

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____Аліна САВЧЕНКО

“ _____ ” _____ 2021 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИЦІ ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТРА”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ
УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ”

**Тема: “Комп'ютерна система розпізнавання облич на основі
нейронних мереж”**

Виконавиця: Закалата Дарина Юріївна

Керівник: доцент Моденов Юрій Борисович

Нормоконтролер: _____ Ігор РАЙЧЕВ

Київ - 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12
“Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Аліна САВЧЕНКО

« ____ » _____ 2021р.

ЗАВДАННЯ

на виконання дипломної роботи студентки

Закалати Дарини Юріївни

(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Комп'ютерна система розпізнавання облич на основі нейронних мереж від 12.10.2021 за № 2228/ст.
- 2. Термін виконання роботи:** з 12.10.2021 по 31.12.2021.
- 3. Вихідні дані до роботи:** теоретичні відомості та основи теорії штучного інтелекту, методи та засоби розпізнавання образів та облич, алгоритми проектування моделей штучної нейронної мережі.
- 4. Зміст пояснювальної записки:** вступ, огляд теоретичної бази розпізнавання образів за допомогою штучного інтелекту, алгоритм нейромережевої організації матриць для розпізнавання образів, опис та технологія розроблення, використання створеної комп'ютерної системи розпізнавання облич, тестування системи та результати роботи, загальні висновки.
- 5. Перелік обов'язкового ілюстративного матеріалу:** ілюстрації у пояснювальній записці, презентація в MS PowerPoint.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання на дипломну роботу та побудова плану-графіку виконання робіт.	12.10.2021 – 15.10.2021	
2.	Огляд та аналіз відомих методів та засобів розпізнавання образів та облич.	16.10.2021 – 19.10.2021	
3.	Виявлення проблематики створення комп'ютерної системи розпізнавання облич.	20.10.2021 – 24.10.2021	
4.	Написання Розділу 1 та Розділу 2 дипломної роботи.	25.10.2021 – 31.10.2021	
5.	Побудова алгоритму нейромережевої організації матриць для розпізнавання облич.	01.11.2021 – 07.11.2021	
6.	Розробка функціональної блок-схеми комп'ютерної системи відповідно до алгоритму та програмна реалізація системи.	08.11.2021 – 17.11.2021	
7.	Написання Розділу 3 дипломної роботи. Завершення створення пояснювальної записки дипломної роботи.	18.11.2021 – 01.12.2021	
8.	Оформлення та друк пояснювальної записки.	02.12.2021 – 11.12.2021	
9.	Створення презентації, доповіді та підготовка до захисту дипломної роботи	12.12.2021 – 20.12.2021	

7. Дата видачі завдання: 12.10.2021р.

Керівник дипломної роботи _____
(підпис керівника)

Юрій МОДЕНОВ

Завдання прийняла до виконання _____
(підпис випускниці)

Дарина ЗАКАЛАТА

РЕФЕРАТ

Пояснювальна записка до дипломної роботи “Комп’ютерна система розпізнавання облич на основі нейронних мереж” складається із вступу, трьох розділів, загальних висновків, списку використаних джерел і містить 82 сторінки тексту, 30 рисунків та два додатки. Список використаних джерел містить 20 найменувань.

Метою дипломної роботи є дослідження процесу розпізнавання облич та образів з використанням нейронних мереж; розробка, проектування і удосконалення комп’ютерної системи розпізнавання облич на основі нейронних мереж.

Предметом дослідження методи розпізнавання обличчя за допомогою нейронних мереж.

Об’єктом дослідження є процес розпізнавання обличчя за допомогою моделі нейронної мережі – перцептрон.

Методами дослідження є програмне моделювання комп’ютерної системи, розробка алгоритму роботи системи, засоби та методи розпізнавання образів за допомогою штучних нейронних мереж та методів їх навчання.

Ключові слова: БІОМЕТРИЧНИЙ МЕТОД ІДЕНТИФІКАЦІЇ ЛЮДИНИ, РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, КОМП’ЮТЕРНА СИСТЕМА, НЕЙРОННА МЕРЕЖА, ПЕРСЕПТРОН, АЛГОРИТМІЧНІ ПОБУДОВИ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 8	
1.1. Огляд існуючих методів розпізнавання образів та обличчя.....	9
1.1.1. Метод гнучкого порівняння на графах	10
1.1.2. Метод штучних нейронних мереж	12
1.1.3. Метод прихованих Марківських моделей	14
1.1.4. Метод головних компонент або principal component analysis	14
1.1.5. Активні моделі зовнішнього вигляду і активні моделі форми	16
1.1.6 Метод Віоли-Джонса	19
1.2. Аспекти, що впливають на якість розпізнавання обличчя	19
1.3. Принцип роботи та архітектура нейронних мереж	21
1.3.1. Згорткова нейронна мережа.....	24
Висновки до першого розділу.....	25
РОЗДІЛ 2. ПРОБЛЕМАТИКА СТВОРЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧЧ	26
2.1. Рецепторна структура сприйняття інформації.....	26
2.1. Рецепторна структура сприйняття інформації.....	37
2.2. Персептрон, як модель розпізнавання	44
Висновки до другого розділу.....	51
РОЗДІЛ 3. РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧЧ.....	52
3.1. Аналіз поставленої задачі та її реалізація.....	52
3.1.1. Алгоритм нейромережевої організації матриць для розпізнавання образів.....	54
3.2. Програмна реалізація системи розпізнавання обличчя	56
3.3. Аналіз отриманих результатів	61
Висновки до третього розділу	65
ВИСНОВКИ.....	66
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ.....	68
Додаток А.....	71
Додаток Б.....	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

БД – база даних

ЗНМ – згорткова нейронна мережа

ІНС – інформаційна нейронна система

НМ – нейронна мережа

ЕОМ – електронно-обчислювальна машинна

ПРО – проблема розпізнавання образів

ПММ – прихована Марківська модель

ААМ – active appearance models

ASM – active shape model

PCA – principal component analysis

FERET – face recognition technology

ВСТУП

Не секрет, що для успішного функціонування багатьох сфер сучасного життя виникає необхідність автоматично і максимально точно ідентифікувати персону. Біометричні методи – це найпоширеніші на сьогоднішній день способи ідентифікації. Засновані ці методи на фізіологічних та поведінкових особливостях людини. Ідентифікація – це процес розпізнавання ідентифікатора користувача. Автентифікація - це процес, під час якого засвідчується істинність ідентифікатора, який був заявлений користувачем.

Розпізнавання осіб – це важливе завдання, яке є актуальним у багатьох сферах, особливо у системах безпеки і в області інтелектуальних середовищ. Вже не перший рік йде пошук, а також реалізація максимально ефективних принципів, які дозволять електронно-обчислювальним системам розпізнавати обличчя людини.

За прогнозами, застосування штучних нейронних мереж і нейрокомп'ютерів якнайкраще дозволять вирішити дане питання. Зараз є велика кількість нейромережових моделей, що допомагають розпізнавати образи. Тим не менш, нейромережам достатньо важко розпізнати образи, які були будь-яким чином деформовані – зсув, масштабування, повороти тощо. Вирішити це завдання можливо завдяки правильному вибору архітектури та способу навчання.

Отже, у підсумку можна сказати, що в наш час актуальним питанням є аналіз існуючих розпізнавальних комп'ютерних систем і формування на основі отриманої інформації оптимальної і максимально точної системи розпізнавання облич.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Достатньо довго розпізнавання різних графічних образів розглядалося в основному з боку психологічного та біологічного аспектів. При цьому вивчалися лише якісні характеристики, проте вони не в змозі були описати механізм функціонування детально і точно.

Дослідження рецепторів органів відчуття (слуху, зору, дотику) допомагали отримати функціональні залежності, але все одно не зрозуміло було, як виникають принципи формування рішення. На початку подібних досліджень вважалося, що функціонування мозку відбувається за певними алгоритмами. Саме тому всі дослідження відбувалися за таким принципом: якщо з'ясувати систему правил, за якими власне і працює мозок, то буде можливість, використовуючи різноманітні обчислювальні і технічні засоби, її відтворити.

За допомогою різноманітних девайсів, які здатні розпізнавати ті чи інші об'єкти, реалізується можливість замінити людину спеціально призначеним для певних дій автоматичним агрегатом. Це допомагає значно збільшити можливості складних систем, основним завданням яких є виконання різноманітних аналітичних, інформаційних, логічних завдань.

Необхідно пам'ятати, що при виконанні завдань людиною, якість роботи залежить від багатьох факторів, серед яких досвід, кваліфікація, стан здоров'я тощо. Разом з тим, якість роботи автомату, за умови якщо він справний, буде завжди однаковою. Такий результат забезпечується тим, що автомат завжди діє за заданим йому алгоритмом, а значить одноманітно.

<i>Кафедра КІТ (47)</i>				<i>НАУ 21.06.64.000 ПЗ</i>			
<i>Виконала</i>	<i>Закалата Д.Ю.</i>			<i>1. Теоретичні відомості та аналіз предметної області</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Моденов Ю.Б.</i>					8	18
<i>Консульт.</i>					<i>УС-211М</i>		<i>122</i>
<i>Н. Контр.</i>	<i>Райчев І.Е.</i>						

Все це можливо завдяки автоматичному контролю складних систем. З його допомогою можна моніторити стан автомату, за необхідності вчасно забезпечувати технічне обслуговування, а також розпізнавати і ліквідувати перешкоди. Також, якість передачі інформації можна значно покращити, завдяки застосуванню певних методів шумозаглушення. На додачу очевидно, що декілька автоматичних систем виконують поставлене завдання в рази швидше, аніж людина.

Отже, результати класичної теорії статистичних рішень на сьогодні служать базою для вирішення завдань віднесення об'єктів до того чи іншого класу. В рамках даної теорії будувалися спеціальні алгоритми дій, які допомагають визначити та описати певний клас, до якого можна віднести той чи інший об'єкт. Робота цих алгоритмів базується на основі експериментальних вимірювань ознак, характеризуючи об'єкт та певних апріорних даних, які описують класи [3].

1.1. Огляд існуючих методів розпізнавання образів та обличь

Методи розпізнавання графічних образів діляться на три групи:

1. Попередня фільтрація і підготовка зображення.
2. Логічна обробка результатів фільтрації.
3. Алгоритми прийняття рішень на основі логічної обробки.

Слід зазначити, що межі між цими групами достатньо умовні. Не обов'язково застосовувати методи з усіх груп для вирішення тієї чи іншої задачі. Іноколи цілком достатньо двох, або навіть одного методу. Наразі існує велике різноманіття для розпізнавання обличчя. Втім, не дивлячись на це, структура процесу розпізнавання для всіх алгоритмів одна, що зображена на Рис. 1.1.

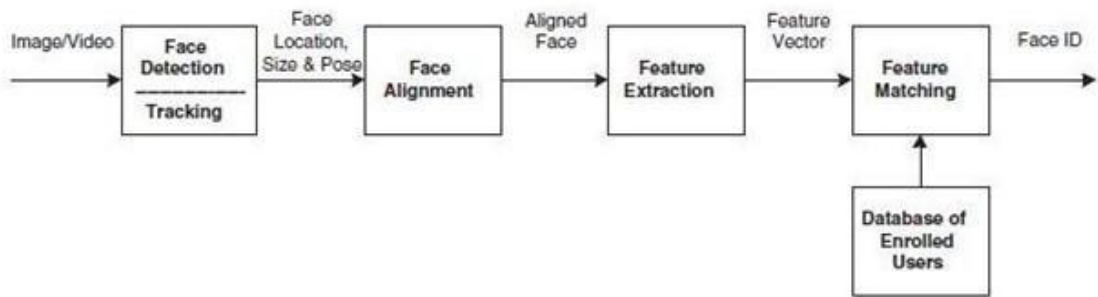


Рис. 1.1. Загальний процес обробки зображення обличчя при розпізнаванні

Перший етап представляє собою виявлення і локалізацію зображеного обличчя. При розпізнаванні зображення обличчя вирівнюється геометрично та за рівнем яскравості. Далі відбувається обчислення ряду ознак і власне розпізнавання - зіставлення визначених ознак зі зразками, які закладені у базу даних. Відмінність усіх представлених алгоритмів полягає в обчисленні ознак і порівнянні між собою їх сукупностей [12].

1.1.1. Метод гнучкого порівняння на графах

Даний метод полягає в еластичному порівнянні графів, які описують зображення обличчя (Рис. 1.2). В даному випадку обличчя виглядають як графи з вершинами і ребрами. При цьому, при ідентифікації, еталонний граф не змінюється, в той час як інший змінює форму. Це відбувається задля того, щоб найкращим чином підігнати граф, що змінюється, до зразкового. Графи в таких системах розпізнавання можуть виглядати і як прямокутна решітка, і як утворена антропометричними точками людини структура.

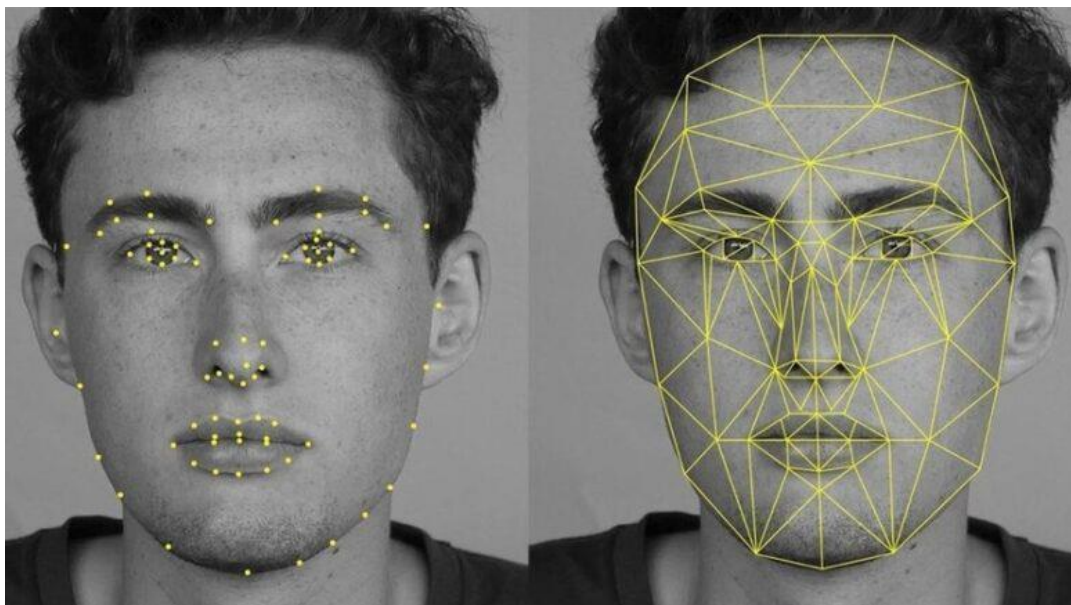


Рис. 1.2. Граф на основі антропометричних точок обличчя

На вершинах графа обчислюються значення ознак. Частіше за все використовують комплексні значення фільтрів Габора або їх впорядкованих наборів – Габорівських вейвлет (строї Габора). Ці значення обчислюються в певній локальній області вершини графа локально шляхом згортки значень яскравості пікселів з фільтрами Габора.

Відстань між суміжними вершинами необхідна для зважування ребер графа. За допомогою деякої цінової функції можна власне обчислити відстань між двома графами. Дана функція для виконання обчислення враховує рівень деформації графа і різницю між значеннями ознак.

Зміна форми графа відбувається таким чином: кожна з його вершин зсувається на певну відстань і в певних напрямках, відносно її початкового розташування. Далі відбувається вибір такої позиції вершини, при якій буде мінімальною різниця між значеннями ознак (відгуків фільтрів Габора) в вершині деформованого графа і відповідної їй вершині еталонного графа. Ця дія буде виконуватися для всіх вершин графа по чергово до тих пір, поки не буде знайдено найменшу загальну відмінність між ознаками еталонного і зміненого графів. В цьому випадку мірою відмінності між початковим зображенням обличчя і еталонним графом буде значення цінової функції деформації при такому положенні деформуючого графа. Така процедура деформації виконується для

всіх еталонних осіб, що додані до бази даних системи. Результатом розпізнавання системи є еталон, який має найкраще значення цінової функції деформації.

1.1.2. Метод штучних нейронних мереж

На сьогоднішній день існує приблизно десяток видів нейронних мереж (далі НМ). Мережа, принцип роботи якої полягає у класифікації поданого на вхід зображення або сигналу відповідно до попередніх налаштувань або навчань системи, на даний момент є однією з найчастіше використовуваних. Побудована ця мережа на багат шаровому перцептроні.

Навчання нейронних мереж відбувається на основі набору навчальних прикладів. Під час навчання здійснюється налаштування ваг міжнейронних зв'язків в процесі рішення оптимізаційної задачі методом градієнтного спуску. Основними етапами навчання НМ є: автоматичне вилучення основних (ключових) властивостей, ідентифікація їх важливості і будування між ними взаємозв'язків. За прогнозами, нейронна мережа, по завершенню навчання, застосовуватиме отриманий досвід при розпізнаванні невідомих образів за рахунок узагальнюючих здібностей.

На даний момент, аналіз публікацій показав, що Convolutional Neural Network або згортова нейронна мережа (далі – ЗНМ) найкраще справляється із завданням розпізнати ту чи іншу персону. Архітектури НМ когнітрон і неокогнітрон стали фундаментом для розвитку ЗНМ (Рис. 1.3). Можливість цих нейромереж обчислювати двовимірні топології зображення, на відміну від багат шарового перцептрона, і стала причиною успіху.

ЗНМ має такі відмінності:

- локальні рецепторні поля, які відповідають за локальну двовимірну зв'язність нейронів;
- загальні ваги (саме вони відповідають за визначення деяких рис у будь-якому місці зображення);

- spatial subsampling – особлива ієрархічна організація з просторовим семплінгом.

Ці нововведення забезпечили ЗНМ стійкість (щоправда, часткову) до всіляких зсувів, поворотів, зміни масштабу, ракурсу тощо.

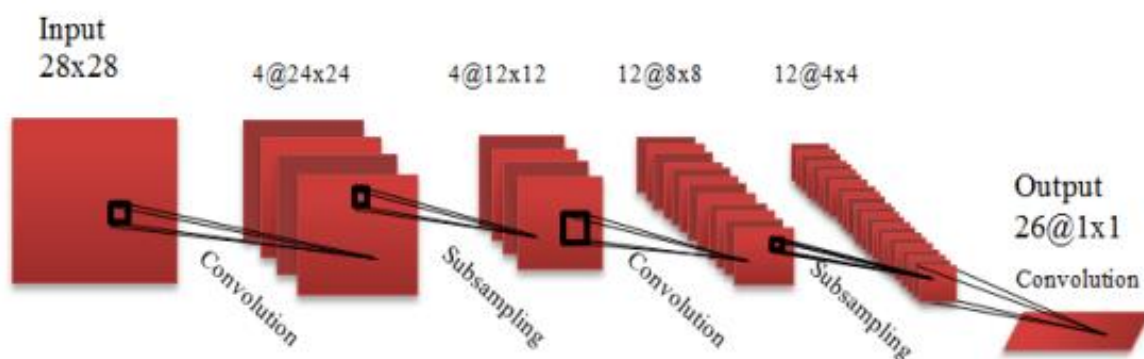


Рис. 1.3. Схематичне зображення архітектури згорткової нейронної мережі

Згорткову нейронну мережу протестували на базі даних ORL. Ця база даних містить в собі зображення людей з різними змінами, такими як зміни масштабу, освітлення, повороти тощо. За результатами тестування, точність розпізнавання цією мережею сягає 96%.

Розробка DeepFace, придбана Facebook з метою розпізнавання облич користувачів соцмережі, стала новим ступенем у розвитку ЗНМ. Особливості архітектури не розголошуються.

У нейронних мереж, звісно, є свої недоліки. Так, наприклад, якщо до бази даних потрібно додати нову еталонну персону, то в подальшому необхідно буде перенавчити нейромережу на всьому наявному наборі. Це доволі тривала процедура, яка, в залежності від розміру вибірки, може зайняти від однієї години до кількох діб. Існують також проблеми математичного характеру, які пов'язані із навчанням нейромережі, наприклад, вибір найбільш прийняттого кроку оптимізації, потрапляння в локальний оптимум, перенавчання тощо. Етап вибору архітектури мережі також достатньо важко формалізується. Підбиваючи

підсумок, можна сказати, що нейронна мережа - це такий собі «чорний ящик», результати якого важко піддаються інтерпретації.

1.1.3. Метод прихованих Марківських моделей (ПММ)

Приховані Марківські моделі (ПММ) з дискретним часом – це ще один статистичний метод розпізнавання обличчя. Принцип роботи ПММ заключається у використанні статистичних властивостей сигналів і врахування їх просторових особливостей. Велика кількість прихованих і спостережуваних станів, початкова їх ймовірність і матриця перехідних властивостей - це основні елементи моделі. Кожному з цих елементів відповідає певна Марківська модель. Під час розпізнавання того чи іншого об'єкта відбувається перевірка згенерованих для заданої бази Марківських моделей. Далі відбувається пошук максимальної із спостережуваних ймовірностей того, що послідовність спостережень для того чи іншого об'єкта згенерована відповідною моделлю. На даний момент немає прикладів застосування ПММ у комерційних цілях.

Основним недоліком ПММ є необхідність підбирати параметри моделі для кожної бази даних. Також, алгоритм навчання здатний максимізувати відгук кожного зображення на свою модель, а от мінімізувати відгук на інші моделі він не може. Це означає, що даний метод не має розрізняючих властивостей.

1.1.4. Метод головних компонент або principal component analysis (PCA)

Серед найбільш опрацьованих, і як наслідок, найбільш відомих методів є метод головних компонент. Даний метод був заснований на перетворенні Карунена Лоєва. Спершу його застосовували в статистиці для зниження простору ознак без суттєвої втрати інформації. Під час розпізнавання обличчя його застосовують з метою представлення зображення особи вектором малої розмірності (головних компонент). Далі цей вектор порівнюється з еталонними векторами, що були закладені у базу даних.

Суттєве зменшення розмірності простору ознак таким чином, щоб воно якнайкраще виконувало опис «типових» образів, які належать великій кількості осіб – це і є головна мета методу головних компонент. Використання даного методу дозволяє виявити різноманітні мінливості в навчальній вибірці зображень облич, а також описати цю мінливість в базисі декількох ортогональних (власних) векторів (eigenface).

Достатньо один раз отримати набір власних векторів на навчальній вибірці зображень. Далі саме цей набір буде використовуватися для кодування усіх інших, представлених зваженою комбінацією даних власних векторів, зображень облич. Використовуючи обмежену кількість власних векторів, ми отримуємо стислу апроксимацію вхідному зображенню персони. В подальшому вона зберігається у базі даних як вектор коефіцієнтів.

Суть цього методу зображена на Рис.1.4: першочергово увесь навчальний набір персон перетворюється на одну загальну матрицю даних. У цій матриці кожен рядок – це один екземпляр зображення обличчя, що розкладений у рядок. Усім особам навчального набору необхідно обов'язково надати єдиний розмір. Вони всі також повинні бути з нормованими гістограмами.

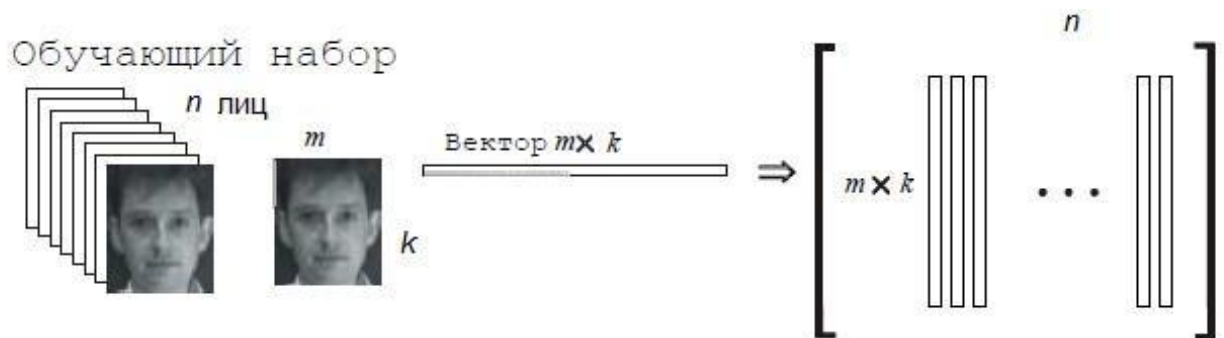


Рис. 1.4. Перетворення навчального набору осіб в одну загальну матрицю X

Найкраще метод головних компонент показав себе у практичних додатках. Втім, слід зазначити, що ефективність методу суттєво знижується, якщо в зображенні змінюється вираз обличчя або рівень освітленості. Це відбувається тому, що PCA вибирає підпростір не для того, щоб виконати дискримінацію між класами осіб, а з метою максимально апроксимувати вхідний набір даних.

1.1.5. Активні моделі зовнішнього вигляду і активні моделі форми

Активна модель має два види параметрів: форми і зовнішності. Перед початком використання, для її успішної роботи, модель потрібно навчити на великій кількості зображень із завчасною ручною розміткою. У кожній мітці на обличчі є особистий номер. Кожна мітка визначає особливу точку, метою якої і є знаходження моделі під час адаптації до нового зображення [18].

Активні моделі зовнішнього вигляду (Active Appearance Models, AAM) – це статистичні моделі рисунків. За допомогою різноманітних змін їх можна підігнати під реальне зображення. Тім Кутса і Кріс Тейлор запропонували в 1998 р. такий тип моделей у двовимірному варіанті. Оцінка параметрів зображень обличчя спочатку була основною метою використання цих моделей. З усієї цієї безлічі нормованих точок, з використанням методу PCA і виділяються основні компоненти (Рис.1.5).

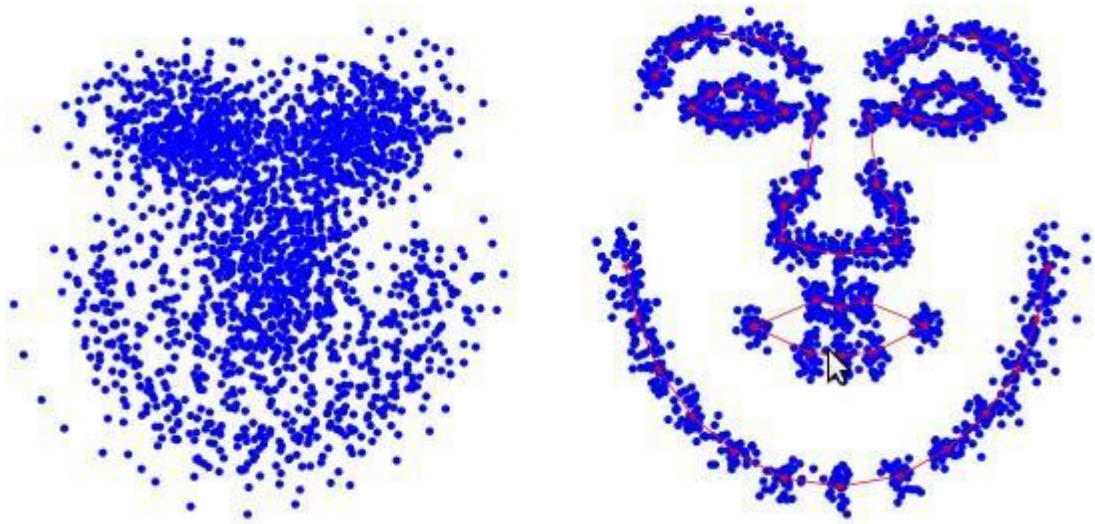


Рис. 1.5. Координати точок форми обличчя до і після нормалізації

Далі відбувається формування матриці. Ця матриця формується із пікселів, що знаходяться всередині трикутників (Рис. 1.6). Трикутники, в свою чергу, були утворені точками форми. Кожен стовпець сформованої матриці має значення пікселів тої чи іншої текстури. Ті текстури, що використовуються для навчання, можуть бути і одноканальними (градації сірого), і багатоканальними (RGB

тощо). У випадку з багатоканальними тестурами спочатку вектори пікселів утворюються окремо на кожному з каналів, а потім склеюються (відбувається так звана конкатенація). Вважається, що модель ААМ навчилась, якщо були знайдені головні компонент матриці текстур.

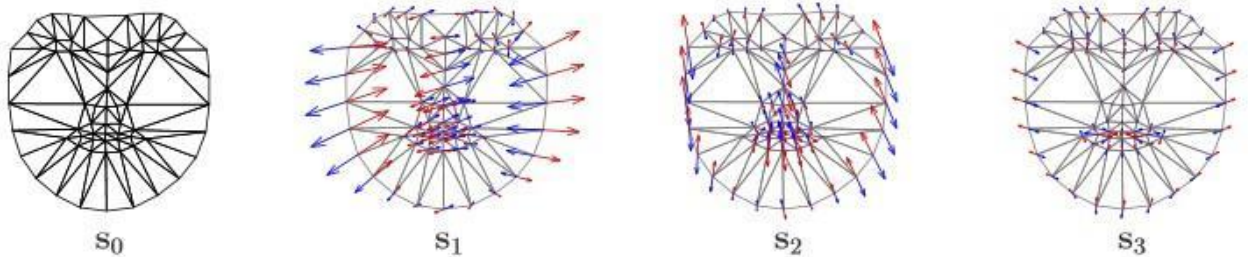


Рис. 1.6. Модель форми ААМ складається з триангуляційної решітки і лінійної комбінації зсувів

ААМ дає можливість моделювати зображення таких об'єктів, яким властива деформація (мається на увазі масштабування, перенесення або поворот). Частина набору параметрів ААМ – це форма особи, інші призначені для створення текстури. Коли ААМ вирішує завдання з локалізації персони, то шукаються параметри, такі як розташування, форма тощо. Потім з цих параметрів утворюється зображення, найбільш схоже на те, що спостерігається. При порівнянні цих зображень і робиться висновок - є обличчя чи немає.

Метод активної моделі форми (Active shape model, ASM) – це облік статистичних зв'язків між розташуванням антропометричних точок. Для такого обліку використовується набір фото із зображеними в анфас людей на них. На зображенні розміщують антропометричні точки, потім однаково їх пронумеровують на кожному зображенні (Рис.1.7).

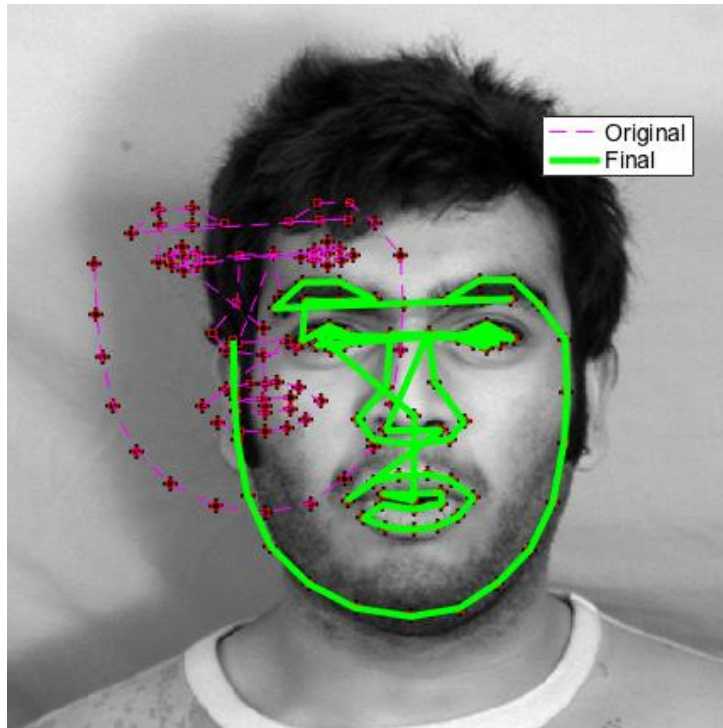


Рис. 1.7. Приклад подання форми обличчя з використанням скінченної кількості точок

Так званий узагальнений Прокрустом аналіз використовують для приведення координатів на всіх зображеннях до єдиної системи. Під час цього аналізу точки центруються, а також їм надається однаковий масштаб. Потім відбувається обчислення середньої форми і матриці коваріації (з якої в подальшому обчислюються власні вектора) для всіх образів, що є у наборі. Врешті-решт, відбувається сортування векторів за зменшенням відповідних їм власних значень. Слід зазначити, що основна мета ААМ і АСМ – це точна локалізація обличчя і антропометричних точок на зображенні і їх обробка, а не розпізнавання облич.

Вирівнювання – це неодмінний майже в усіх алгоритмах етап перед класифікацією. На цьому етапі зображення вирівнюється фронтально відносно камери або ж сукупність персон приводиться до єдиної системи координат. Щоб успішно реалізувати вирівнювання, відбувається локалізація таких антропометричних точок як куточки очей або зіниці. Ці точки обробляються саме тому, що властиві усім особам. Зазвичай, для того, щоб зменшити обчислювальні витрати, розробниками виділяється не більше десяти подібних

точок. Саме для максимально точної локалізації антропометричних точок і використовують моделі AAM і ASM.

1.1.6 Метод Віоли-Джонса

Основою метода Віоли-Джонса, розробленого у 2001 році, є: інтегральне уособлення фото за ознаками Хаара, зведення класифікатора, основою якого є алгоритм адаптивного бустингу та спосіб поєднання класифікаторів у єдину каскадну структуру. Цей метод особливо ефективний під час пошуку у реальному часі об'єктів на зображеннях і відео. Ймовірність помилково визначити обличчя у даного метода дуже низька. Виявлення обличчя відбувається навіть якщо стежити за ним під кутом 30 градусів. 90% - саме такої цифри може сягати точність розпізнавання. Цей метод дуже простий і напрочуд ефективний, а тому його часто застосовують на практиці (найчастіше – у системах відеоспостереження). Даний алгоритм можна знайти в бібліотеці OpenCV.

Методу гнучкого порівняння на графах властива велика обчислювальна складність, а також він достатньо довго ідентифікує обличчя. Метод головних компонент набагато гірше від інших розпізнає обличчя під час зміни міміки тощо. Аналіз показав, що нейронні мережі і метод Віоли-Джонса найкраще підходять для використання у системах відеоспостереження як найбільш ефективні алгоритми. Бажаними мовами для програмної системи ситуаційного відеоспостереження є Delphi, C, C ++, C #. Метод нейронних мереж зараз не реалізується на цих мовах. Отже, перевага була надана методу Віоли-Джонса, як такому, що реалізується засобами вільної бібліотеки комп'ютерного зору OpenCV. Працювати з бібліотекою можна на мовах C++ і C# (з останньою - за допомогою оболонки EmguCV).

1.2 Аспекти, що впливають на якість розпізнавання облич

Освітленість сильно впливає на якість розпізнавання, а тому є однією з основних проблем вдалої ідентифікації. За умови фото людей при однаковому

освітленні, більша частина алгоритмів без проблем успішно ідентифікують зображення. Проте, якщо в базі є фото з різним рівнем освітлення (в приміщенні, у сутінках, на вулиці), то алгоритмам стає складніше розпізнавати обличчя.

Для успішної ідентифікації необхідно вдало вибрати вихідний опис осіб для аналізу і розпізнавання. Якщо вибір буде вдалим, то з розпізнаванням не виникне проблем, а невдалий вибір значно ускладнить обробку[3]. Алгоритми, які працюють з двомірними зображеннями добре справляються зі скануванням особи анфас при хорошому освітленні в приміщенні. Алгоритм проводить аналіз унікальних точок і відстані між ними, а потім відбувається ідентифікація шляхом виводу коефіцієнту відмінності між знімком у базі і живим зображенням.

Більш стійкими до зміни світлового потоку є тривимірні технології. Вони можуть розпізнати зображення, навіть якщо воно буде відхилене на 45 градусів від фронтального ракурсу. Алгоритм, окрім ліній і точок, також аналізує властивості поверхонь і відстані між ними. Тим не менш, щоб такі алгоритми безперебійно працювали, необхідно обробляти відеозапис високої якості, частота якого сягає мінімум 200 к/с.

Отже, для того, щоб алгоритми показували високу ефективність і краще розпізнавали людей, необхідно забезпечити їх незалежність від цілого ряду факторів, серед яких: освітленість, ракурс тощо. Також, для коректної роботи алгоритмів, необхідно використовувати пристрої, які відповідають обчислювальним і продуктивним вимогам.

FERET (face recognition technology) – це створена агенством DARPA і дослідницькою лабораторією армії США програма для оцінки ефективності алгоритмів розпізнавання. Кожного року FERET звітує про порівняльні тестування сучасних алгоритмів. В останніх звітах публікується лише інформація про тестування комерційних систем і немає жодної інформації про принципи роботи алгоритмів. На сьогоднішній день, лідером серед комерційних систем є розроблена компанією NEC система NeoFace.

Тестування даною програмою проходили засновані на гнучкому порівнянні на графах алгоритми, а також різні версії методу головних компонент.

Усі протестовані алгоритми показали приблизно однакову ефективність, що призвело до складнощів у знаходженні відмінностей між ними. Так, фронтальні зображення, які датуються одним і тим же днем, алгоритми розпізнавали з точністю 95%.

80% - точність розпізнавання фото з різним освітленням, зроблених різними фотокамерами. Близько 50% - точність розпізнавання фото, різниця між якими один рік (але для системи такого типу, навіть 50% - це хороший результат).

1.3 Принцип роботи та архітектура нейронних мереж

Людський мозок – це дуже складна нелінійна паралельна обчислювальна система. Завдяки дослідженням нейронних мереж у людства з'явилася можливість створити штучну систему, яка за своїми вміннями зможе максимально наблизитись до можливостей обробки інформації мозком людини. Мозок організовує свої нейрони (структурні компоненти) таким чином, що вони здатні виконувати ті чи інші задачі набагато швидше, чим це роблять надпотужні сучасні комп'ютери.

Нейронні мережі у своїй роботі моделюють роботу процесів у людському мозку. Це системи, які здатні навчатися. Навчання нейронної мережі відбувається за допомогою аналізу впливів – як позитивних, так і негативних. Структурним компонентом у нейронній мережі є названий на честь біологічного прототипу нейрон. Формальний нейрон(Рис. 1.8) – це складова частина більшості нейромереж [4].

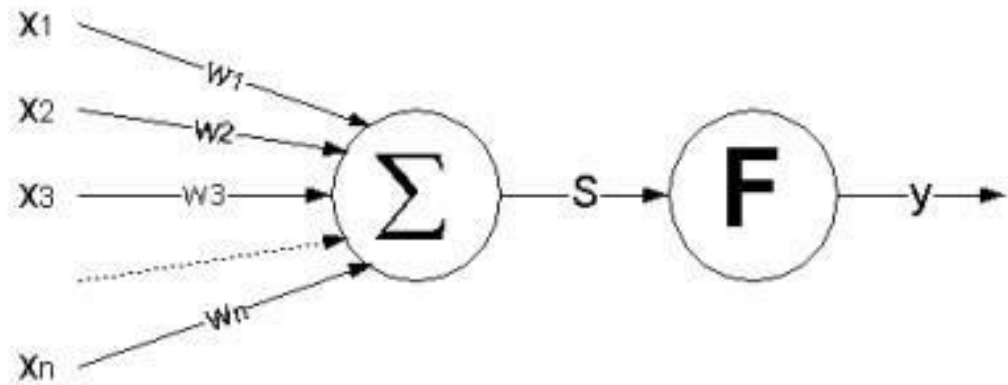


Рис. 1.8. Формальний нейрон

Тут $x_1..x_n$ – значення, що надходять на входи (синапси) нейрона, $w_1..w_n$ – ваги синапсів (можуть бути як гальмуючі, так і підсилюючі), S - зважена сума вхідних сигналів.

$$S = \sum_{i=1}^n w_i x_i - T = (W, X) - T = |W| * |X| * \cos \alpha - T,$$

де T – поріг нейрона (у багатьох моделях обходяться без нього), F – функція активації нейрона, що перетворює зважену суму в вихідний сигнал. Зважена сума вхідних сигналів може бути інтерпретована як проекція вхідного вектора на вектор ваг, де α – кут між цими векторами.

Як було сказано вище, біологічний нейрон - це прототип формального нейрона. Пов'язані формальні нейрони – це і є штучна нейромережа. Можливості штучних нейронних мереж, насправді, широкі. Вони виконують будь-які логічні операції. Втім, очевидно, що функціонування біологічних нейронів – це набагато складніший процес. 10 трильйонів нейронів, які пов'язані 10^{14} синапсами – ось що таке людський мозок. Настільки потужної техніки, яка б повністю моделювала людський мозок, поки не існує.

Проте, у штучних нейронних мереж є великі перспективи розвитку – адже, хоч структура мозку і є генетично визначеною з народження, зв'язки між нейронами створюються, розвиваються і змінюються весь час, поки живе

людина. Простіше кажучи, людина навчається, і саме цим займаються нейронні мережі.

Нейронні мережі бувають повнозв'язні (кожен нейрон пов'язаний з іншими) і шаруваті (кожний шар нейронів пов'язаний лише з попереднім шаром). Нейронні мережі також відрізняються і за характером функціонування – бувають однопрохідні і релаксаційні.

Нейронні мережі здатні навчатися на основі даних довкілля. Результатом цього навчання є підвищення продуктивності нейронної мережі, яке відбувається поступово. Інтерактивний процес коригування синаптичних ваг і порогів - це те, за допомогою чого і відбувається навчання нейронної мережі.

Нейронні мережі також відрізняються за видом вхідної інформації (вона може бути двійковою, аналоговою тощо) і методом навчання.

Навчатися нейронні мережі можуть такими способами[14]:

- Навчання з учителем. Налаштування необхідних зв'язків відбувається у процесі навчання. Такий вид навчання передбачає, що еталонні значення результатів роботи відомі з самого початку.
- Самонавчання. Еталонні значення невідомі і не потрібні. Під час навчання мережа організовує вхідні образи на основі їх подібності.
- Фіксований зв'язок. Визначаються характером розв'язуваної задачі.

Контроль якості води, знаходження бомби у багажі – ці та безліч інших задач виконують програмні продукти, створені на базі нейронних мереж.

Потенційна висока швидкодія нейронних мереж стає можливою завдяки використанню масового паралелізму обробки інформації. Продуктивність нейронних мереж, у випадку несприятливих умов, не сильно падає – спостерігається лише незначне ускладнення вилучення інформації. Суттєво вплинути на працездатність нейронної мережі можуть лише серйозні поломки її структури. Така стійкість можлива завдяки розподіленому характеру зберігання інформації у нейронній мережі.

1.3.1. Згорткова нейронна мережа

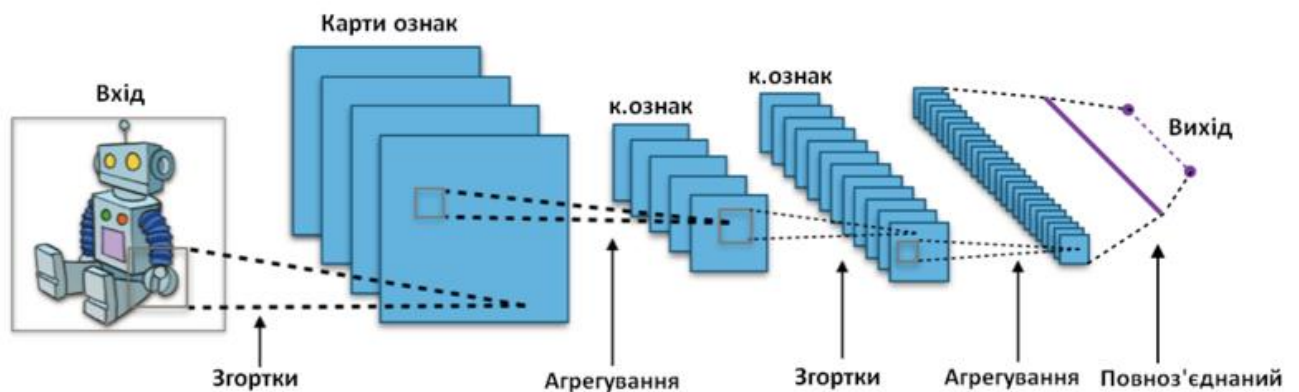


Рис. 1.9. Архітектура згорткової нейронної мережі

У звичайному перцептроні (який є повнозв'язною нейронною мережею) кожен нейрон пов'язаний з усіма нейронами попереднього шару. При цьому у кожного зв'язка є свій ваговий коефіцієнт.

Згорткова нейронна мережа за допомогою згортки зменшує кількість інформації, що зберігається в пам'яті, і одночасно виділяє опорні ознаки зображення – ребра, контури, грані. На наступному рівні обробки з цих ребер і граней система розпізнає фрагменти повторюваних текстур. В подальшому вони складаються у фрагменти фотографії.

Кожен з шарів нейронної мережі використовує власне перетворення. В перших шарах мережа працює з ребрами, гранями, контурами тощо. На наступних етапах вже працює з текстурою і частинами об'єктів. Результатом такої роботи стає можливість правильно ідентифікувати зображення або виділити на завершальному етапі власне об'єкт, який шукається на фото.

Висновки до першого розділу

У сучасному світі системи розпізнавання грають важливу роль у житті людини. Технології розпізнавання образів, і зокрема облич використовуються у різних сферах діяльності людини, серед яких: системи охорони; фейс-контроль у сфері громадського харчування для ідентифікації потенційно небезпечних або підозрілих гостей; онлайн-платежі; фототехніка; криміналістика; мобільні додатки; пошук фото у великих базах знімків тощо. Можливості систем розпізнавання облич зараз дуже широкі. Так, наприклад, соціальна мережа Facebook сама розпізнає ваших друзів у вас на фото і вам більше не потрібно при завантаженні фото відмічати друзів, адже мережа робить це самостійно.

На сьогоднішній день є такі основні засоби розпізнавання: нейронні мережі, метод головних компонент, метод Віюлі-Джонса, метод порівняння на графах. Найефективнішими вважаються нейронні мережі. Втім, в них теж є свої мінуси. Основні їх недоліки заключаються в повторному навчанні нейромережі при зміні еталону, виборі кроку оптимізації, виборі архітектури. Попри недоліки нейронних мереж, саме цей метод обрано для розробки програми по розпізнаванню облич і в наступному розділі будемо більше аналізувати проблематику і вирішення питань з розробки нейронної.

РОЗДІЛ 2.
ПРОБЛЕМАТИКА СТВОРЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ
РОЗПІЗНАВАННЯ ОБЛИЧ

2.1. Рецепторна структура сприйняття інформації

Для того, щоб бути свідомо сприйнятою людиною, інформація проходить довгий цикл попередньої обробки. На першому етапі в око потрапляє світло. Фотони, після того, як пройдуть всю оптичну систему, потрапляють на сітківку (це такий шар паличок і колбочок - особливих клітин, які дуже чутливі до світла). І хоч сітківка знаходиться далеко від головного мозку, саме з неї починається перший етап обробки інформації. Так відбувається тому що у ссавців, наприклад, два шари нервових клітин, які призначені для виконання відносно легкої роботи, знаходяться одразу за світлочутливими клітинами.

Далі отримана інформація направляється до головного мозку по зоровому нерву. В головному мозку знаходяться такі собі "зорові горби", на які, власне, і проектується та зорова інформація, яку ми бачимо у підсумку. Далі зорова інформація направляється до відділів мозку, основною задачею яких є виділити з неї окремі складові, такі як лінії (горизонтальні, вертикальні, діагональні), області світлого, темного, кольорового та контури. Образи поступово стають все складнішими і більш розмитими. Тим не менш, перш ніж досягнути рівня свідомості, графічному образу картини потрібно буде пройти ще декілька етапів.

За аналогією з біологічними процесами можна підходити і до завдання розпізнавання образів. Спроможність тварин до розпізнавання образів, у певних умовах значно вище, ніж спроможність будь-якого пристрою, який можна побудувати.

Кафедра КІТ (47)				НАУ 21.06.64.000 ПЗ			
<i>Виконала</i>	<i>Закалата Д.Ю.</i>			<i>2. Проблематика створення комп'ютерної системи розпізнавання облич</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Моденов Ю.Б.</i>					26	26
<i>Консульт.</i>					УС-211М		122
<i>Н. Контр.</i>	<i>Райчев І.Е.</i>						

Люди також легко перевершують машини при класифікації, яка заснована на сенсорному досвіді, тобто при розпізнаванні осіб або мови. Проте, дії людей будуть менш ефективними у так званих “несенсорних ситуаціях”. Це означає, що люди програють програмам класифікації образів у випадку, коли у правильному способі класифікації присутні такі логічні комбінації абстрактних властивостей як розмір, форма і колір. Розпізнавання образів - це функція нейронів. Тому, саме у властивостях нейрона потрібно шукати ключ до біологічного розпізнавання образів.

Нейрон також можна розглядати у якості граничного елемента. Це значить, що за умови досягнення суми його входів певного значення, він на виході дає постійну величину, або залишається пасивним. Мак-Каллоком і Піттсом було доведено, що належним чином організована мережа ідеальних нейронів (це порогові елементи, логічні властивості яких можна надати реальному нейрону) може допомогти втілити в життя будь-яку обчислювальну функцію. Основна задача полягає в тому, чи є можливість знайти такий розумний принцип перелаштування мережі, який дозволить випадково об'єднаній групі ідеальних нейронів самоорганізуватися в обчислювальний пристрій, який зможе вирішити будь-яке випадкове завдання з розпізнавання образів. Такий принцип перелаштування з'явився б теорією навчання, яку можна було б застосовувати на рівні окремого нейрона.

Важливий вплив на штучний інтелект справила висунута канадським психологом Хеббом нейрологічна теорія. Спочатку дана теорія використовувалась лише як призначена для психології модель. В подальшому ж її модифікували і застосовували під час визначення перцептронів - принципів системи розпізнавання образів. Описані Розенблаттом перцептрони здатні існувати і як програми, і як спеціально розроблені комп'ютери.

Класифікаційна сукупність у системі класифікації, що об'єднує в собі деяку групу об'єктів за певною ознакою, називається клас або образ. Образне сприйняття світу є важливою властивістю живого мозку. Вона дозволяє розібратися в безмежному потоці інформації, що сприймається, і впевнено

орієнтуватися серед різних даних про навколишній світ. Коли ми сприймаємо зовнішній світ, то завжди класифікуємо вхідну інформацію. Це означає, що відбувається розподіл інформації на групи схожих, але не повністю відповідних явищ. Наприклад, хоча всі букви “Р”, написані різними почерками і будуть суттєво відрізнятися між собою, вони все одно належатимуть до однієї групи. Для того, щоб скласти поняття про всю групу сприйняття, достатньо ознайомитися з невеликим числом її представників. Саме завдяки цій властивості мозок може сформувати таке поняття як образ [7].

Характерна властивість образу полягає в тому, що завдяки ознайомленню з кінцевою кількістю явищ з однієї і тієї самої безлічі, можна дізнатись будь-яке велике число його представників. Відповідно до цього, різні люди, для навчання яких використовуються різні матеріали спостережень, в основному, незалежно один від одного, однаково класифікують одні і ті ж самі об'єкти. Саме завдяки об'єктивності усіх образів, люди з усього земного шару можуть один одного розуміти.

За допомогою можливості сприймати зовнішній світ у вигляді образів можна з деякою вірогідністю дізнаватися безмежну кількість об'єктів, ознайомившись з їх кінцевим числом. В той же час, за допомогою об'єктивного характеру головної ознаки образів, можна зробити модель процесу їх розпізнавання.

Ідентифікація об'єкта може відбуватися по його зображенню (оптичне розпізнавання) або аудіозапису (акустичне розпізнавання). Вдаючись до допомоги зорового і слухового апаратів, еволюціонуючи, більша частина тварин навчилася успішно розпізнавати образи. Однак, створення штучних систем, які зможуть розпізнавати образи, все ще залишається непростю технічною проблемою.

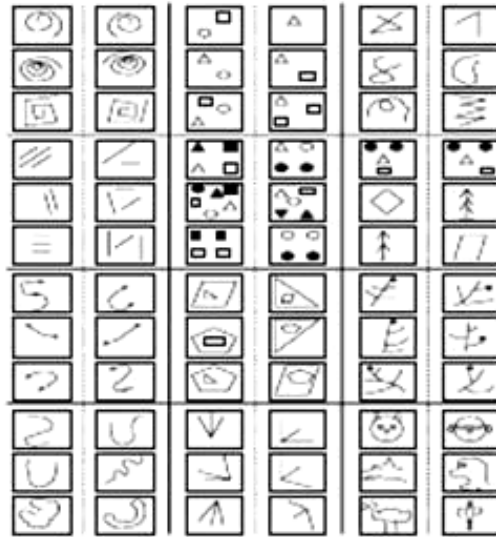


Рис. 2.1. Архітектура згорткової нейронної мережі

Проблема розпізнавання образів або ПРО ділиться на дві частини – навчання і розпізнавання. Під час навчання відбувається демонстрація окремих об'єктів, при цьому обов'язково зазначається їх приналежність до певного способу. У разі, якщо навчання пройде успішно, система зможе однаково реагувати на всі об'єкти одного образу, та по-різному – на об'єкти відмінних образів. Процес навчання обов'язково має завершитись лише шляхом показів кінцевого числа об'єктів. Об'єктами навчання можуть виступати картинки (Рис. 2.1), а також інші візуальні символи, наприклад, букви або цифри[1]. Під час навчання вказуються лише власне об'єкти, а також їх належність образу. Після навчання йде процес розпізнавання нових об'єктів. Він визначає дії вже навченої системи. Автоматизація даних дій є проблемою навчання розпізнавання. Цю проблему можна частково вирішити, якщо замість програми навчання (головну частину проблеми) буде проходити людина – це означає, що вона буде сама придумувати або розгадувати, а потім нав'язувати машині правила класифікації.

Розпізнавальні системи допоможуть вирішити велику кількість задач. Окрім власне розпізнавання слухових і зорових образів, це можуть бути завдання, пов'язані із класифікацією складних процесів і явищ, які можуть виникнути, наприклад, при виборі найбільш прийняттого управління технологічними, транспортними, військовими завданнями тощо. Перед аналізом

об'єкта обов'язково необхідно отримати про нього інформацію. Одним з основних завдань проблеми розпізнавання образів є вибір вихідного опису об'єктів. Якщо вибір буде вдалий, то завдання розпізнавання виявиться простим, і навпаки, при невдалому виборі початкового опису обробити інформацію буде складно, а інколи взагалі неможливо.

Зображення, що виникає під час навчання в результаті спостереження об'єкта, можна представити як вектор i , відповідно, як точку певного простору ознак. Якщо зображення при їх показі можна віднести до одного з двох або більше образів, то це означає, що в певному просторі є дві або більше області, які не мають спільних точок, і що зображення – це точки з даних областей. Кожній такій області можна дати відповідну до образу назву.

Тепер, обмежившись поки варіантом розпізнавання лише двох образів, пояснимо процес адаптації розпізнавання в термінах геометричної картини. Попередньо відомим вважається ті факти, що необхідно розділити дві області в певному просторі і що показуються точки лише з цих областей. В той же час не відома інформація про розташування їх меж, а також про правила визначення належності точки до певної області.

Під час навчання надаються рандомно вибрані з цих областей точки і повідомляється до якої області вони належать. Ніякої додаткової інформації про області не надається. Головною метою навчання є:

а) побудова поверхні, яка розділяє всі, а не лише показані точки, що належать цим областям;

б) побудова поверхні, що обмежує дані області таким чином, що в кожній з них знаходяться лише точки одного образу.

Якщо надані зображення належать не двом, а більшій кількості образів, то потрібно по показаним під час навчання точкам побудувати поверхню, яка розділяє відповідні образам області одну від одної. Це можна зробити, побудувавши функцію, яка набуває однакового значення, знаходячись над точками кожної з областей. При цьому над точками з різних областей її значення повинно відрізнятися.

Спочатку може здаватися, що для відокремлення всієї області недостатньо знати лише деяку кількість точок. Справді, можна вказати необмежену кількість областей, які містять ці точки, і побудувати поверхню, що виділяє область. І також можна вказати іншу область, яка і містить вказані точки, і в той же час перетинає поверхню. Тим не менш відомо, що завдання про наближення функції за інформацією про неї в обмеженій множині точок, істотно більш вузької, ніж вся безліч, на якій власне і задана функція, є звичайним математичним завданням про апроксимації функцій. Для того, щоб вирішити це завдання, необхідно ввести деякі обмеження на клас розглянутих функцій.

Які саме обмеження слід ввести залежить від типу інформації, яку може ввести вчитель на етапі навчання. Гіпотеза про компактності образів є однією з таких підказок. Завдання апроксимації функції буде тим легше, чим більш компактні і більш рознесені у просторі будуть області, які необхідно поділити. Наприклад, поділ у випадку, що показаний на Рис. 2.2 (а), свідомо простіший, ніж на Рис. 2.2 (б). Дійсно, у випадку, що на Рис. 2.2 (а), області можуть бути розділені площиною, і навіть при суттєвих погрішностях у визначенні функції, вона все одно буде продовжувати розділяти області. У випадку на Рис. 2.2 (б) поділ здійснюється хитромудрою поверхнею, до того ж незначні відхилення в її формі призводять до помилок поділу. Саме це інтуїтивне уявлення про те, що області можна порівняно легко розділити, і стало причиною появи гіпотези компактності.

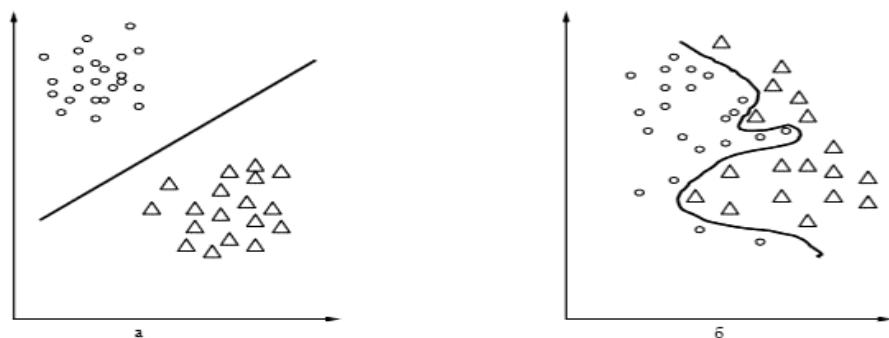


Рис. 2.2. Архітектура згорткової нейронної мережі

Окрім геометричної інтерпретації проблеми адаптації розпізнавання образів є ще один підхід, який називається лінгвістичний або структурний.

Даний підхід можна пояснити на прикладі розпізнавання зорових зображень. В першу чергу виділяється набір вихідних понять – типових фрагментів, які можна зустріти на зображеннях, і характеристик взаємного розташування фрагментів, наприклад, зліва, знизу, всередині тощо. На основі цих вихідних понять утворюється словник, з допомогою якого можна будувати різноманітні логічні твердження. Задача полягає в тому, щоб з великого числа тверджень, які можна створити за допомогою цих понять, відібрати найбільш визначальні для конкретного випадку.

На наступному етапі, при перегляді кінцевої і бажано невеликої кількості об'єктів з кожного образу, потрібно побудувати опис даних образів. Опис має бути максимально повним і точним, щоб з його допомогою можна було визначити, якому образу належить цей об'єкт. Для того, щоб реалізувати лінгвістичний підхід, потрібно побудувати вихідний словник і задати правила опису елементів даного словника.

При використанні лінгвістичної інтерпретації, ми проводимо аналогію між синтаксисом мови і структурою зображень. Для цієї операції використовується апарат математичної лінгвістики. Для опису структури зображень даний апарат можна використовувати лише після проведення сегментації зображень на складові частини - побудовані слова для опису фрагментів, а також методи їх пошуку. Після виділення слів власне і виникають лінгвістичні завдання. Їх складовою частиною є завдання автоматичного граматичного розбору описів для розпізнавання зображень. Під час цих процесів виникає самостійна область досліджень. Для успішного її опанування необхідні як знання засад математичної лінгвістики, так і опанування прийомів, що були розроблені для лінгвістичної обробки фотографій і інших зображень.

Припустимо, що простір ознак в процесі навчання формується згідно із задуманою класифікацією. Тоді завдання в просторі ознак вже саме по собі задає властивість, під дією якої образи у даному просторі легко розділяються. Згідно з теорії компактності, компактним образам притаманні відповідні компактні безлічі у просторі ознак. Компактна безліч – це так звані згустки точок в просторі

зображень, між якими, імовірно, існують розрядження, які їх розділяють. І хоча гіпотезу не завжди вдавалось підтвердити експериментально, в рамках завдань, коли вона добре виконувалась (Рис. 2.2, а) завжди вдавалось знайти просте рішення. У тих же випадках, коли гіпотезу неможливо було підтвердити (Рис. 2.2, б), вирішити завдання було складно, а іноді і зовсім неможливо. Після таких випадків справедливість гіпотези компактності поставили під сумнів. Проте, незважаючи на це, гіпотеза все одно продовжувала викликати інтерес у тих випадках, коли її можна було підтвердити. Гіпотеза компактності стала ознакою можливості задовільного вирішення завдань розпізнавання.

Навчання - це процес, результатом якого є вміння системи відповідати необхідними реакціями на зовнішній вплив. Адаптація – це підстроювання параметрів і структури системи для того, щоб досягнути бажаної якості управління при нескінченних змінах зовнішніх умов. Всі зображення на Рис. 2.1, характеризують завдання навчання. Для кожного з цих завдань задається навчальна послідовність правильно вирішених завдань. Так зване завдання навчання без вчителя можна було б поставити у випадку, якщо вийде помітити якусь загальну властивість, яка не залежить від природи образів та їх зображень, а визначальною її рисою буде здатність до роздільності. Це завдання можна описати так: системі демонструються одночасно або по черзі об'єкти без будь-якої інформації про їх належність до образів.

Вхідний пристрій системи відображає безліч об'єктів на безліч зображень. Далі він самостійно класифікує об'єкти, використовуючи деяку заздалегідь закладену в нього властивість розділення образів. По закінченню самонавчання, система зможе розпізнавати не лише знайомі об'єкти, а й до цього їй невідомі. Завдяки процесу самонавчання, система без підказки вчителя вироблятиме однакові реакції на зображення об'єктів одного і того ж самого образу і різні - на зображення різних образів[3]. Вчитель у цьому процесі лише підказує властивості (наприклад, властивість компактності) об'єкта, які будуть однакові для всіх образів, і аналізує, чи здатні безлічі об'єктів ділитися на образи.

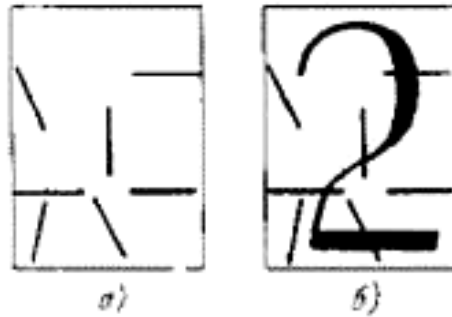


Рис. 2.3. Схема розпізнавання цифр

Під час навчання у системі виробляється та чи інша реакція на групи зовнішніх ідентичних сигналів. Цей процес відбувається у результаті неодноразового впливу на систему зовнішнього коригування – так зване заохочення і покарання. Алгоритм навчання майже повністю визначає механізм генерації даного коригування. Основною відмінністю самонавчання від навчання є те, що системі недоступна будь-яка додаткова інформація про вірність реакції.

Переважає число відомих алгоритмів самонавчання здатні виділити тільки абстрактні образи. В результаті самонавчання стає відомо, чи можна використати обраний простір для навчання розпізнаванню. Якщо виділені під час навчання абстрактні образи відповідають реальним, то це значить, що простір було обрано успішно. Чим менше відповідності між реальними і абстрактними образами, тим більш невдалим є обраний простір.

Адаптацією називають зміну параметрів і структури системи для досягнення необхідного стану системи в умовах початкової невизначеності і у випадку, якщо умови роботи є мінливими.

Для побудови розпізнавання машин, можна використовувати спосіб, оснований на відмінностях будь-яких ознак, підлягаючих розпізнаванню фігур. Ознаками можуть служити різноманітні особливості фігур – геометричні, топологічні тощо. В деяких машинах для розпізнавання букв або цифр використовується так званий метод зондів (Рис. 2.3). Згідно цього методу, розпізнавання відбувається за кількістю перетинів контура фігури з кількома особливим способом розташованими прямими.

Спроектувавши цифри на поле із зондами, то можна виявити, що кожна цифра повністю перетинає деякі зонди. При цьому для всіх 10 цифр комбінації перетину будуть різними. Саме ці комбінації і використовують як ознаки і з їх допомогою розрізняють цифри. Машини такого типу успішно виконують певні завдання, наприклад, читання машинописного тексту. Тим не менш, їх можливості обмежені шрифтами, для яких розробили систему ознак. Створювати набір еталонних фігур або систему ознак обов'язково повинна людина. Якість роботи машини залежить від попередньої підготовки і не може бути покращена без людського втручання.

Для вводу зображення в машину необхідно перевести його на машинну мову – комбінацію певних символів, з якими працює машина. Кодувати плоскі фігури можна різними способами. Краще намагатися досягти більш природного кодування зображень. Фігури будемо зображати на полі, розбитому на однакові квадрати горизонтальними і вертикальними лініями. Елементи, на які впало зображення, необхідно повністю затемнити, а все інше залишити білим. Будемо вважати, що чорні елементи – це одиниці, а білі – нулі. Далі потрібно ввести послідовну нумерацію елементів поля, наприклад у кожному рядку зліва направо. В цьому випадку кожна фігура, яка намальована в такому полі, однозначно відобразатиметься кодом, складеним з тієї кількості одиниць і нулів, що відповідає кількості елементів на полі.

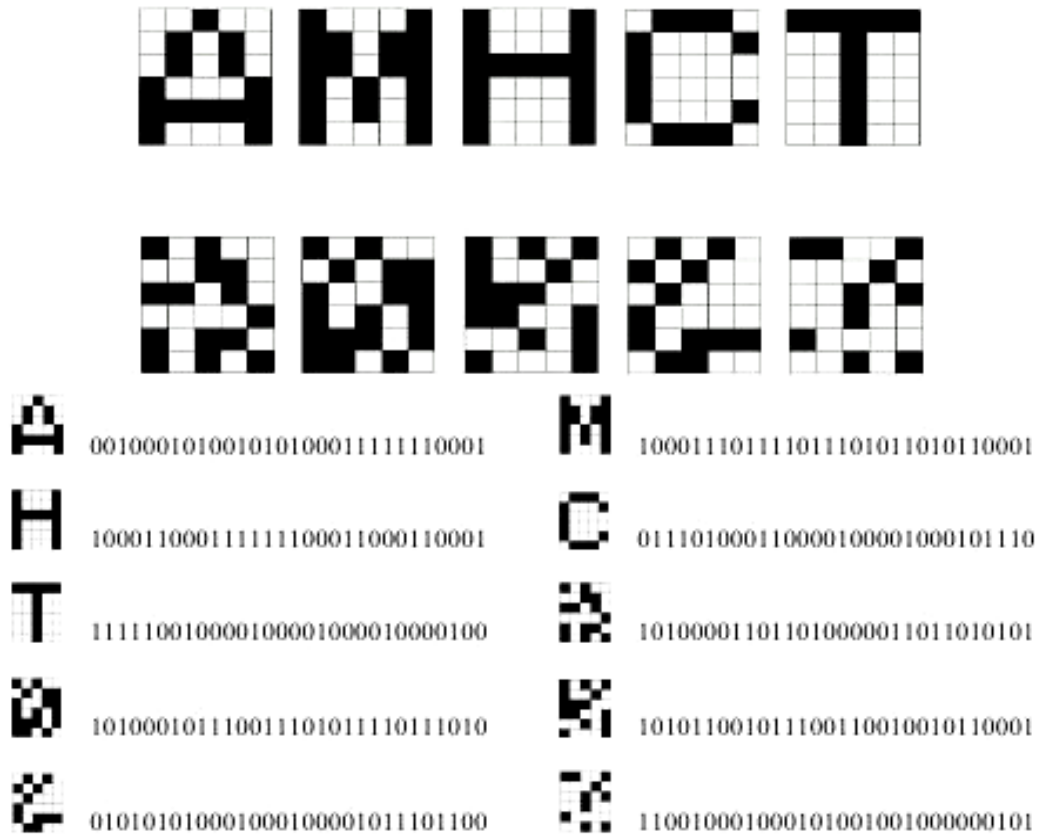


Рис. 2.4. Приклади кодування зображень

В основі зорового апарату людини лежить розбиття зображення на елементи, тому саме показаний на Рис. 2.4 спосіб кодування вважається найбільш природним. Сітківка ока людини складається з великої кількості чутливих елементів – паличок і колбочок. Вони пов'язуються за допомогою нервових волокон з головним мозком, а саме з його зоровими відділами. Сигнали, що подають ці елементи на головний мозок, відрізняються інтенсивністю. Вона залежить від того, наскільки освітлений елемент. Отже, зображення, яке оптична система спроектувала на сітківку, ділять палички і колбочки. Далі інформація передається у головний мозок. Рецепторами називають окремі елементи поля.

Усі плоскі фігури, що можна зобразити на полі рецепторів, складають деяку безліч. Об'єктом цієї безлічі, якому відповідає певний код, є кожна окремо взята фігура. Зображенню на полі також відповідає код. Зображення, завдяки повній відповідності коду, можна за допомогою цього самого коду і відтворити. Тому при виконанні завдання можна оперувати лише кодом.

Ємність ІНС – це певна кількість образів, що надаються на входи ІНС для ідентифікації. Щоб поділити множину вхідних образів, наприклад, на два класи (де “1” і “0” будуть відповідати окремому класу), потрібен лише один вихід. З двома виходами вже можна закодувати чотири класи, і так далі. Щоб класифікація була максимально достовірною, необхідно ввести надмірність. Це можна зробити шляхом виділення для кожного класу один або декілька нейронів у вихідному шарі. Кожен з цих нейронів буде навчатися визначати належність образу до класу, виходячи зі свого ступеню достовірності – високого, середнього або низького. За допомогою таких ІНС можна класифікувати вхідні образи, які поєднані в нечіткі безлічі. Така властивість ІНС дозволяє наблизити їх до умов реального життя.

2.1. Рецепторна структура сприйняття інформації

Можливість програмно реалізовувати на ЕОМ будь-які алгоритми перетворення інформації називається алгоритмічною універсальністю ЕОМ. Тим не менш, помилково вважати, що ЕОМ або роботи можуть виконати будь-які завдання [10]. Доведено, що є такі завдання, які неможливо вирішити за допомогою єдиного ефективного алгоритму. Саме тому ЕОМ не здатні виконати завдання такого типу.

Важливою властивістю штучних нейронних мереж є те, що вони можуть навчатися різними методами. Навчання нейромереж схоже на процес інтелектуального розвитку людини. І хоча можливості навчання нейромереж все ще обмежені, вже існують такі важливі досягнення як “розмовляюча мережа” Сейновскі, і регулярно знаходяться нові практичні застосування [5]. Навчання мережі відбувається для того, щоб вона могла давати для певної безлічі входів відповідну безліч виходів. Безліч входів і виходів представлені як вектори. Під час навчання ваги мережі стають такими, щоб кожним вхідним вектором вироблявся вихідний вектор.

Навчальні алгоритми бувають з вчителем та без нього. В основі навчання з вчителем лежить порівняння бажаних і дійсних значень виходів із подальшим

їх коригуванням шляхом зворотнього зв'язку. Ця модель навчання є біологічно неправдоподібною, а тому її часто критикують.

Навчання з вчителем є більш правдоподібним. Запропонували цю модель Кохонен та інші. Їй не потрібен цільовий вектор для виходів, а значить, вона не потребує порівняння з певними ідеальними відповідями. До навчальної множини належать лише вхідні вектори. У процесі навчання виділяються статистичні властивості навчальної множини, а схожі вектори групуються у класи. Пред'явлення на вхід вектора з даного класу дасть певний вихідний вектор, але до навчання немає можливості передбачити, який буде вироблятися даним класом вхідних векторів. Отже, виходи такої мережі мають перетворитися на зрозумілу форму, зумовлену процесом навчання.

Стадії процесу навчання Штучною Нейронної Мережі (ІНС) нового класу задач:

1. Постановка задачі і визначення набору ключових характеристик, які описують предметну область.

2. Вибір парадигми нейронної мережі, яка максимально підходить для вирішення певного класу завдань.

3. Підготовка набору навчальних прикладів. Вони представляють собою набір вхідних даних, які асоціюються з відомими вихідними значеннями. Вхідні значення можуть бути неповні або частково суперечливі.

4. Вхідні дані почергово демонструються ІНС, а потім значення, яке отримано на вході, порівнюється із еталоном.

5. Навчання продовжується до досягнення сумарної помилки у безлічі всіх вхідних значень прийнятної рівня.

Такий метод навчання називається зворотним поширенням помилки (error backpropagation). Він є одним з класичних алгоритмів нейроматематики. Налагоджену і навчену ІНС можна використовувати на реальних вхідних даних. Вона може не лише пропонувати коректне рішення користувачу, а й оцінювати ступінь його достовірності.

Одним з багатьох алгоритмів, який навчає машину розпізнавати образи, є алгоритм, заснований на методі січних гіперплощин (Рис. 2.5). Він полягає в апроксимації розділяючої гіперповерхні "шматками" гіперплощин і складається з таких етапів:

1. Навчання. В цей час формується розділена поверхня: проводяться січні площини, вирізаються зайві площини або їх частини.

2. Розпізнавання об'єктів.

Метод паралельних варіантів дозволяє на одному матеріалі навчати декілька машин. Навчання машин відбувається одночасно і незалежно одна від одної. При вивченні нових об'єктів кожна машина відносить до певного образу, не обов'язково до одного і того самого. Далі відбувається так зване голосування машин, у результаті якого об'єкт відноситься до того образу, який обрала більшість машин.

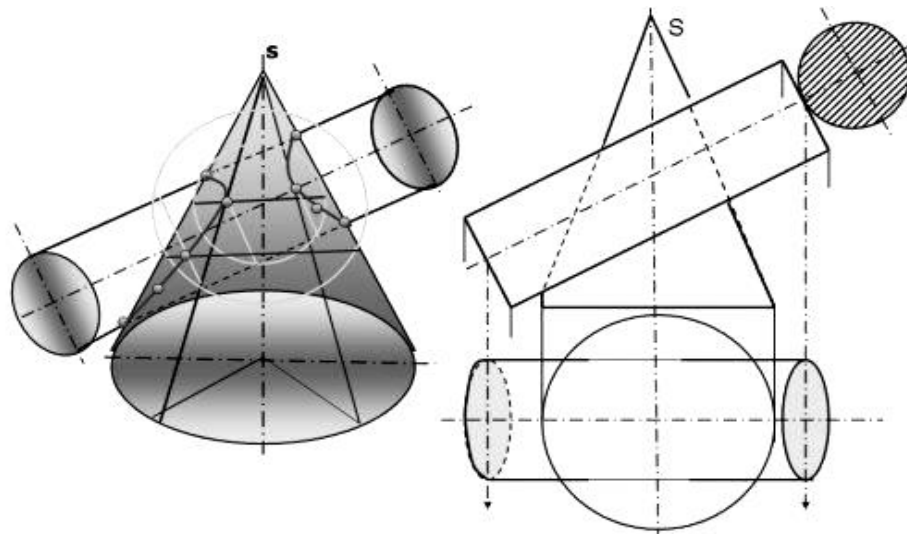


Рис. 2.5. Метод січних гіперплощин

Спосіб підвищення надійності розпізнавання – це дещо покращений спосіб проведення січних площин. Припускається, що якщо провести січні площини близько до площини, яка проходить через середину прямої, що з'єднує об'єкт і опонент, перпендикулярній цій прямій, то в результаті поверхня буде ближче до істинних кордонів між образами. Це припущення підтвержене експериментально.

Алгоритм, який заснований на методі потенціалів, полягає в тому, що з кожним збудженим елементом поля рецепторів можна пов'язати деяку функцію. Ця функція повинна дорівнювати одиниці на цьому елементі і спадати в усіх напрямках від нього. Це означає, що функція φ відповідає електричному потенціалу, а їх різниця полягає в тому, що R в нашому випадку - це відстань між двома сусідніми елементами поля рецепторів. Ця функція може бути апроксимована ступінчастою функцією, постійною в межах одного рецептора і стрибкоподібно змінюватися на межах рецепторів.

$$\varphi(R) = \frac{1}{1 + \alpha R^2}$$

Підрахунок відбувається таким чином: власний потенціал кожного збудженого на полі рецепторів елемента дорівнює одиниці. Він в свою чергу збільшує на $\frac{1}{2}$ потенціали всіх (в тому числі і порушених) сусідніх з ним елементів по горизонталі, вертикалі і діагоналях.

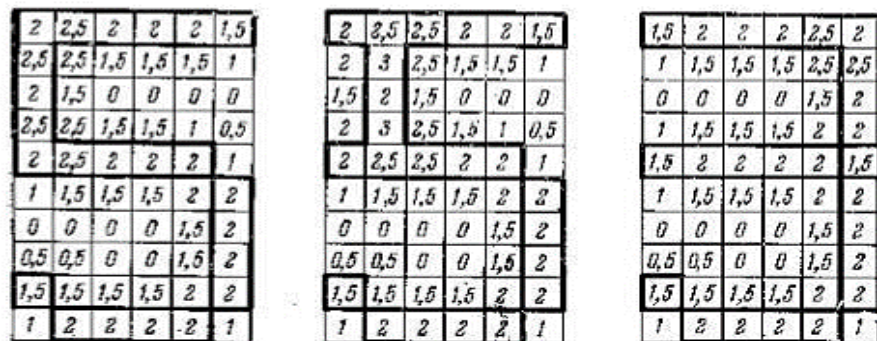


Рис. 2.6. Метод потенціалів

Найпростіший побудований на методі потенціалів алгоритм розпізнавання має два етапи:

1. Навчання. В процесі алгоритм запам'ятовує коди всіх точок, які з'явилися, а також вказівки, до якого з образів відноситься та чи інша точка.
2. Впізнавання. Відбувається ідентифікація і надається інформація до якого образу належить закодована матриця.

Незважаючи на те, що модель персептрона Розенблатта має певні обмеження вихідної форми, для багатьох сучасних алгоритмів навчання з учителем саме вона послужила основою[6]. На Рис. 2.7 показаний вид, який можна розглядати як приклад персептрона з нерекурентною мережею, що використовує навчання з вчителем. Компоненти вхідного вектора розглядаються як безперервний діапазон значень, компоненти вектора мети представлені двійковими величинами. По закінченню навчання, на вході мережа отримує набір безперервних входів, і генерує потрібний вихід, що представляє собою вектор із бінарними компонентами.

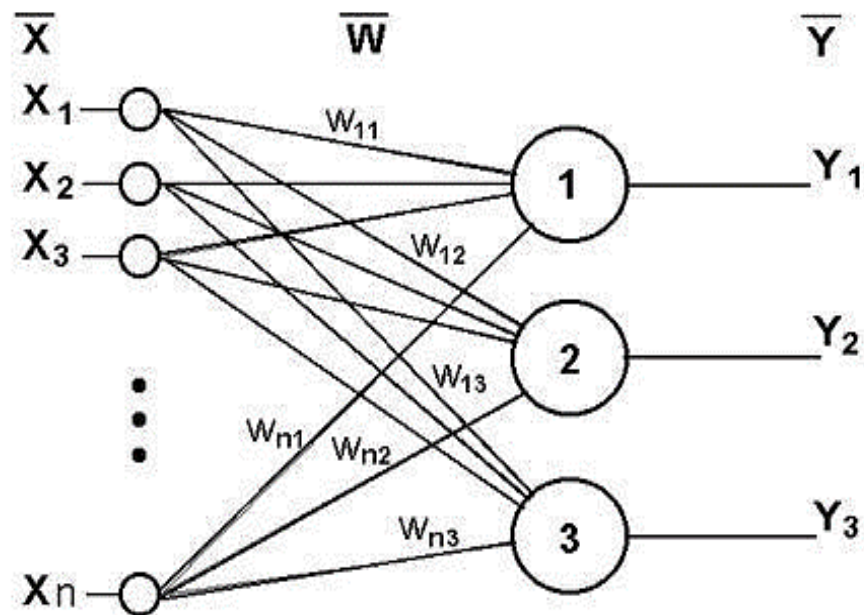


Рис. 2.7. Багатошарова нейронна мережа

Процес навчання ділиться на наступні етапи:

1. Рандомізація всіх ваг мережі в малі величини.
2. Подача на вхід мережі вхідного навчального вектора X і обчислення сигналу NET від кожного нейрона із використанням наступного виразу:

$$NET_j = \sum_i xw$$

3. Обчислення значення порогової функції активації для сигналу NET від кожного нейрона.

4. Обчислення помилки для кожного нейрона:

$$error_j = target_j - OUT_j$$

5. Модифікація ваги наступним способом:

$$W_{ij}(t + 1) = w_{ij}(t) + ax_j error_j$$

6. Повторення попередніх кроків, поки помилка не стане достатньо малою.

Перцептрон Розенблатта обмежується бінарними виходами. На випадок безперервних виходів, Уїдроу і Хоффа, із використанням сигмоїдальної функції, змогли розширити алгоритм навчання. Вчені розробили математичний доказ того, що при певних умовах мережа буде сходиться до будь-якої функції, яку можна уявити. У Адаліна, їх першої моделі, є один вихідний нейрон. У більш пізньої моделі, Мадаліні, може бути декілька вихідних нейронів.

Вирази, які описують процес навчання АДАЛІН подібні до перцептронів. Відрізняється четвертий крок. У ньому, замість бінарних OUT, використовуються безперервні сигнали NET. Змінений етап 4 для АДАЛІН виглядає так[8]:

4. За допомогою віднімання отриманого виходу з необхідного виходу відбувається обчислення помилки для кожного нейрона.

Результати досліджень Кохонена на самоорганізованих структурах для задач розпізнавання образів класифікують образи, представлені векторними величинами, в яких кожен компонент вектора відповідає елементу образу. Алгоритм Кохонена побудований на техніці навчання без вчителя. По закінченню навчання, при подачі вхідного вектору з певного класу, відбуватиметься збудження кожного вихідного нейрона. Представляти класифікацію буде нейрон із максимальним порушенням. Під час навчання без вчителя цільовий вектор не

вказується. Саме тому відсутня можливість попередньо визначити якому класу вихідних нейронів відповідатиме даний нейрон. Тим не менш, таке планування можна провести шляхом тестування мережі після проходження навчання.

Алгоритм трактує набір з n вхідних ваг нейрона як вектор в n -вимірному просторі. Кожний компонент цього вектора ваг перед початком навчання ініціалізується у випадкову величину. Далі кожний вектор нормалізується у вектор, довжина якого становить 1 символ у просторі ваг. Це відбувається шляхом ділення кожної випадкової ваги на квадратний корінь із суми квадратів компонент цього вагового вектора.

Відбувається також нормалізація всіх вхідних векторів навчального набору. Навчання мережі ділиться на такі етапи:

1. Подання вектору X на вхід мережі.
2. Визначення відстані D_j (в n -вимірному просторі) між X і ваговими векторами W_l кожного нейрона. В евклідовому просторі відстань обчислюється таким чином:

$$D_j = \sqrt{\sum_i (x - w)^2}$$

3. Той нейрон, що має найближчий до X ваговий вектор, перемагає. Даний ваговий вектор, називається W_c . Він стає основним в групі вагових векторів, що лежать в межах відстані D від W_c .

4. Відбувається налаштування групи вагових векторів у відповідності з наступним виразом:

$$W_j(t + 1) = W_j(t) + \alpha[X - W]$$

5. Повторення попередніх кроків для кожного вхідного вектора.

Значення D і α під час навчання нейронної мережі поступово зменшуються.

Коефіцієнт α на початку навчання промінь встановлюється приблизно рівним 1 і зменшуються до 0 в процесі навчання. В той же час D на початку навчання може бути рівним максимальній відстані між ваговими векторами, а в кінці може стати настільки маленьким, що буде навчатися лише один нейрон.

Існує точка зору, що додаткове навчання сприяє покращенню точності класифікації. Щоб отримати хорошу статистичну точність, Кохонен стверджує, що кількість навчальних циклів має бути щонайменше в 500 раз більше, ніж кількість вихідних нейронів.

У зв'язку з тим що всі вектори встановлюються у вектори довжиною в один символ, їх можна розглядати як точки на поверхні одиничної гіперсфери. Група сусідніх вагових точок під час навчання наближається до точки вхідного вектора.

Передбачається, що вхідні вектори збираються у класи відповідно до їх становища у векторному просторі. Той чи інший клас асоціюватиметься з певним нейроном, і буде рухати його ближче до центру. Він також сприятиме його порушенню за умови, якщо на вході з'явиться будь-який вектор цього класу. По закінченню навчання класифікація реалізується шляхом подачі до входу мережі випробуваного вектора, вирахування збудження для кожного нейрона і подальшого вибору нейрону з найбільшим збудженням. Цей нейрон і буде виконувати роль індикатора правильної класифікації.

2.2. Перцептрон, як модель розпізнавання

Нейрофізіолог Френк Розенблаг зробив великий вклад у розвиток нейрокібернетики. Він запропонував модель розпізнавання “Перцептрон”, назва якої пішла від латинського *percepto* – “розумію, пізнаю”. При розробці даної моделі вчений брав за основу певні узгоджені уявлення про структуру зорового апарату і мозку. Розенблаг хотів відтворити функції людського мозку, тому використовував просту модель біологічного нейрона (Рис. 2.8) і систему зв'язків між ними.

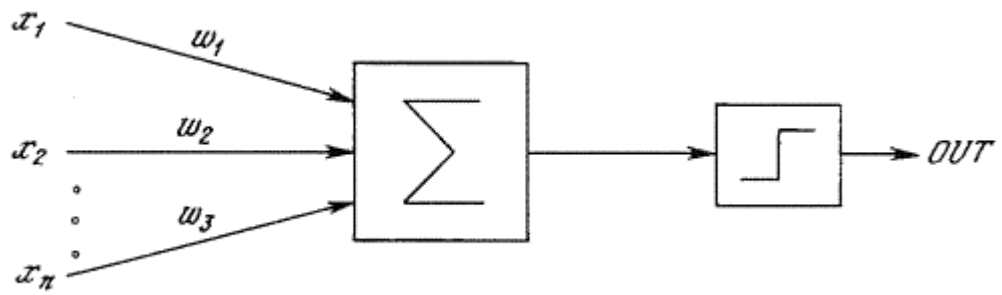


Рис. 2.8. Персептронний нейрон

В персептроні сприймаючим пристроєм служить фотоелектрична модель сітківки – це поле рецепторів, складовими частинами якого є декілька сотень фотоопорів або S-елементів (Рис. 2.9). Кожен елемент на полі рецепторів може бути у збудженому або спокійному стані. Стан залежить від того, чи падає на той чи інший фотоопір контур фігури, що спроектована на полі. На виході кожного елемента з'являється сигнал x_i ($i = 1, 2, \dots, n$, де n - число елементів). В разі якщо елемент збуджений, сигнал відповідає одиниці, якщо ж навпаки – нулю. Далі в моделі персептрона йдуть асоціативні або А-елементи. У кожного А-елементу є кілька входів і один вихід. Під час підготовки персептрона до експерименту, виходи рецепторів підключають до входів А-елементу. Причому підключити можна як зі знаком плюс, так і зі знаком мінус.

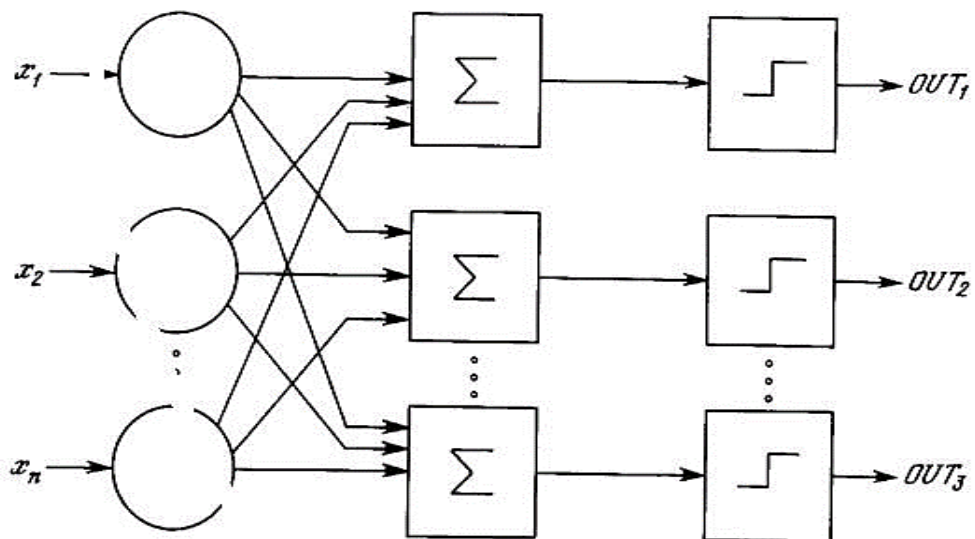


Рис. 2.9. Персептронний нейрон з багатьма виходами

Вибір рецепторів, які потрібно підключити, а також знаку підключення, відбувається рандомно. Під час експерименту зв'язок рецепторів з А-елементами не міняється. А-елементи продукують алгебраїчне підсумовування сигналів, які надійшли на їх входи, а потім порівнюють отриману суму з величиною θ , яка є однаковою для всіх А-елементів[19].

У разі коли сума більше θ , то відбувається збудження А-елементу. На виході він видає сигнал, який дорівнює одиниці. А-елемент залишається збудженим, а його вихідний сигнал рівний нулю, коли сума менше θ . Отже, вихідний сигнал j -го А-елемента, де величина g_{ij} набуває значення $+1$, якщо i -й рецептор підключений до входу j -го А-елемента зі знаком плюс; -1 , якщо рецептор підключений зі знаком мінус, і значення 0 , якщо i -ий рецептор до j -го А-елемента не може підключитися ($j = 1, 2, \dots, m$, де m – число А елементів).

За допомогою підсилювачів відбувається множення вхідних сигналів А-елементів на змінні коефіцієнти λ_j . Кожен з цих коефіцієнтів може бути позитивним, негативним або дорівнювати нулю. Кожен коефіцієнт також може змінюватися незалежно від інших коефіцієнтів. Вихідні сигнали, які виробляють підсилювачі, підсумовуються. Далі сумарний сигнал надходить до входу R-елемента. Якщо σ позитивна або дорівнює нулю, R-елемент видає на виході одиницю, а якщо σ негативна, то нуль.

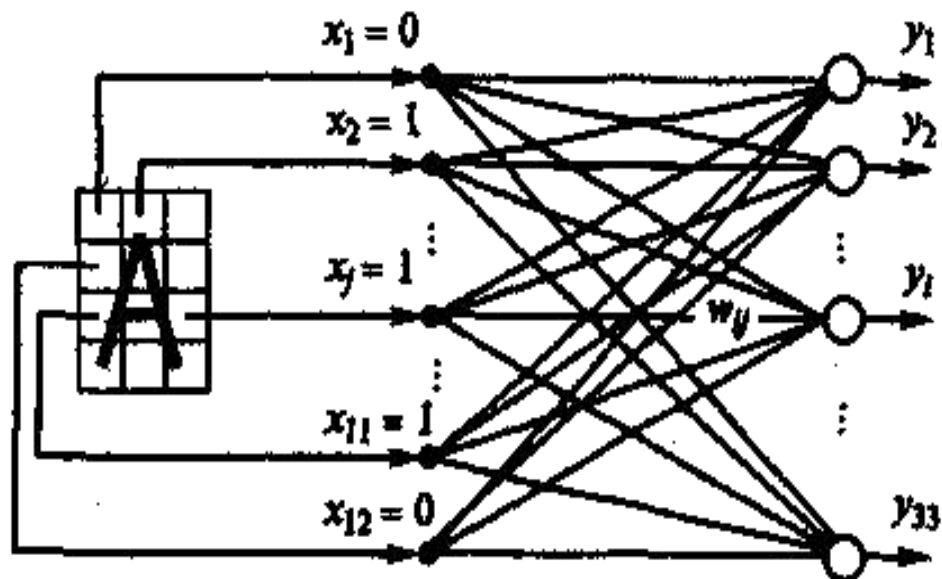


Рис. 2.10. Перцептрон з декількома виходами

Давайте припустимо, що фігури, які проектується на поле рецепторів, належать до двох різних образів. Якщо можна досягти такого стану персепторону, у якому він з достатньою надійністю видавав на виході 1 при появі на його вході фігур одного образу, то це означатиме, що персептрон може навчатися розрізненню двох образів.

Описана вище структура персептрона дозволяє розділяти об'єкти лише на дві безлічі. Якщо потрібно розпізнати більшу кількість образів, то можна застосувати персептрон, який побудований за схемою на Рис. 2.10. Вихідний сигнал А-елемента у даному випадку надходить на декілька підсилювачів, а не на один. Кількість підсилювачів відповідає кількості образів, які необхідно розрізнити. Після множення на λ вихідні сигнали надходять на суматори Σ , кількість яких також дорівнює кількості образів. Замість R-елемента встановлюється пристрій, який порівнює між собою вихідні сигнали суматорів. Наданий об'єкт відноситься до того образу, чий акумулятор має найбільший сигнал.

Для розпізнавання декількох образів може бути використаний персептрон дещо іншої структури. У такому персептроні А-елементи розбиті на кілька груп, кожна з яких пов'язана зі своїм сумматором і R-елементом. Сукупність вихідних сигналів R-елементів можна розглядати як виражений в двійковому коді номер образу, що і дає такому персептрону можливість розбивати об'єкти на кілька класів. Наприклад, для класифікації на вісім класів буде достатньо трьох груп. У цьому випадку можливі наступні вісім комбінацій вихідних сигналів трьох R-елементів: 000, 001, 010, 011, 100, 101, 110, 111. Появу кожної з цих комбінацій можна розглядати як віднесення персептроном пред'явленої йому фігури до одного з восьми образів.

Кожна з груп А-елементів, яка поєднана зі своїм R-елементом, за своєю структурою і дією повністю відповідає персептрону, який може розбивати об'єкти на два класи. Ряд послідовних тактів є складовою частиною навчання персептрона. У кожному такті персептрону пред'являється об'єкт одного з образів. Далі, залежно від того, як відреагує на фігуру персептрон, відбувається

зміна коефіцієнтів λ_j . Стає можливим за певну кінцеву кількість тактів досягти такого стану перцептрону, при якому він впевнено буде розпізнавати образи.

Навчання перцептрона може відбуватися за двома типами алгоритмів. Перший з них не враховує правильні відповіді перцептрона у той час, поки він навчається. При такому алгоритмі, λ_j в кожному такті змінюється незалежно від того, розпізнав чи не розпізнав перцептрон показаному йому в цьому такті фігуру. В другому ж алгоритмі при зміні коефіцієнтів λ_j будуть враховуватися правильні відповіді перцептрона.

Алгоритм першого типу реалізується наступним чином. Заздалегідь ставляться умови, що після навчання перцептрон повинен видавати на виході 1 встановить обґрунтованість, наприклад, об'єктів образу А, і нуль – при пред'явленні образу – В. Потім пред'являють перцептрону об'єкти кожного з образів. У кожному з тактів перцептрон, у відповідь на показаний йому об'єкт, порушує деякі А-елементи. Суть навчання полягає в тому, що коефіцієнти λ_j А-елементів, які були порушені в цьому такті, збільшуються на певну величину (наприклад, один), якщо в даному такті був показаний об'єкт образу А і навпаки зменшуються на ту ж величину, якщо був показаний об'єкт образу В.

Природно, що така зміна коефіцієнтів λ_j повинна призводити до підвищення правильності відповідей перцептрона, так як збільшення λ_j порушених А-елементів призводить до збільшення сигналу на вході R-елемента, а їх зменшення – до зменшення сигналу. Відповідно до прийнятої умови перцептрон буде давати правильні відповіді якщо образу А будуть відповідати позитивні, а образу В – негативні сигнали на вході R-елемента.

При розробці перцептрона Ф. Розенблат намагався моделювати деякі властивості живого мозку [11]. Перцептрон або будь-яка програма, що імітує процес розпізнавання, працюють в двох режимах: в режимі навчання і розпізнавання. Перш за все, на відміну від розглянутих раніше алгоритмів, алгоритм перцептрона в ході навчання не вимагає запам'ятовування пред'явлених об'єктів, а при розпізнаванні – перебору всіх "відомих" йому фігур. У цьому сенсі робота перцептрона має певну схожість з роботою мозку, який формує уявлення

про образ, не запам'ятовуючи окремих його об'єктів, і дізнається нові об'єкти без порівняння їх з кожним, що зустрічався раніше. Також структура персептрона частково схожа на структуру вищої нервової системи. Так, рецептори персептрона – це достатньо близька аналогія рецепторів зорового апарату, а А-елементи схожі на нейрони. Відомо, що нейрони можуть порушувати, якщо інтенсивність сигналу, одержуваного нейроном від пов'язаних з ним рецепторів (або інших нейронів), перевищує деяку граничну величину.

Персептрон може допускати характер зв'язків "рецептор – А-елемент". Це є аналогією деяких властивостей структури головного мозку. Цілком можливо, що деякі зв'язки між нейронами мозку також носять рандомний характер, тобто випадково варіюються у різних тварин одного біологічного виду. Якщо ж навпаки допустити, що всі зв'язки між нейронами мозку однакові у тварин одного виду, і їх зміна може стати причиною порушення роботи головного мозку, то вірним буде і припущення, що відомості про ці зв'язки повинні передаватися у спадок. Проте, так як кількість нейронів мозку обчислюється мільярдами, то таке припущення призводить до фантастично великих обсягів генетичної інформації. Разом з тим на прикладі персептрона видно, як біологічно природним є поняття компактної безлічі, бо навчання розпізнаванню таких множин виявляється можливим при випадкових зв'язках між рецепторами і нейронами.

Відомо, що мозок може зберігати або відновлювати багато своїх функцій при серйозних пошкодженнях, викликаних травмами або захворюваннями. Стійкість персептрона до порушень його структури має певну схожість з цією властивістю мозку. З усіх цих міркувань не можна зробити висновок, що алгоритми мозку і персептрона збігаються. Однак в даний час персептрон є, мабуть, найбільш правдоподібною моделлю мозку.

Найважливішою властивістю нейронних мереж є їх здатність до навчання. Як і біологічні системи, які вони моделюють, нейронні мережі моделюють самі себе з метою досягнення кращої моделі поведінки [17].

Використання критерій лінійної роздільності допоможе виявити здатність одношарової нейронної мережі до виконання потрібної опції. Навіть якщо відповідь буде позитивною, користі вона не особливо не принесе, якщо ми не матимемо спосіб знайти необхідні значення для ваг і порогів. Для того, щоб мережа мала практичну цінність, необхідний алгоритм - систематичний метод для обчислення даних значень. Розенблатт зробив це в своєму алгоритмі навчання персептрона, і разом з тим довів, що персептрон має можливість навчитися всьому, що він реалізує.

Навчання буває з учителем і без нього. Навчання з учителем потребує зовнішнього вчителя, задача якого - оцінювати поведінку системи і керувати її майбутніми модифікаціями. Під час навчання без учителя, система самостійно змінюється шляхом самоорганізації.

Алгоритм навчання персептрона можна реалізувати на цифровому комп'ютері або іншому електронному пристрої. Мережа в певному сенсі здатна сама підстроюватися. Саме тому процес підстроювання ваг прийнято називати навчанням, а про мережу кажуть, що вона навчається. Доказ Розенблатта дав потужний імпульс для розвитку досліджень у цій сфері.

Висновки до другого розділу

У другому розділі було розглянуті проблеми, з якими можна зіткнутися під час створення комп'ютерної системи розпізнавання облич на основі нейронних мереж. Проблема розпізнавання образів (ПРО) має дві частини – навчання та розпізнавання. Під час навчання системі демонструються окремі об'єкти, при цьому зазначається їх належність до того чи іншого образу. По закінченню навчання система розпізнавання повинна вміти реагувати однаковими реакціями на всі об'єкти одного образу і різними - на всі об'єкти різних образів. Важливим є те, що навчання має завершитися лише шляхом показів кінцевого числа об'єктів. З метою вирішення проблеми навчання, були розглянуті різні методи навчання штучних нейронних мереж. Виявлено, що на сьогоднішній день можливості навчання штучних нейронних мереж обмежені. Під час навчання системі демонструються вхідні вектори, і одночасно, відповідно до певної процедури, підстроюються ваги. У процесі навчання ваги мережі поступово стають такими, що кожен вхідний вектор виробляє вихідний вектор. В результаті, була обрана модель розпізнавання образів – "Персептрон", як найефективнішу серед усіх розглянутих.

РОЗДІЛ 3.
РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ
ОБЛИЧ

3.1. Аналіз поставленої задачі та її реалізація

Проблема розпізнавання образів із застосуванням методів нейрообробки інформації в наш час є досить актуальною. Стандартні схеми аналізу зображень та оцінки даних не мають гнучкості і прив'язані до певного алгоритму, який не може адаптуватися. Є безліч моделей, що розпізнають графічні образи, але вони підлаштовані під конкретний образ, і не мають можливості самоадаптуватися [17]. Це і є їх головним недоліком. Застосування альтернатив, які дозволять моделі самоналаштуватися під нові недетерміновані об'єкти, було б більш універсальним.

У даній роботі поставлене завдання можливості розпізнавання облич із використанням комплексної моделі, штучної нейронної мережі, яка розділену модель нейронів і ймовірносно-статистичний аналіз растрових масивів. Для досягнення мети було побудовано гібридну програму.

Можливість вирішення проблеми залежить від багатьох факторів. У першу чергу потрібно набрати Базу Даних (БД) представників від кожного класу. Чим краще підібрані еталонні представники, тим вище буде точність і ефективність розпізнавання символів. І навпаки, якщо в еталонну БД буде внесено погано детермінованих або спотворених представників, то класифікація образів проходитиме значно важче. А це значить, що точність розпізнавання значно знизиться.

<i>Кафедра КІТ (47)</i>				<i>НАУ 21.06.64.000 ПЗ</i>			
<i>Виконала</i>	<i>Закалата Д.Ю.</i>			<i>3. Розробка комп'ютерної системи розпізнавання облич</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Моденов Ю.Б.</i>					52	14
<i>Консульт.</i>					<i>УС-211М 122</i>		
<i>Н. Контр.</i>	<i>Райчев І.Е.</i>						

Умовою для якісного розпізнавання є достатній набір введених еталонів, під час використання яких можна проводити аналітичні звірки і в подальшому застосовувати алгоритми нейромережевого розпізнавання. Щоб еталони були самодостатніми, представники кожного символу БД повинні мати такі властивості: бути класифікованими, не на 100% бути схожими між собою, не мати всередині класів сильно спотворені елементи. В процесі розпізнавання образів, зображення подаються на матрицю рецепторів аналогічно з тим, як зображення потрапляє на сітківку ока. Зображення, яке сприймається далі надходить в мозок для подальшої переробки і розпізнавання, тобто зарахування образу до того чи іншого класу символів.

За аналогією будується модель з полем рецепторів, які представляють собою прямокутний масив, на якому можна зображувати всілякі конфігурації символів. Введення інформації в комп'ютер – це процес сканування зображення символів за допомогою периферійного пристрою - сканера, і збереження зображення у форматі графічного файлу. Іншим способом може служити створення зображення в будь-якій графічній програмі. Збережене зображення являє собою графічний файл, тобто послідовність кодових знаків, які несуть в собі інформацію про структуру візуального зображення. Однак для використання цієї інформації в математичних цілях, необхідно перетворити цей послідовний зашифрований код в більш доступне матричне подання нулів і одиниць (0 – не зафарбовані, 1 – зафарбовані точки) [17].

Існуючі на ринку програми розпізнавання текстів застосовують лексикон слів для звірки тексту з метою усунення можливих помилок, що дозволяє штучно підняти відсоток розпізнавання образів. Крім того, наявні ПО для розпізнавання образів, в основному, працюють під певні шрифти, що є ще одним недоліком і звужує коло завдань розпізнавання. Сучасні програми розпізнавання і дослідження образів також використовують матриці з високою роздільною здатністю. Чим вище дозвіл (наприклад, номінальне 120 по горизонталі на 160 по вертикалі = 19 200 точок), тим точніше можна виявити закономірності і правильно підлаштувати рекогнаційну модель.

Для отримання більш точних характеристик повинна бути введена досить велика БД еталонних елементів, що складається не менше ніж з 1 000 різновидів одного елемента. Набір еталонної БД для подальшого скрупульозного дослідження графічної структури знімку може бути достатньо трудомістким і зайняти багато часу і ресурсів. Якщо, наприклад, номінальна кількість еталонних екземплярів 1 000 для одного символу і дозвіл 19 200 точок (120x160), то обсяг інформації при 25% -вому зафарбуванні займатиме 4 800 000 точок. Якщо ми в загальному випадку маємо справу, наприклад з 100 різновидами символів (заголовні, рядкові, синонімних і ін.), то обсяг інформації буде становити 480 000 000 (480 мільйонів точок) еталонної БД. Очевидно, що таке завдання не посильне для одного дослідника і потрібен великий колектив операторів.

3.1.1. Алгоритм нейромережевої організації матриць для розпізнавання образів

По факту, завдання полягає в тому, щоб на основі 40 введених конфігурацій в БД для кожного символу виявити їх характерні риси і розпізнати зображення символу, що показаний для розпізнавання. Завдання складається з таких кроків:

1. Завантаження бази даних, яка містить 2760 введених еталонних символів з MS Excel файлу в програму.
2. Введення зображення довільного символу з метою надалі розпізнавати на матриці рецепторів (12x16) і збереження в форматі .BMP (16 Color).
3. Зчитування введеного зображення для розпізнавання з файлу в програму.
4. Перетворення зображення з двійкового представлення в табличний масив 0 і 1. 0 – це незафарбовані, а 1 – зафарбовані клітинки масиву.
5. Підрахунок конгруентної проміжної нейронної матриці 2-го шару.
6. Розрахунок пасивної проміжної нейронної матриці 2-го шару.
7. Знаходження характеристик симбіозних матриць і обчислення підсумкових значень на виходах.

8. Виявлення максимального значення з виходів симбіозних матриць, визначення його коду і асоціювання з котрі вступили на вхід зображення.

9. Коригування суматорної матриці 1-го рівня (з учителем або без вчителя) в разі високої похибки.

Блок-схема данного алгоритму представлена у Додатку Б.

Штучна нейро-матрична мережа, що самоорганізується (ІСНМС), побудована за наступною схемою (Рис. 3.1): на вхід нейронної мережі, який можна уявити як матрицю рецепторів, подається растрове зображення символу. Зображення можна представити у вигляді послідовності 0 і 1, де 0 - незафарбовані, а 1 - зафарбовані клітинки. Далі інформація з матриці рецепторів подається і трансформуються на другий рівень - клас нейронних матриць представників і критеріїв оцінок для кожного образу бази даних. На третьому рівні відбувається підрахунок проміжних ваг і застосування критеріїв відбору елементів, що також прописані у матричному вигляді.

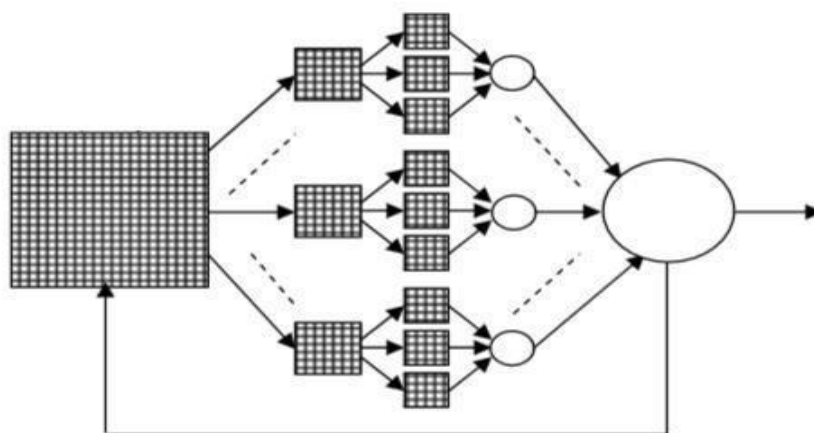


Рис. 3.1. Умовний граф матричної нейронної мережі

На наступному етапі отримані результати зважуються шляхом обчислення цільової функції кожної матричної схеми, з урахуванням різних критеріїв. На останньому етапі, після обчислення всіх ваг, отримані результати переглядаються, в найбільш наближений образ відповідатиме найвищому значенню. У створеній схемі використовується самоорганізація мережі – це значить, що після визначення способу відбувається переналаштування ваг різних

матриць критеріїв в циклі. Таким чином, схема не є статичною – вона здатна до динамічного підлаштування і ефективного сходження.

На основі принципово нової побудованої штучної нейронної схеми з незалежними нейронами і застосування ймовірнісно-статистичних аналізів були отримані високі результати розпізнавання. На наведеному графі зображено структурну схему штучної нейронної мережі, що здатна самоорганізуватися (ІСНС) для розпізнавання графічних образів.

3.2. Програмна реалізація системи розпізнавання облич

Комп'ютерна система розпізнавання графічних образів реалізована на основі Google Cloud Vision API. Дане програмне забезпечення швидко класифікує зображення на тисячі категорій (наприклад, “корабель”, “тигр”, “Пізанська вежа”), виявляє окремі об'єкти і особи в межах зображень, а також знаходить і читає надруковані слова, що містяться в зображеннях. Існує можливість створювати метадані каталогу зображень, помірно образливого змісту, або включити нові маркетингові сценарії за допомогою аналізу зображень. Зображення для аналізу завантажуються із запитом або інтегрується з бази зображень на Google Cloud Storage.

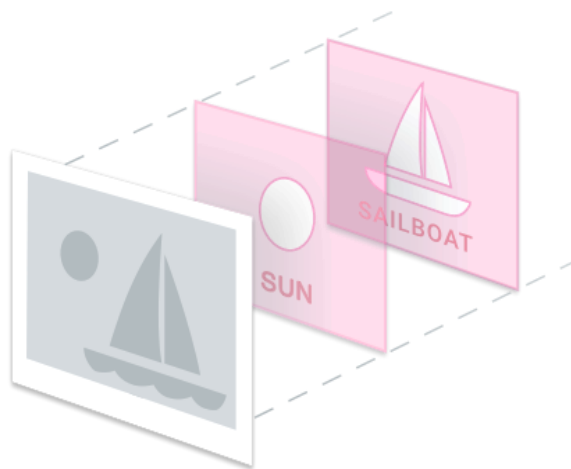


Рис. 3.2. Приклад класифікацій графічних образів

За допомогою даної комп'ютерної системи легко виявити об'єкти на зображенні, наприклад дерева, комахи, або тисячі інших категорій об'єктів, які зазвичай зустрічаються в зображеннях. VisionAPI з плином часу, вводить нові поняття і точність підвищується.

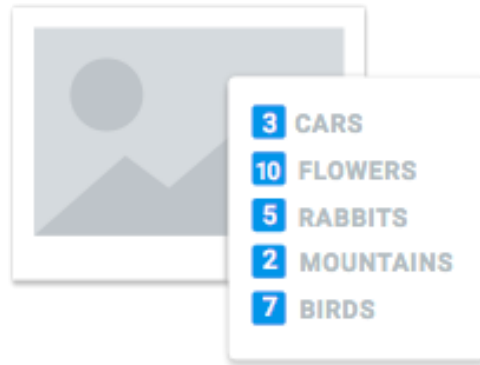


Рис. 3.3. Приклад виявлення графічних об'єктів

Розроблена комп'ютерна програма складається з наступних модулів:

1. Label Detection. Виявляє широкі набори категорій об'єктів в межах зображення, починаючи з видів транспорту до тварин.
2. Explicit Content Detection. Виявлення явного змісту на зображенні.
3. Logo Detection. Виявлення популярних логотипів продукту в межах зображення.
4. Landmark Detection. Виявлення популярних природних і штучних споруд в межах зображення.
5. Optical Character Recognition. Виявлення і вилучення тексту всередині зображення, з підтримкою широкого спектру мов, поряд з підтримкою автоматичної ідентифікації мови.
6. Face Detection. Виявлення декількох осіб в межах зображення, поряд з такими ключовими атрибутами особи, як емоційний стан або носіння головних уборів. Розпізнавання особи не підтримується.
7. Image Attributes. Виявлення загальних атрибутів зображення, таких як домінуючий колір.

Для використання Cloud API необхідно налаштувати облікові дані для вашого програмного продукту для перевірки автентичності особистості до певного API та отримати дозвіл на виконання завдання. (Ці облікові дані, пов'язані з механізмами схеми авт.). При отриманні доступу до API Google Cloud Platform, рекомендовано встановити ключ API для тестування, а також створити обліковий запис служби для використання продукції.

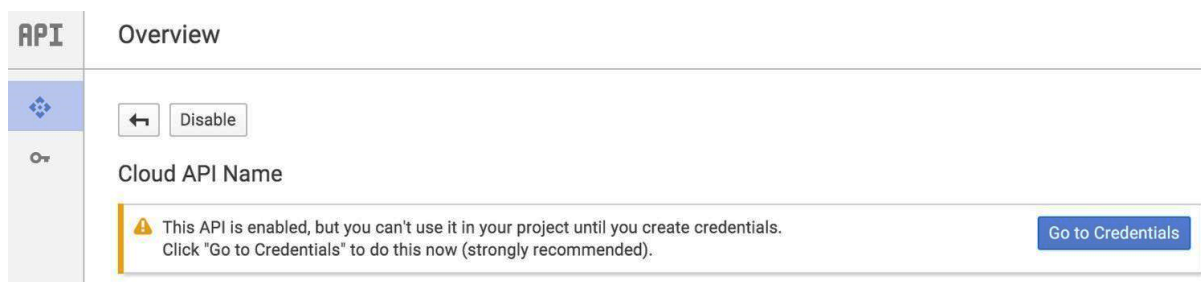


Рис. 3.4. Налаштування проекту

Для додавання облікових даних необхідно натиснути кнопку “Go to Credentials”.

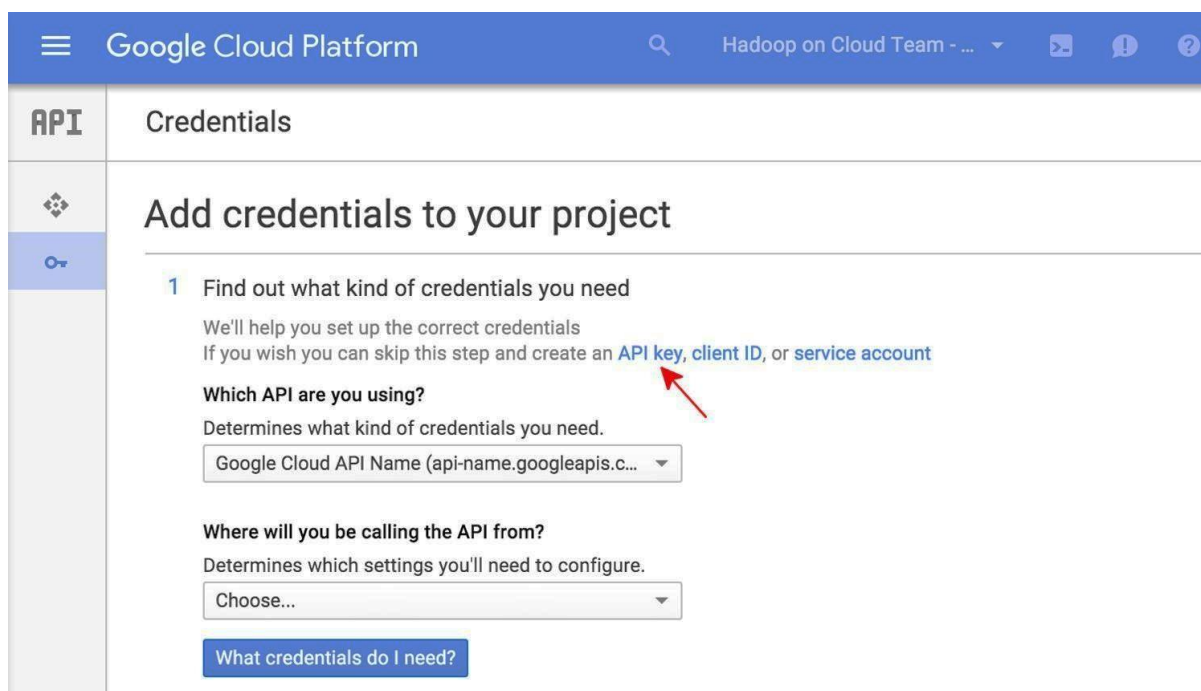


Рис. 3.5. Додавання облікових даних

Ауθενфікація і авторизація на Google Cloud Platform API зазвичай виконується за допомогою облікового запису служби. Програма попросить ввести код, щоб відправити дані програми безпосередньо на Cloud API. Службовий обліковий запис – це обліковий запис користувача, який представляє собою адресу електронної пошти. На відміну від облікового запису користувача, службовий обліковий запис відноситься тільки до додатку, і може бути використаний для доступу до API, для якого він був створений. В якості прикладу, ми покажемо, як створити обліковий запис служби за допомогою консолі платформи Google Cloud.

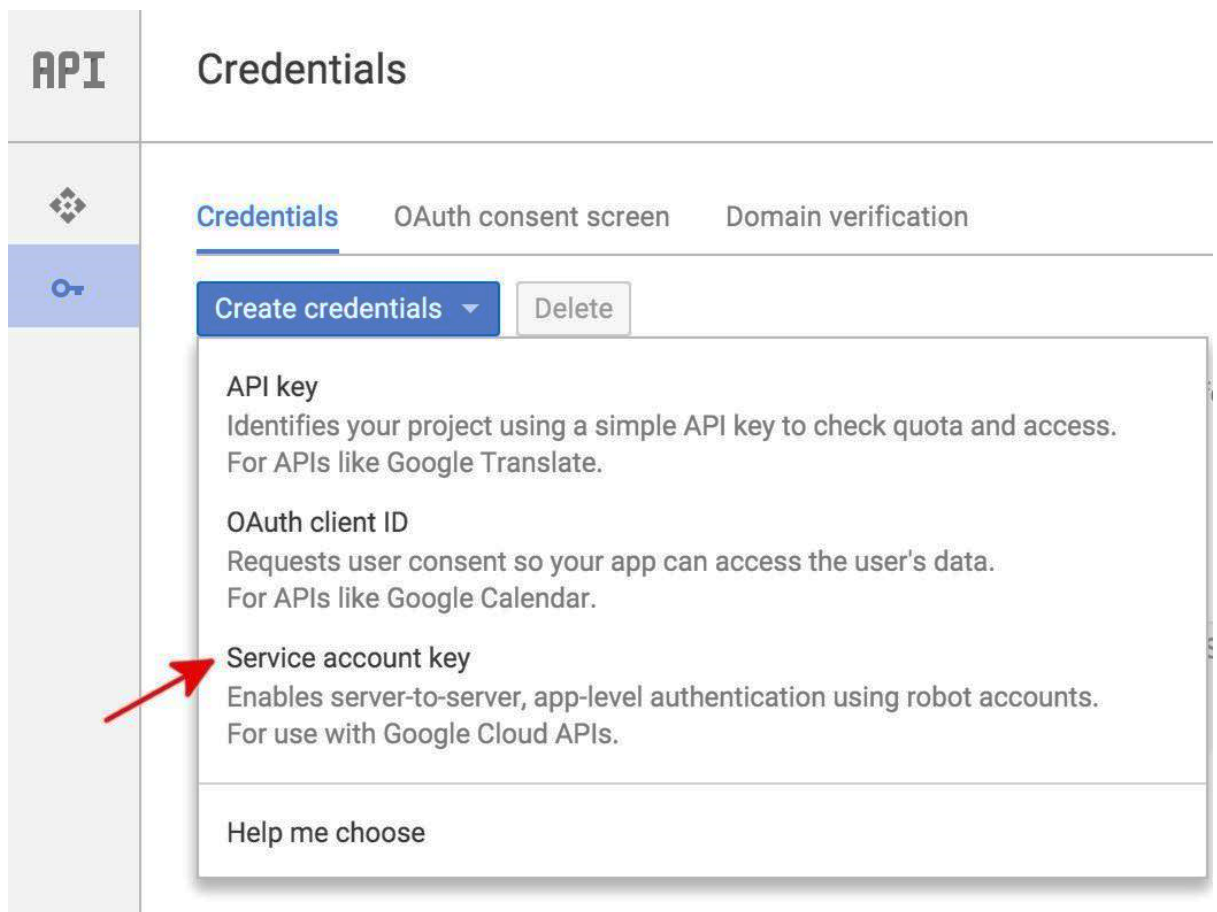


Рис. 3.6. Створення API ключа

Розглянемо реалізацію модулів комп'ютерної системи. Label Detection здатний виявити широкі набори категорій в межах зображень – від видів транспорту до тварин. Далі можна побачити приклад виявлення міток у локальному файлі зображення. Звернення до `images:annotate` відбувається через

API endpoint. Щоб провести виявлення мітки, необхідно зробити запит POST і подати до відповідного органу запиту:

```
POST
https://vision.googleapis.com/v1/images:annotate?key=YOUR_API_KEY
{ "requests": [
  { "images": {
    "content": "/9j/7QBEUGhvdG9zaG9...base64-encoded-
imagecontent...fXNWzvDEeYxxxzj/Coa6Bax/Z"
  }, "features": [
    { "type": "LABEL_DETECTION" } ] } ] }
```

Реалізація Label Detection через клієнську бібліотеку виглядає так:

```
useGoogle\Cloud\Vision\VisionClient;
// $projectId = 'YOUR_PROJECT_ID';
// $path = 'path/to/your/image.jpg'
$vision =newVisionClient([
    'projectId'=> $projectId,
]);
$image = $vision-
>image(file_get_contents($path),['LABEL_DETECTION']);
$result = $vision->annotate($image);
print("LABELS:\n");
foreach($result->labels()as $label){ print($label->description(). PHP_EOL);}
```

Давайте розглянемо приклад реалізації Face Detection. Для отримання доступу до APIGoogle, за допомогою офіційного клієнта SDKs, потрібно створити об'єкт сервісу, заснований на відкритті документа API, який описує API в SDK. Необхідно отримати його від служби Vision API's, використовуючи свої облікові дані:

```
useGoogle\Cloud\Vision\VisionClient;
// $projectId = 'YOUR_PROJECT_ID';
// $path = 'path/to/your/image.jpg'
$vision =newVisionClient([
    'projectId'=> $projectId,
]);
```

Щоб створити запит до API бачення, насамперед необхідно звернутися до документації по API. У цьому випадку, ви будете давати запит до ресурсу зображень на дозвіл коментувати зображення. Запит на цей API приймає форму об'єкта зі списком запитів. Кожен елемент цього списку містить два біта інформації:

```
$image = $vision-
>image(file_get_contents($path),['FACE_DETECTION']);
$result = $vision->annotate($image);
```

Анотація, як відповідь на наш запит, буде включати в себе велику кількість метаданих про виявлених осіб, серед яких будуть координати охоплюючого обличчя багатокутника. Поки це виглядає лише як список номерів. Їх використовують для підтвердження того, що програма дійсно знайшла обличчя на зображенні.

3.3. Аналіз отриманих результатів

Комп'ютерна система, за допомогою використання нейронних мереж, може розпізнавати на графічних зображеннях об'єкти різного типу. Реалізовано розпізнавання тексту, кольорів, облич тощо.

Розглянемо розпізнавання об'єктів на графічному Рис.3.7.



Рис. 3.7. Вихідне графічне зображення для тестування

Результат виконання розпізнавання комп'ютерною системою виглядатиме відповідно до Рис. 3.8.

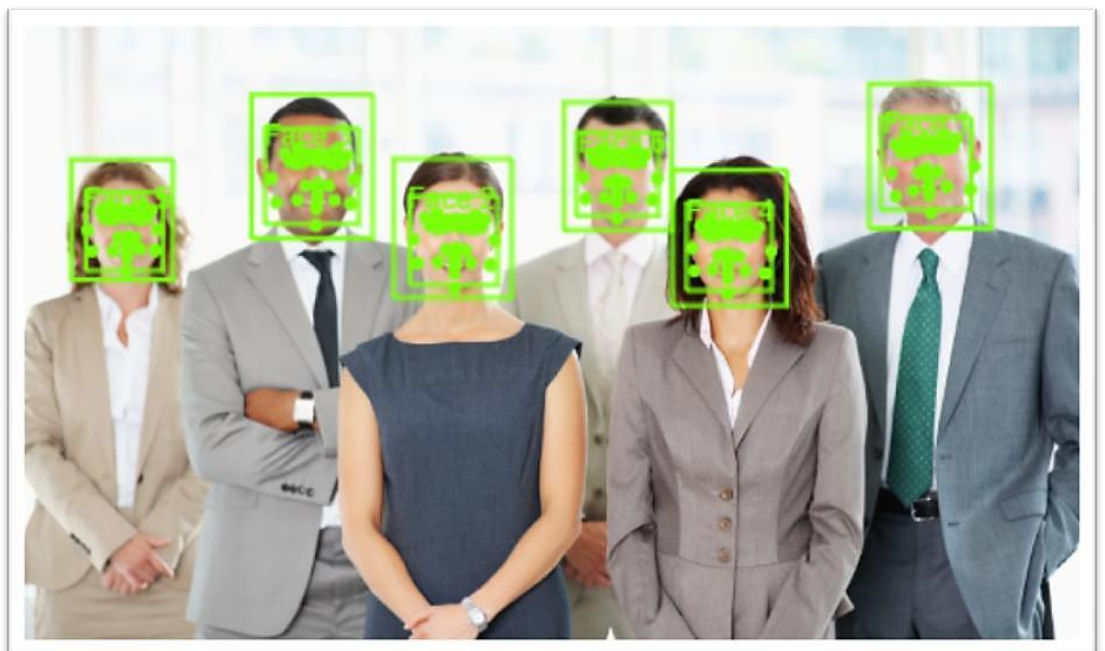


Рис. 3.8. Вихідна фотографія після обробки КС

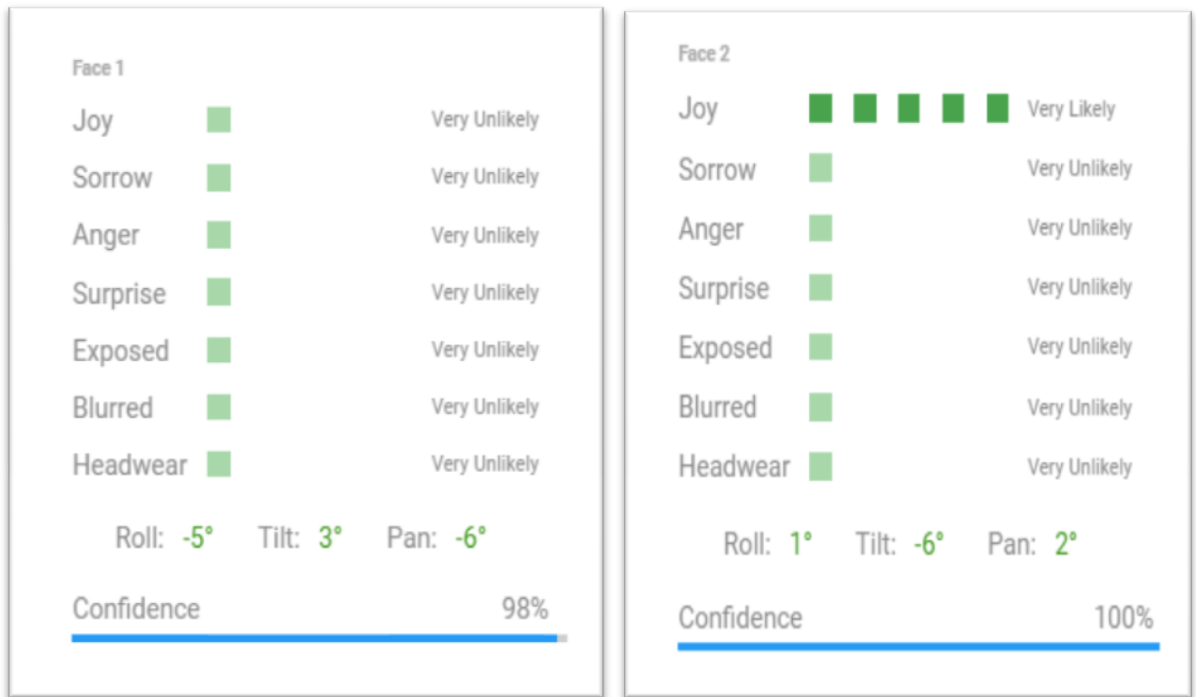


Рис. 3.9. Результат розпізнавання емоцій на двох обличчях

Дане графічне зображення модуль Label Detection позначив міткою “Team”. Це є найбільш очевидним результатом при визначенні цього зображення.



Рис. 3.10. Результат додавання міток до даного графічного зображення



Рис. 3.11. Результат розпізнавання кольорів

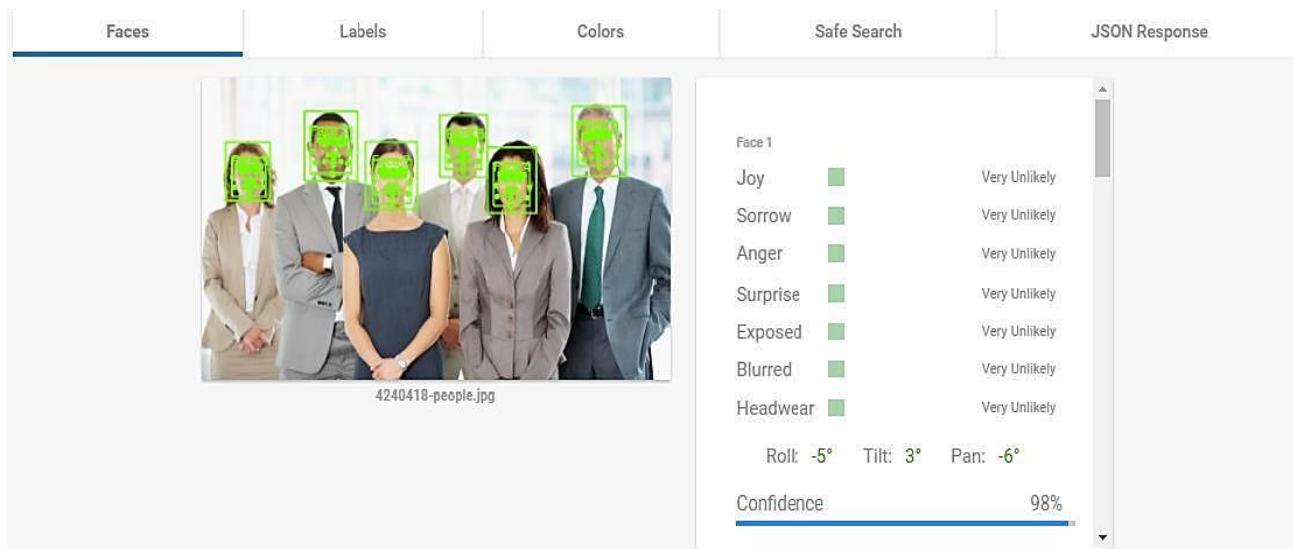


Рис. 3.12. Інтерфейс комп'ютерної системи

Згідно з отриманими результатами, можна зробити висновок, що ця система працює повноцінно і повністю відповідає поставленим перед створенням задачам. Системі має значно підвищену швидкодію розпізнавання графічних образів, а також суттєво знижені апаратні затрати.

Висновки до третього розділу

В даному розділі була створена програма на основі Google Cloud Vision API та модуля Face Detection. На основі побудованої принципово нової штучної нейронної схеми з незалежними нейронами і застосування ймовірнісно-статистичних аналізів були отримані високі результати розпізнавання. Наведений граф демонструє структурну схему побудованої штучної нейронної мережі для розпізнавання образів, що самоорганізується (ІНС).

Дана комп'ютерна система розпізнавання облич та графічних образів, шляхом використання нейронних мереж, дозволяє розпізнавати об'єкти різного типу на графічних зображеннях. Було реалізовано розпізнавання позначок, тексту, кольорів, облич тощо. Спираючись на виконані тести, можна ствердити, що розроблена комп'ютерна система відповідає поставленим вимогам.

ВИСНОВКИ

На основі проведених досліджень запропоновано принципово новий метод розпізнавання образів за допомогою неординарного алгоритму і програмної реалізації, здійснює функціонування нової розділеної структури штучної нейронної мережі (ІНС), що використовує нейро-матричну реалізацію (конгруентність, диверсивних, пасивних і нейтральних масивів растрових крапок), із застосуванням ймовірностно-статистичного аналізу для класифікації оброблюваних зображень по спеціальних мітках.

На основі аналізу і обробки спеціальної літератури та ресурсів Інтернету 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, теорії множин, дискретної математики, кодування / декодування, структури графічних форматів файлів, теорії ймовірності та математичної статистики, алгоритмічних побудов блок-схем і модульного об'єктно-орієнтованого програмування (ООП) на мові PHP.

У роботі створена принципово нова штучна нейро-матрична мережу (ІНМС) без вчителя, яка обробляє і модифікує еталонну базу образів - суматорних матрицю при порівнянні з поданими на вхід тестовими зображеннями. Підстроювання ведеться посимвольний, змінюючи вагові коефіцієнтний характеристики двомірних масивів з БД, і застосовує нові методи адаптації для підвищення ефективності розпізнавання образів. Основою нового методу є ймовірностно-статистичний фундамент оцінок в якості вагових коефіцієнтів для багатосарової мережі і застосування нової ідеї змішування і оцінок піксельних мас для виявлення дискретної приналежності до класифікованих груп.

Результати експериментальних досліджень досягнення високих результатів класифікації при мінімально можливому дозволі матриці рецепторів і кількості представників образів з кожного набору символьних елементів бази

даних, підтверджують ефективність принципово нового розробленого методу вагових параметрів синаптичних масивів для навчання ІНС, мають практичну цінність і може бути впроваджено в різних областях науки і практики.

Змодельована, протестована і налагоджена комп'ютерна програма з принципово нової комплексної структурної моделі ІНС, що використовує ймовірно-статистичний аналіз нейро-матричних масивів графічної інформації з мульти-розділеною мережею нейронів, для паралельних процесів класифікації образів.

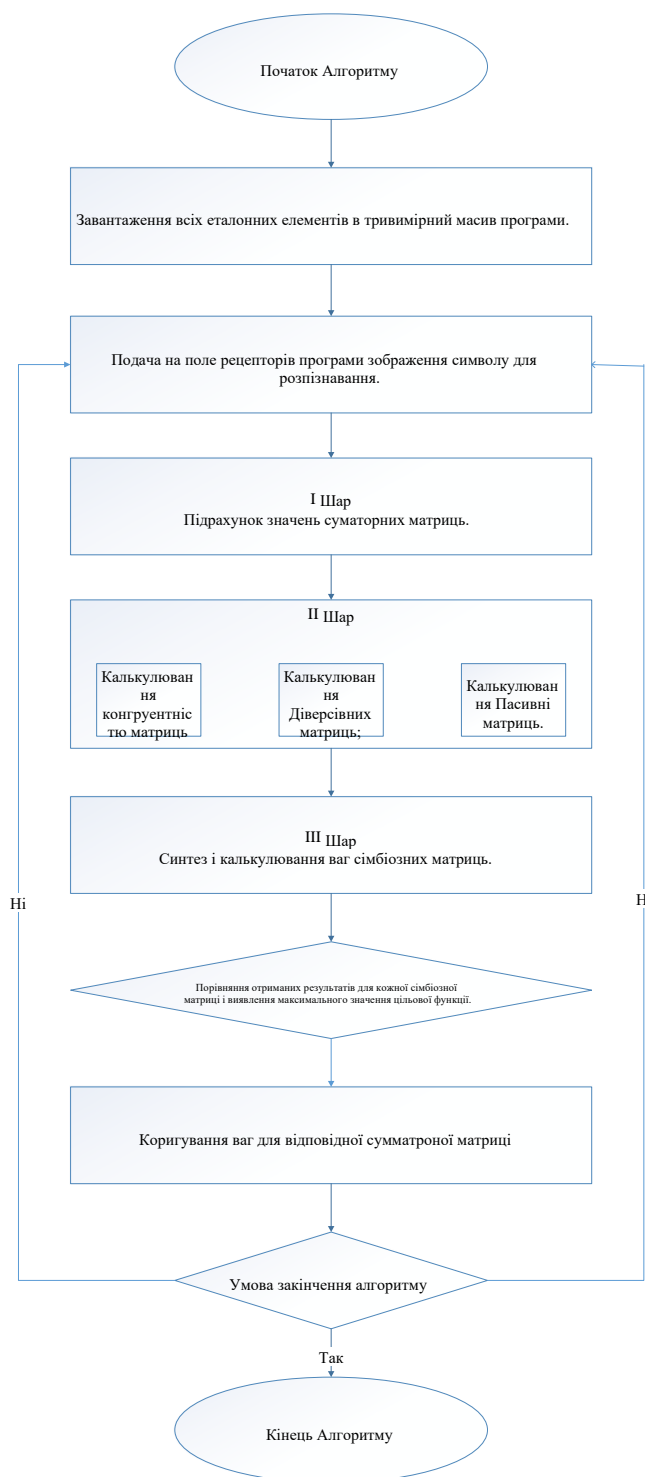
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Анализ существующих подходов к распознаванию лиц [Электронный ресурс] – режим доступа: <https://habr.com/ru/company/synesis/blog/238129/>.
2. Алгоритмы обнаружения лица человека для решения прикладных задач анализа и обработки изображений. URL: <http://tekhnosfera.com/algoritmyobnaruzheniya-litsa-cheloveka-dlya-resheniya-prikladnyh-zadach-analiza-iobrabotki-izobrazheniy>
3. Васильев А. А. Оптический операционный усилитель для обработки изображения / А. А. Васильев // II Всесоюзная конференция по оптической обработке информации. – Фрунзе : Илим, 1990. – С. 69-70.
4. Головкин В.А. Нейроинтеллект: Теория и применения. Книга 1. Организация и обучение нейронных сетей с прямыми и обратными связями. – Брест: БПИ, 1999. – 260с.
5. 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.
6. Распознавание лиц на групповых фотографиях с использованием алгоритмов сегментации: Франц В.А., Левина О.М., Воронин В.В., Кожин Р.А. Первичная обработка карты глубины изображения. Успехи современной радиоэлектроники. Радиотехника – режим доступа: <https://cyberleninka.ru/article/n/raspoznvanie-lits-na-gruppovyh-fotografiyah-sispolzovaniem-algoritmov-segmentatsii/viewer>.
7. Разработка и анализ алгоритмов детектирования и классификации объектов на основе методов машинного обучения. URL: <https://www.dissercat.com/content/razrabotka-i-analiz-algoritmov-detektirovaniya-iklassifikatsii-obektov-na-osnove-metodov-ma>
8. Фу К. Последовательные методы в распознавании образов и обучении машин / К. Фу. – М. : Наука, 1971. – 255 с.

9. Overview of the Face Recognition Grand Challenge.
URL: <https://ieeexplore.ieee.org/document/1467368>
10. DeepFace Closing the Gap to Human-Level Performance in Face Verification. URL:
https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf
11. Design, Implementation and Evaluation of Hardware Vision Systems dedicated to Real-Time Face Recognition. URL:
https://www.researchgate.net/publication/221786186_Design_Implementation_and_Evaluation_of_Hardware_Vision_Systems_Dedicated_to_Real-Time_Face_Recognition
12. Face Detection A Survey. URL:
<https://www.cin.ufpe.br/~rps/Artigos/Face%20Detection%20-%20A%20Survey.pdf>
13. Image-based Face Recognition — Issues and Methods. URL: https://www.face-rec.org/interesting-papers/general/chapter_figure.pdf
14. Face Recognition: A Convolutional Neural-Network Approach: IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 8, NO. 1, JANUARY 1997 – режим доступа:
<https://pdfs.semanticscholar.org/ee1e/322b5f8f15ad3fcd17762fba3da209b0c484.pdf>.
15. Face Recognition by Elastic Bunch Graph Matching: CRC Press, ISBN 08493-2055-0, Chapter 11, pp. 355-396, (1999) – режим доступа:
https://www.facerec.org/algorithms/EBGM/WisFelKrue99_FaceRecognition-JainBook.pdf.
16. The FERET evaluation methodology for face-recognition algorithms. URL:
<https://ieeexplore.ieee.org/document/879790>
17. Towards a Practical Face Recognition System Robust Alignment and Illumination by Sparse Representation. URL:
https://www.researchgate.net/publication/224240923_Toward_a_Practical_Face_Recognition_System_Robust_Alignment_and_Illumination_by_Sparse_Representation

18. Chen C. H. Handbook of pattern recognition and computer vision / C. H. Chen, L. F. Rau // Singapore-New Jersey-London-Hong Kong : World Scientific Publishing Co. Pte. Ltd., 1995. – P. 256-258.
19. Overview of the Face Recognition Grand Challenge. URL: <https://ieeexplore.ieee.org/document/1467368>
20. Face Recognition Vendor Test (FRVT) [Электронный ресурс] – режим доступа: <https://www.nist.gov/programs-projects/face-recognition-vendor-test-frvt>.

Блок схема нейронної мережі для розпізнавання образів



Лістинг коду комп'ютерної системи

DetectFaceCommand.php

```
use Symfony\Component\Console\Command\Command;
use
Symfony\Component\Console\Input\InputArgument;
use Symfony\Component\Console\Input\InputOption;
use
Symfony\Component\Console\Input\InputInterface;
use
Symfony\Component\Console\Output\OutputInterface
;

/**
 * Command line utility to detect which language some
text is written in. */
class DetectFaceCommand extends Command
{
    private $imageCreateFunc = [
'png' => 'imagecreatefrompng',
'gd' => 'imagecreatefromgd',
    'gif' => 'imagecreatefromgif',
    'jpg' => 'imagecreatefromjpeg',
    'jpeg' => 'imagecreatefromjpeg',
    ];
    private $imageWriteFunc = [
    'png' => 'imagepng',
    'gd' => 'imagegd',
    'gif' => 'imagegif',
    'jpg' => 'imagejpeg',
    'jpeg' => 'imagejpeg',
    ];
    protected function configure()
    {
        $this
            ->setName('face')
            ->setDescription('Detect faces in an image using
,
                . 'Google Cloud Vision API')
```



```
        ->setHelp(<<<EOF
The <info>%command.name%</info> command finds faces in an
image using the Google Cloud Vision API.
```

```
<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>
```

```
EOF
```

```
        )
        ->addArgument(
            'path',
            InputArgument::REQUIRED,
            'The image to examine.'
        )
        ->addArgument(
            'output',
            InputArgument::OPTIONAL,
            'The output image with bounding boxes.'
        )
        ->addOption(
            'project',
            'p',
            InputOption::VALUE_REQUIRED,
            'Your Project ID.'
        )
    };
}

protected function execute(InputInterface $input,
OutputInterface $output)
{
    $projectId = $input->getOption('project');
    $path = $input->getArgument('path');
    if (preg_match('/^gs:\/\/\/([a-z0-9\._\-\-]+)\/(\S+)\/$',
    $path, $matches)) {
        list($bucketName, $objectName) =
array_slice($matches,
1);
        $result = require __DIR__ .
'/snippets/detect_face_gcs.php';
    } else {
        $result = require __DIR__ .
'/snippets/detect_face.php';
    }
}
```

```

    }
if (
    isset($result->info()['faceAnnotations'])
&& $outFile = $input->getArgument('output')
) {
    copy($path, $outFile);
    $ext = strtolower(pathinfo($path,
PATHINFO_EXTENSION));
    if (!in_array($ext, array_keys($this-
>imageCreateFunc))) {
        throw new \Exception('Unsupported
image extension');
    }
    $outputImage = call_user_func($this-
>imageCreateFunc[$ext], $outFile);
# [START highlight_image]
    foreach ($result->info()['faceAnnotations'] as
$annotation) {
        if (isset($annotation['boundingPoly'])) {
            $vertices =
$annotation['boundingPoly']['vertices'];
            $x1 = isset($vertices[0]['x']) ?
$vertices[0]['x'] : 0;
            $y1 = isset($vertices[0]['y']) ?
$vertices[0]['y'] : 0;
            $x2 = isset($vertices[2]['x']) ?
$vertices[2]['x'] : 0;
            $y2 = isset($vertices[2]['y']) ?
$vertices[2]['y'] : 0;
            imagerectangle($outputImage, $x1, $y1,
$x2,
$y2, 0x00ff00);
        }
    }
# [END highlight_image]
    call_user_func($this->imageWriteFunc[$ext],
$outputImage, $outFile);
    printf('Output image written to %s' . PHP_EOL,
$outFile);
}
}
}

```

DetectImagePropertyCommand.php

```
use Symfony\Component\Console\Command\Command;
use
Symfony\Component\Console\Input\InputArgument;
use Symfony\Component\Console\Input\InputOption;
use
Symfony\Component\Console\Input\InputInterface;
use
Symfony\Component\Console\Output\OutputInterface
;

/**
 * Command line utility to detect which language some
text is written in. */
class DetectImagePropertyCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('property')
            ->setDescription('Detect image proerties in an
image using '
                . 'Google Cloud Vision API')
            ->setHelp(<<<EOF
The <info>%command.name%</info> command detects image
properties in an image using the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
                )
            ->addArgument(
                'path',
                InputArgument::REQUIRED,
                'The image to examine.'
            )
            ->addOption(
                'project',
                'p',
                InputOption::VALUE_REQUIRED,
                'Your Project ID.'
            )
    }
}
```

```

        )
        ;
    }

    protected function execute(InputInterface $input,
OutputInterface $output)
    {
        $projectId = $input->getOption('project');
        $path = $input->getArgument('path');
        if (preg_match('/^gs:\/\/\/([a-z0-9\._\-\-]+)\/(\S+)$/ ',
        $path, $matches)) {
            list($bucketName, $objectName) =
array_slice($matches,
1);
            $result = require __DIR__ .
'/snippets/detect_image_property_gcs.php';
        } else {
            $result = require __DIR__ .
'/snippets/detect_image_property.php';
        }
    }
}

```

DetectLabelCommand.php

```

use Symfony\Component\Console\Command\Command;
use
Symfony\Component\Console\Input\InputArgument;
use Symfony\Component\Console\Input\InputOption;
use
Symfony\Component\Console\Input\InputInterface;
use
Symfony\Component\Console\Output\OutputInterface
;

/**
 * Command line utility to detect which language some
text is written in. */
class DetectLabelCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('label')

```

```

        ->setDescription('Detect labels in an image using
,
        . 'Google Cloud Vision API')
        ->setHelp(<<<EOF
The <info>%command.name%/info> command labels objects seen
in an image using the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
        )
        ->addArgument(
            'path',
            InputArgument::REQUIRED,
            'The image to examine.'
        )
        ->addOption(
            'project',
            'p',
            InputOption::VALUE_REQUIRED,
            'Your Project ID.'
        );
    }

    protected function execute(InputInterface $input,
OutputInterface $output)
    {
        $projectId = $input->getOption('project');
        $path = $input->getArgument('path');
        if (preg_match('/^gs:\/\/\/([a-z0-9\._\-\-]+)\/(\S+)\$/',
$path, $matches)) {
            list($bucketName, $objectName) =
array_slice($matches,
1);
            $result = require __DIR__ .
'/snippets/detect_label_gcs.php';
        } else {
            $result = require __DIR__ .
'/snippets/detect_label.php';
        }
    }
}

```

DetectLandmarkCommand.php

```
use Symfony\Component\Console\Command\Command;
use
Symfony\Component\Console\Input\InputArgument;
use Symfony\Component\Console\Input\InputOption;
use
Symfony\Component\Console\Input\InputInterface;
use
Symfony\Component\Console\Output\OutputInterface
; /**
 * Command line utility to detect which language some
text is written in. */
class DetectLandmarkCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('landmark')
            ->setDescription('Detect landmarks in an image
using '
                . 'Google Cloud Vision API')
            ->setHelp(<<<EOF
The <info>%command.name%</info> command landmarks objects
seen in an image using the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
        )
        ->addArgument(
            'path',
            InputArgument::REQUIRED,
            'The image to examine.'
        )
        ->addOption(
            'project',
            'p',
            InputOption::VALUE_REQUIRED,
            'Your Project ID.'
        )
    }
};
```

```

    }

    protected function execute(InputInterface $input,
OutputInterface $output)
    {
        $projectId = $input->getOption('project');
        $path = $input->getArgument('path');
        if (preg_match('/^gs:\:\/\/([a-z0-9\._\-\-]+)\\/(\S+)$\/',
        $path, $matches)) {
            list($bucketName, $objectName) =
array_slice($matches,
1);
            $result = require __DIR__ .
'/snippets/detect_landmark_gcs.php';
        } else {
            $result = require __DIR__ .
'/snippets/detect_landmark.php';
        }
    }
}

```

DetectLogoCommand.php

```

use Symfony\Component\Console\Command\Command;
use
Symfony\Component\Console\Input\InputArgument;
use Symfony\Component\Console\Input\InputOption;
use
Symfony\Component\Console\Input\InputInterface;
use
Symfony\Component\Console\Output\OutputInterface
;

/**
 * Command line utility to detect which language some
text is written in. */
class DetectLogoCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('logo')
            ->setDescription('Detect logos in an image using
,

```

```

        . 'Google Cloud Vision API')
        ->setHelp(<<<EOF
The <info>%command.name%/info> command logos objects seen
in an image using the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
    )
    ->addArgument(
        'path',
        InputArgument::REQUIRED,
        'The image to examine.'
    )
    ->addOption(
        'project',
        'p',
        InputOption::VALUE_REQUIRED,
        'Your Project ID.'
    )
    ;
}

protected function execute(InputInterface $input,
OutputInterface $output)
{
    $projectId = $input->getOption('project');
    $path = $input->getArgument('path');
    if (preg_match('/^gs:\/\/\/([a-z0-9\._\-\-]+)\/(\S+)\/$',
$path, $matches)) {
        list($bucketName, $objectName) =
array_slice($matches,
1);
        $result = require __DIR__ .
'/snippets/detect_logo_gcs.php';
    } else {
        $result = require __DIR__ .
'/snippets/detect_logo.php';
    }
}
}

```


DetectSafeSearchCommand.php

```
use Symfony\Component\Console\Command\Command;
use
Symfony\Component\Console\Input\InputArgument;
use Symfony\Component\Console\Input\InputOption;
use
Symfony\Component\Console\Input\InputInterface;
use
Symfony\Component\Console\Output\OutputInterface
;

/**
 * Command line utility to detect which language some
text is written in. */
class DetectSafeSearchCommand extends Command
{
    protected function configure()
    {
        $this
            ->setName('safe-search')
            ->setDescription('Detect adult content in an
image using '
                . 'Google Cloud Vision API')
            ->setHelp(<<<EOF
The <info>%command.name%</info> command detects adult
content in an image using the Google Cloud Vision API.

<info>php %command.full_name% -k YOUR-API-KEY
path/to/image.png</info>

EOF
        )
        ->addArgument(
            'path',
            InputArgument::REQUIRED,
            'The image to examine.'
        )
        ->addOption(
            'project',
            'p',
            InputOption::VALUE_REQUIRED,
            'Your Project ID.'
        )
    }
}
```

```

        )
    ;
}

protected function execute(InputInterface $input,
OutputInterface $output)
{
    $projectId = $input->getOption('project');
    $path = $input->getArgument('path');
    if (preg_match('/^gs:\\/\\/([a-z0-9\\.\\-]+)\\/(\S+)$/ ',
    $path, $matches)) {
        list($bucketName, $objectName) =
array_slice($matches,
1);
        $result = require __DIR__ .
'/snippets/detect_safe_search_gcs.php';
    } else {
        $result = require __DIR__ .
'/snippets/detect_safe_search.php';
    }
}
}

```