medfilt2 (I)`, де I - вихідне зображення. Також функція `medfilt2` є ефективним засобом фільтрації шуму типу «salt and pepper». Для даного класу шуму ця нелінійна фільтрація зазвичай дає кращі результати, ніж фільтрація на основі операції згортки. Медіанна фільтрація є окремим випадком рангової фільтрації, описаної нижче.

Як маски фільтра використовується матриця розміру 3×3 (якщо вхідне зображення не задовольняє умовам застосування маски, для того, щоб розміри вхідного і вихідного зображень були однакові, вхідне зображення тимчасово доповнюється необхідною кількістю рядків і стовпів нульових пікселів). Пікселі вихідного зображення у відтінках сірого, які відповідають усім елементам маски фільтра становлять упорядковану послідовність A . пікселів $D(r, c)$, де r і c - координати поточного положення центрального елемента маски, присвоюється значення медіани послідовності A . Операція застосовується нерекурсивно для всіх положень маски [11].

Перевіряти кожного подокно (розміру 32×32) вхідного зображення (розмір якого може дуже великим) з метою класифікації за допомогою нейронної мережі вкрай неефективно. Тому хотілося б вгадати місце розташування дефекту, щоб не перевіряти кожне можливе місце. Після деякого тестування, ми прийшли до висновку, що для ефективної детекції можливих місць розташування дефектів можна виміряти міру схожості (ступеня кореляції) двох зображень: дефект-подібного способу (заздалегідь відомого зображення) і вхідного зображення. Надалі будемо називати дефектоподібний образ шаблонним (типовим) дефектом.

В результаті було використано два шаблонних дефекту: один з яскравим фоном, другий з темним (чорним) фоном.

Для простоти будемо вважати взаємну кореляцію сверткой, але без повороту ядра. Перед проведенням згортки зображення і шаблонні дефекти необхідно нормалізувати (про процес нормалізації буде розказано далі). У matlab для обчислення згортки двовимірних масивів X і Y використовується функція $\text{conv2}(X, Y)$, яка проводить обчислення за такою формулою (3.23):

$$C(j, k) = \sum_p \sum_q X(p, q)Y(j - p + 1, k - q + 1) \quad (3.23)$$

де p і q пробігають всі значення, які призводять до коректним зверненнями за індексами для $X(p, q)$ та $Y(j - p + 1, k - q + 1)$.

Після того, як ми провели операцію згортки для обох шаблонних дефектів із зображенням, використовуємо функцію $\text{imregionalmax}(I)$, де I - вихідне зображення, яка визначає локальні максимуми масиву I . Результат роботи функції imregionalmax - бінарне зображення, що збігається за розміром з вихідним, на якому більшість пікселів - нулі, і тільки кілька - одиниці, які відповідають розташуванню локальних максимумів. Локальні максимуми є зв'язковими компонентами пікселів з однаковими значеннями інтенсивності t , чий зовнішні пікселі мають значення менше ніж t . Потім зображення з локальними максимумами для обох результатів згортки з шаблонними дефектами об'єднуємо в одне (щоб бути впевненими, що ми не втратили ніяких важливих точок).

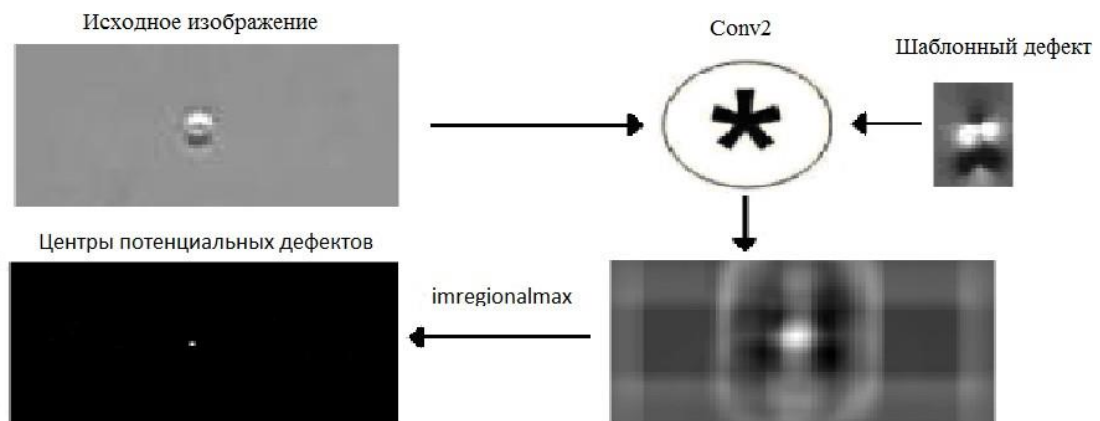


Рисунок 3.11 - Алгоритм виявлення можливих місць розташування дефектів

В цілому процес детекції можливих місць розташування дефектів наведено на рис. 3.11.

Для кращої якості роботи алгоритмів все вхідні дані необхідно нормалізувати. Найбільш популярний діапазон для нормалізації $[-1; 1]$, але можна використовувати і будь-який інший діапазон.

Однак бажано, щоб вибраний діапазон був симетричний щодо нуля. Для нормалізації скористаємося такою формулою (3.24):

$$output = 2\left(\frac{input - min}{max - min} - 0.5\right) \quad (3.24)$$

де $max = \max(\max(input))$, $min = \min(\min(input))$

Ми можемо мати за фактом одне і те ж зображення, але з різною контрастністю (С) і яскравістю (В). Значення кожного пікселя зображення розраховується за формулою (3.25):

$$pixelvalue = C \cdot pixel\ valueold + B \quad (3.25)$$

де $pixelvalue$ - поточне значення пікселя зображення, C - контрастність, B - яскравість, $pixelvalueold$ - значення пікселя зображення до накладення певної яскравості і контрастності. У простому випадку C і B фіксовані для всіх пікселів подокна зображення. Чим більше яскравість зображення, тим більші значення приймають пікселі в напівтоновому зображенні. Тому змінюючи лінійно значення рівня сірого пікселя, ми отримуємо те ж зображення, той же дефект, але темніше або яскравіше вихідного. Погано те, що нейронна мережа буде бачити два різних зображення, хоча людський зір і мозок бачать одне й те саме зображення, але з різною яскравістю.

Гістограма - це дискретна функція $f(x)$, являє собою

розподіл яскравості пікселів зображення і задається за формулою (3.26):

$$f(x) = \frac{n_i}{n} \quad (3.26)$$

де n_i - число пікселів з яскравістю i ($i = 0 \dots 255$), n - загальна кількість пікселів.

Зміна контрастності і яскравості насправді схоже на зміну освітленості. Але якщо освітленість змінюється фізично (змінюється джерело світла і його

інтенсивність), то це відбувається швидше за все нелінійно, і освітленість для різних частин зображення буде різною (рисунок 3.3). Є два шляхи вирішення цієї проблеми. Перший шлях: моделювання фонового освітлення зображення. В цьому випадку усереднюється освітленість всього зображення або його подокна чи застосовуються фільтри на зразок фільтра Гаусса (якщо ми віднімемо відповідь фільтра для зображення з зображення, то отримаємо зображення без впливу фонового освітлення).

3.9 Метрики аналізу моделей і результати окремих моделей

У разі, коли стоїть завдання класифікації на 3 і більше класів, то природним способом вимірювання якості класифікації є матриця помилок. Дана метрика характеризує здатність класифікатора вірно визначати клас для об'єктів, по ній також легко зрозуміти, які класи дискримінуються і на чию користь, таке можливо, наприклад, при незбалансованості вибірки. Метрика є матрицею, по рядках якої розташовуються справжні класи об'єктів, а по стовпцях то, що було передбачене класифікатором. В результаті виходить точна картина того, як класифікатор розділяє об'єкти, скільки об'єктів було класифіковано вірно, а скільки невірно.

Для розрахунку метрик точність і AUC використовувалася частина тренувального датасета. В результаті точність розпізнавання на тестовому безлічі реалізованої і використаної в даній роботі моделі сверточное нейронної мережі склала 94,5% (рисунок 3.12 і таблиця 3.3).

В процесі навчання згортальних нейронних мереж в MATLAB можна відстежити, як змінювалися точність Mini-batch і величина втрат в залежності від ітерації і епохи навчання, а також побачити скільки часу займала та чи інша епоха навчання. Для CNN з простою архітектурою і самонавчання фільтрами, а також для CNN з ускладненою архітектурою дані у вигляді таблиць представлені в додатку 2. Як приклад невелика таблиця даних наводиться нижче, а також

графік залежності розміру помилки від кількості епох нейронної мережі для CNN з простою архітектурою (рисунок 3.12).

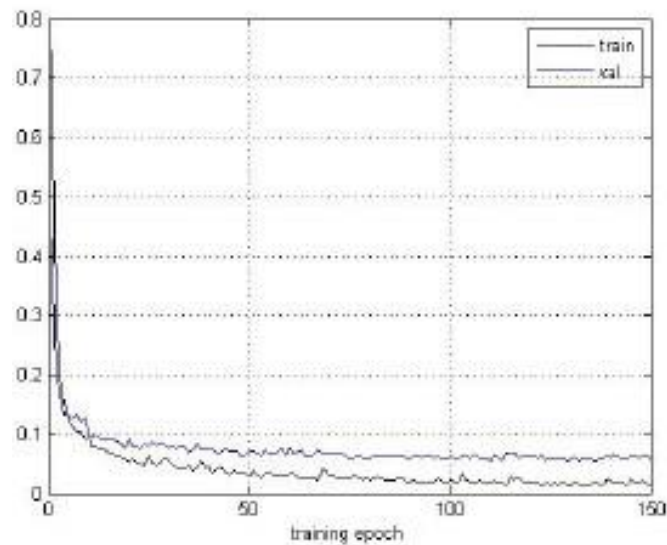


Рисунок 3.12 Графік залежності помилки від кількості епох нейронної мережі

Отримані матриці помилок для CNN з простою архітектурою (Рисунок 3.13 і Рисунок 3.14) і для CNN з ускладненою архітектурою (Рисунок 3.15).

Confusion Matrix

Output Class	1	400 58.5%	28 4.1%	0 0.0%	93.5% 6.5%
	2	0 0.0%	112 16.4%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	144 21.1%	100% 0.0%
		100% 0.0%	80.0% 20.0%	100% 0.0%	95.9% 4.1%
	1	2	3		Target Class

Рисунок 3.13: Матриця помилок для CNN з простою архітектурою, самонавчання фільтрами і 100 епохами навчання

Для згорткової нейронної мережі з простою архітектурою, самонавчання фільтрами і 20 епох навчання (мережа дообучалась, тому спочатку точність Mini-batch висока, а величина втрати - низька) нижче (див. таблицю 3.2) представлена інформація про процес навчання. Результати порівняння побудованих моделей за різними метрик якості наведені в таблиці 3.3.

Таблиця 3.2 - Процес навчання CNN з простою архітектурою протягом 20 епох

Епоха навчання	Ітерація	Минулий час (сек.)	Величина втрати Mini-batch	Точність Mini-batch	Темп навчання
2	50	3.46	0.7213	82.97%	0.001000
3	100	4.24	0.3253	89.62%	0.001000
4	150	4.99	0.2814	91.22%	0.001000
6	200	5.75	0.2318	91.22%	0.001000
7	250	6.54	0.2214	92.01%	0.001000
8	300	7.28	0.2111	92.76%	0.001000
10	350	8.04	0.2016	92.77%	0.001000
11	400	8.77	0.1511	92.94%	0.001000
12	450	9.52	0.1217	93.32%	0.001000
14	500	10.28	0.1219	93.35%	0.001000
15	550	11.06	0.0820	93.71%	0.001000
16	600	11.94	0.0422	94.23%	0.001000
18	650	12.71	0.0488	94.21%	0.001000
19	700	13.43	0.0423	94.22%	0.001000
20	750	14.63	0.0424	94.22%	0.001000

Output Class	1	451 30.1%	34 2.3%	0 0.0%	93.0% 7.0%
	2	28 1.9%	440 29.3%	0 0.0%	94.0% 6.0%
	3	21 1.4%	26 1.7%	500 33.3%	91.4% 8.6%
		90.2% 9.8%	88.0% 12.0%	100% 0.0%	92.7% 7.3%
		1	2	3	
		Target Class			

Рисунок 3.14 - Матриця помилок для CNN з простою архітектурою, фільтрами Габола для першого сверточного шару і 400 епохами навчання

За отриманою матриці помилок можна зробити висновок, що 90,2% дефектів першого виду (пора) були правильно класифіковані як пори. 88% дефектів другого виду (тріщини) були правильно класифіковані як тріщини. І все зображення, що не містять дефектів, були правильно класифіковані як що не містять дефектів. З усіх передбачених мережею пір 93% вірні і 7% пророкувань помилкові. З усіх передбачених мережею тріщин 94% вірні і 6% пророкувань помилкові. З усіх передбачених мережею відсутностей дефектів 91,4% вірні, 8,6% пророкувань помилкові. В цілому 92,7% пророкувань вірні і 7,3% помилкові.

Output Class	1	400 58.5%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	140 20.5%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	144 21.1%	100% 0.0%
		100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
		1	2	3	
		Target Class			

Рисунок 3.15 - Матриця помилок для CNN з ускладненою архітектурою і 400 епохами навчання

Таблиця 3.3 - Метрики якості для різних моделей

Метрика	CNN (проста архітектура, самонавчан ня фільтри для першого сверточного шару, 400 епох навчання)	CNN (Проста архітектура, фільтри Габора для першого сверточного шару, 400 епох навчання)	CNN (Ускладнена архітектура, самонавчанн я фільтри для першого згорткового шару, 400 епох навчання)	Feed- Forward MLP+ згортка зображенн я з фільтрами Габора	SVM з набором функцій
Точність (Accuracy)	0.9450	0.9100	0.9910	0.8930	0.8600
Час навчання (хв.)	34,68	10,54	46,01	4, 34	1,02
Час обробки одного дефекту (мс)	250	249	263	320	300

В даному розділі ми розробили методику попередньої обробки вхідної зображення для якісної роботи використовуваних класифікаторів, а також для ефективної детекції дефектів. Ми провели порівняльні тести розроблених моделей за кількома метриками якості, оцінили якість роботи досліджуваних алгоритмів розпізнавання і визначили середній час роботи програми для обробки одного дефекту і одного великого зображення з декількома дефектами.

ВИСНОВКИ

В першу чергу в роботі виконано аналіз сучасних методів розпізнавання зображень та створена класифікація типів існуючих підходів до розпізнавання візуальних зображень.

Таким чином можна зробити висновок, що існує багато методів та способів розпізнавання зображень за допомогою нейронних мереж. Розпізнавання візуальних зображень та їх обробка це важкий і кропіткий процес, але перспектива таких технологій дуже висока. Інструменти розпізнавання зображень мають в собі реалізації складних алгоритмів, що забезпечують швидку обробку пікселів зображень і виведення результатів. За допомогою різних інструментів є можливість реалізовувати методи розпізнавання візуальних зображень. Нами було розглянуто популярність та структуру нейронних мереж і їх можливості, а ще кожен крок обробки вхідних даних. Тому можна сказати, що розпізнавання візуальних зображень як напрямок комп'ютерних технологій розвивається дуже швидко і відкриває все нові можливості, які широко застосовуються в нашому повсякденному житті.

По-друге на основі аналізу і обробки спеціальної літератури та інтернет ресурсів були опрацьовані, виявлені і застосовані принципи і знання з теорії нейробіології, нейроінформатики, кібернетики, штучного інтелекту, штучних нейронних мереж і нейромережевої самоорганізації, перцептронів, кластеризації та категоризації об'єктів, аналізу сцен, машинного розпізнавання образів.

На підставі проведених досліджень можна впевнено сказати, що технологія розпізнавання візуальних зображень актуальна для компаній різного профілю. Сама технологія не є новою, але завдяки тому що зараз є можливість її технічної реалізації, вона набула такої популярності. В якості досліджуваної нейронної мережі, мною обрана згорткова нейронна мережа, її ефективність буде доведена в третьому розділі данної роботи.

По-третє у роботі розроблені і досліджені різні моделі для розпізнавання дефектів поверхонь металів на графічних зображеннях поверхонь.

Найкращу якість показав алгоритм на основі згорткових нейронних мереж. Особливістю даного алгоритму виявилось те, що він стійкий до зміни освітлення та кутів повороту.

Розроблено програму, що реалізує запропоновані алгоритми на мові MATLAB з використанням пакетів розширень Computer Vision System Toolbox, Image Processing Toolbox, Neural Network Toolbox і Parallel Computing Toolbox. Реалізована програма дозволяє оцінити якість запропонованих моделей, а також розпізнати дефекти на вхідному зображенні.

Розроблено методику експерименту, в ході якого було встановлено оцінки якості роботи алгоритмів. Алгоритми були протестовані на реальних зображеннях. Було встановлено, що згорткова нейронна мережа незалежно дає вірні відповіді приблизно в 95% випадків для «простої» архітектури і в 99,9% випадків для «ускладненої» архітектури, тоді як інші методи (наприклад, SVM, Feed Forward MLP) розпізнають 87% випадків, внаслідок нестійкості алгоритму детектування до певного виду шумів.

В перспективі можна використовувати навчені класифікатори при впровадженні на конкретних промислових підприємствах і реалізувати власні програми, які використовують навчені моделі алгоритмів класифікації і детектування.

ПЕРЕЛІК ПОСИЛАНЬ

1. Хайкин С. Нейронные сети: полный курс, 2-е издание: Пер.с англ. –М.: Изд.дом "Вильямс", 2006. – 1104 с.
2. Уоссерман Ф. Нейрокомпьютерная техника. Теория и практика. М: 2010. 184 с .
3. Осовский С. Нейронные сети для обработки информации /Пер. с польского И.Д. Рудинского. – М.: Финансы и статистика, 2012. – 344 с
4. Галушкин А.И. Нейронные сети: основы теории. / А.И. Галушкин - М.: Горячая линия - Телеком, 2010г. - 496 с.
5. Горбань А.Н. Обучение нейронных сетей / А.Н.Горбань - М.: СП Параграф, 2001.-256 с.
6. Журавлев Ю.И. Распознавание. Математические методы. Программная система. Практические применения. / Ю.И. Журавлев, В.В. Рязанов, О.В. Сенько - М.: ФАЗИС, 2006. - 159 с.
7. Букатов А.А., Дацюк В.Н., Жегуло А.И. Программирование многопроцессорных вычислительных систем / А.А. Букатов, В.Н. Дацюк, А.И. Жегуло. - Ростов-на-Дону: Изд-во ООО "ЦВВР", 2003. - 208 с.
8. Васильев Д. Знакомьтесь, Erlang / Д. Васильев // Системный администратор. - 2009. - №8
9. Воронцов, К. В., Лекции по искусственным нейронным сетям // К.В. Воронцов. - Воронеж, 2007. - 29 с.
10. Галушкин А.И. Нейронные сети: история развития: учеб. Пособие для вузов / А.И. Галушкин, Я.З. Цыпкин. - М.: ИПРЖ, 2001. - 839 с.
11. Герасименко М.С. Вычисление искусственных нейронных сетей на вычислительных кластерах или ЛВС / М.С. Герасименко / Вестник Воронежского государственного университета, Серия: Системный анализ и информационные технологии, №1, 2010. - С. 120-125
12. Королюк В.С. Справочник по теории вероятностей и математической статистике / В.С. Королюк, Н.И. Портенко, А.В. Скрятоход, А.Ф. Турбин. -

- М.: Наука, Главная редакция физико-математической литературы, 1985. - 640 с.
13. Красовский Г.И., Филаретов Г.Ф. Планирование эксперимента / Г.И. Красовский, Г.Ф. Филаретов. - Минск: изд-во БГУ, 2002. - 302 с.
 14. Крил П. Функциональное программирование - друг параллелизма / П. Крил // Открытые системы. - 2010. - №8
 15. Круглов В.В. Нечеткая логика и искусственные нейронные сети // В.В. Круглов, М.И. Дли, Р.Ю. Голунов: Учеб. пособие. - М.: Физматлит, 2011. - 224 с.
 16. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская и др. - М. Горячая линия - Телеком, 2004. - 452 с.
 17. Тархов Д.А. Нейронные сети. Модели и алгоритмы. Кн. 18 / Д.А. Тархов. - М.: Радиотехника, 2005. - 256 с.: ил.
 18. Чезарини Ф., Томпсон, С. Программирование в Erlang / Ф. Чезарини, С. Томпсон. - М.: ДМК Пресс, 2012. - 488 с.
 19. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities / J.J. Hopfield, Proceedings of National Academy of Sciences, vol. 79 no. 8, 1999. - pp. 2554-2558
 20. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators, Neural Netw., vol. 2, no. 5, pp. 359-366, 2009
 21. IBM BladeCenter HS22 Technical Introduction [Электронный ресурс]. - Режим доступа: <http://www.redbooks.ibm.com/redpapers/pdfs/redp4538.pdf>.
 22. Leshno M., Lin V.Y., Pinkus A., Schocken S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Netw., vol. 6, no. 6, pp. 861-867, 1993
 23. Форсайт Дэвид А. Компьютерное зрение. Современный подход? Дэвид А. Форсайт, Жан Понс - пер. с англ.. - М. : Издательский дом «Вильяме», 2004. - 928 с.
 24. Вахитов А.Т. Обзор алгоритмов стереозрения/ А.Т.Вахитов, Л.С.Гуревич, Д.В.Павленко. Стохастическая оптимизация в информатике, вып.4 / Под

- ред. О.Н.Граничина- СПб. : Из-во Санкт- Петербургского ун-та, 2008. - 299 с.
- 25.Вапник В.Н. Теория распознавания образов (Статистические проблемы обучения). М.: Наука, 1974. - 416 с.
- 26.Вороновский Г.К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. - Харьков: Основа, 2007. -112 с.
27. Гладков Л.А. Генетические алгоритмы / Л.А. гладков,В.В. Курейчик, В.М.Курейчик/Под ред. В.М,Курейчика.-2-е изд., испр. И доп.- М. :ФИЗМАТЛИТ,2006.-320с.
28. Журавлев Ю.И. Распознавание. Математические методы. Программная система. Практические применения. / Ю.И. Журавлев, В.В. Рязанов, О.В. Сенько - М.: ФАЗИС, 2006. - 159 с.
29. Коробкин А.А. Модели и методы искусственного интеллекта. Использование генетических алгоритмов т аппарата нечеткой логики для организации учебного процесса в ВУЗах./ И.Ф.Астахова, А.А.Коробкин-Berlin: LAP LAMBERTAcademic Publishing GmbH& CO. KG, 2012.-137 с.
- 30.Круглов В.В. Искусственные нейронные сети. Теория и практика. / В.В. Круглов, Борисов В.В. - М.: Горячая линия - Телеком, 2002г. - 382с.
- 31.Грузман И. С. Алгоритмы распознавания объектов, устойчивые к геометрическим искажениям: сдвигу, масштабу, повороту / И. С. Грузман, В. Г. Никитин // Автометрия. – 2004. – № 3. – С. 46-53.
- 32.Курченко О. V. Transformation and compression of images in optoelectronic logic and time environments / Kozhemiako V. P., Maidaniuk V. P., Kurychenko O. V. // Naukaistudia. – № 21 (89). – 2013. – P. 35–43. – ISSN 1561- 6894.