

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Нейромережева система ідентифікації персоналу підприємства
(тема)

Виконав: студент 2 курсу, гр. КТРСм-19-1

Паскарюк Д.О.

(прізвище, ініціали)

Спеціальність

151 Автоматизація та комп'ютерно-інтегровані технології

освітньої програми Комп'ютеризовані та робототехнічні системи

(код і повна назва)

Тип програми

освітньо-професійна

(повна назва освітньої програми)

Керівник

проф. Сінотін А.М.

(посада, прізвище, ініціали)

Допускається до захисту
зав. кафедри

(підпис)

Невлюдов І.ІІІ.

(прізвище, ініціали)

2020

Харківський національний університет радіоелектроніки

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми	освітньо-професійна
Освітня програма	Комп'ютеризовані та робототехнічні системи (код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Паскарюку Денису Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Нейромережева система ідентифікації персоналу підприємства», затверджена наказом по університету від 02.11. 2020 р. № 1509 Ст.
2. Термін подання студентом роботи до екзаменаційної комісії 11.12. 2020 р.
3. Вихідні дані до роботи: вимоги до програмного забезпечення для ідентифікації персоналу підприємства, апаратне забезпечення – персональний комп'ютер з операційними системами Windows, MacOS та Linux, наявність камери, доступ до мережі Internet.
4. Перелік питань, що потрібно опрацювати в роботі
 - 4.1 Вступ
 - 4.2 Аналіз існуючих нейронних мереж ідентифікації об'єктів
 - 4.3 Методи фільтрації візуальної інформації
 - 4.4 Поняття штучної нейронної мережі
 - 4.5 Методика навчання нейромережевих систем
 - 4.6 Типи навчання нейромережевих систем
 - 4.7 Багатошарова нейронна мережа
 - 4.8 Розробка архітектури нейронної мережі
 - 4.9 Види архітектур нейронної мережі
 - 4.10 Розробка архітектури нейронної мережі
 - 4.11 Проектування системи ідентифікації
 - 4.12 Нормалізація нейронної мережі
 - 4.13 Висновки та перелік джерел посилання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.ppt) – 10 с., формату А4
6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	01.11.2019 р.	Виконано
2	Аналіз літератури з нейромережових систем розпізнавання	07.11.2020 р.	Виконано
3	Аналіз засобів програмного моделювання системи розпізнавання рухів людини	15.11.2020 р.	Виконано
4	Розробка програмного забезпечення для розпізнавання рухів людини	24.11.2020 р.	Виконано
5	Розробка розділу із безпеки життєдіяльності та охорони праці	25.11.2020 р.	Виконано
6	Оформлення пояснювальної записки	26.11.2020 р.	Виконано
7	Подання роботи до експертної комісії	11.11.2020 р.	Виконано

Дата видачі завдання 02.11.2020

Студент

(підпис)

Паскарюк Д.О.

(прізвище, ініціали)

Керівник роботи

(підпис)

проф. Сінотін А.М.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 95 с., 0 табл., 31 рис., 3 дод., 20 джерел.

АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ, ГЛИБОКЕ НАВЧАННЯ, ІДЕНТИФІКАЦІЯ ОБ'ЄКТІВ, НЕЙРОМЕРЕЖЕВА СИСТЕМА, НЕЙРОННІ МЕРЕЖІ.

В сучасному світі використовуються системи, які автоматично вирішують велику кількість операцій за секунду, тому діяльність людини у цих системах є дуже суперечливою. Саме тому, зростає зацікавленість штучними нейронними мережами, нейрокомп'ютерами та нейромережевими системами для удосконалення вирішення складних задач та мінімізації похибок при вирішуванні важливих задач, які потребують негайного рішення.

Об'єктом дослідження є процес ідентифікації персоналу підприємству на основі використанні нейромережевої системи ідентифікації.

Метою роботи є розробка архітектури та практична реалізація нейромережевої системи для ідентифікації персоналу підприємств різного профілю та рівней.

Предметом дослідження є підвищення ефективності нейромережевої системи ідентифікації персоналу підприємства.

В магістерській атестаційній роботі проведено аналіз існуючих архітектур нейромережових систем та методів ідентифікації, досліджено процес розпізнавання образів системою ідентифікації об'єктів на зображеннях та у реальному часі. Розроблена нейромережева система ідентифікації персоналу підприємства, здатна розпізнавати обличчя, індивідуальні ознаки та емоції людини.

ABSTRACT

An explanatory note is made by: 95 pp., 0 tab., 31 fig., 3 add., 20 sources.

NEURAL NETWORK ARCHITECTURE, DEEP LEARNING, OBJECT IDENTIFICATION, NEURAL NETWORK SYSTEM, NEURAL NETWORKS.

In a modern world, there are systems that automatically run through a large number of operations per second, which makes people even more intelligent in these systems. The very fact, the growth of the performance of piece neuronal framing, neurocomputers and neuromechanical systems for more sophisticated transmission of folding tasks and minimization of failures when searching for important tasks, such as require a non-secret solution.

The subject of advising the effectiveness of the neuromechanical system and identification of the personnel of the enterprise.

About the process of identification to the personnel of the enterprise on the basis of the victorian neuro-neuronal system and identification.

By the method of robots, the development of architecture is a practical implementation of a neuromechanical system for identifying the personnel of enterprises with a growing profile and equal.

In the master's attestation robot, an analysis of the visual architectures of the neuro-cutter systems and methods of identification was carried out, as well as the process of developing images by the system of identification of images on the images of the images. The neuromerezheva system of identification has been broken up to the personnel of the enterprise, it has been given to identify the individual, the individual signs of the people.

ЗМІСТ

	с.
Скорочення та умовні позначки	8
Вступ.....	9
1 Аналіз існуючих нейронних мереж ідентифікації об'єктів	11
1.1 Розпізнавання об'єктів.....	11
1.2 Методи фільтрації візуальної інформації	13
1.3 Поняття штучної нейронної мережі.....	16
1.4 Методика навчання нейромережевих систем.....	23
1.5 Типи навчання нейромережевих систем.....	25
1.6 Багатошарова нейронна мережа	26
1.7 Висновки до першого розділу... ..	26
2 Розробка архітектура нейронної мережі.....	30
2.1 Види архітектур нейронної мережі	30
2.2 Функція активації (RELU).....	33
2.3 Техніка dropout	35
2.4 Методи еволюції нейронних систем.....	37
2.5 Нормалізація нейронної мережі.....	41
2.6 Лінійний класифікатор	42
2.7 Висновки до другого розділу.....	42
3 Проектування системи ідентифікації.....	42
3.1 Особливості проектування системи ідентифікації персоналу	43
3.2 Метод пошуку серед множини облич	45
3.3 Розташування облич.....	50
3.4 Локальні моделі сірого рівня (Local Grey-level Models).....	52
3.5 Опис методів програмного забезпечення	56
3.6 Алгоритм роботи системи розпізнавання	66
3.7 Порівняння систем ідентифікації у реальному часі та роздільної ідентифікації	68
3.8 Охорона праці під час виконання проєкту	71
3.9 Висновки до третього розділу.....	75
Висновки	76
Перелік джерел посилання.....	77

Додаток А Фрагменти програмного коду.....	79
Додаток Б Презентаційні матеріали.....	84
Додаток В Відомість атестаційної роботи.....	95

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AOM – аналогові мережі;

НМ – нейронна мережа;

ІНМ – штучна нейронна мережа;

ANN – artificial neural networks;

HOG – Histogram of Oriented Gradients;

CNN – convolutional neural network;

DNN – deconvolutional neural networks;

MLP – multilayered perceptron;

ReLU – rectified linear unit;

RGB – red, green, blue.

ВСТУП

За останні роки, одним з найбільш розповсюдженим та вагомим відкриттям у інноваційному світі є технологія розпізнавання. У середині минулого століття проблематикою у сфері розпізнавання займалася наука під назвою комп'ютерний зір. Першопрохідцем у роботі над пізнанням нейронних клітин можемо вважати Сантьяго Рамона-і-Кахаля, який сформлював структуру та описав основні типи нервових клітин людини.

Сприйняття світу проходить через когнітивні функції людини (сприйняття простору, мислення, прийняття рішень та інші). Саме розпізнавання образів є невід'ємною частиною життєдіяльності людини в цілому. Візуальне сприйняття є одним з важливіших «операцій» мозку. За допомогою зору ми обробляємо приблизно 60 відсотків інформації щодня. При візуальній ідентифікації ми проводимо велику кількість операцій відразу, а саме, пошук необхідних або відомих об'єктів, аналіз їх первинних характеристик, визначаємо пріоритети операцій з цим об'єктом.

Термін «штучна нейронна мережа» вперше використали Уоррен Маккалок та Уолтер Пітсс у 1943 році. За основу гіпотези створення штучної мережі були взяті робота мозку та правила створення зв'язків між фізичними нейронами. Згідно з гіпотезою нейрони спрощено розглядались пристроєм, який оперував двійковими числами. Також це створило ієрархічну структуру, яка показувала зорову систему людини, засновану на шарах складних і простих клітин, розташованих в зорових зонах мозку.

Концепція продемонстрована Уорреном Маккалоком та Уолтером Пітссом була використана вченими для створення першої теоретичної моделі розпізнавання об'єктів за допомогою комп'ютерного зору. Проте, створення складної багаторівненої моделі в тот час не було можливим з причини невеликої обчислювальної потужності, тому, протягом наступних десятиліть використовували прості одношарові нейронні мережі.

На сучасних підприємствах все частіше використовують ідентифікацію персоналу. Це допомагає вести облік надходження людей, та збільшує безпеку підприємства. У сучасних реаліях за умови всесвітніх карантинних умов, нейромережі ведуть температурний аналіз працівників. Саме тому тематика даної роботи є актуальною.

В даній роботі використано архітектуру згорткової нейронної мережі, що створює нейромережеву систему розпізнавання персоналу, використовуючи підхід до класифікації із застосуванням ієрархічного класифікатору.

Об'єктом дослідження є процес ідентифікації персоналу підприємству на основі використання нейромережевої системи ідентифікації.

Предметом дослідження є підвищення ефективності нейромережевої системи ідентифікації персоналу підприємства.

Метою роботи є розробка архітектури та практична реалізація нейромережевої системи для ідентифікації персоналу підприємств різного профілю та рівней.

Для досягнення мети, планується розв'язати наступні завдання:

- провести аналіз існуючих нейронних мереж ідентифікації [5];
- розглянути існуючі нейромережі ідентифікації;
- знайти оптимальне рішення для розробки моделі системи ідентифікації;
- спроектувати модель нейромережі;
- оформити атестаційну роботу у відповідності з вимогами, що вказані у ДСТУ 3008:2015 [1], навчальному посібнику з дипломного проектування [2], методичних вказівках до випускної кваліфікаційної роботи рівня «Магістр» [3] положенні про протидію академічному плагіату [4], навчальному посібнику наукових дослідень [5].

1 АНАЛІЗ ІСНУЮЧИХ НЕЙРОННИХ МЕРЕЖ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ

1.1 Розпізнавання об'єктів

Розпізнавання образів (об'єктів, сигналів, ситуацій, явищ або процесів) - завдання ідентифікації об'єкта або визначення будь-яких його властивостей по його зображенню (оптичне розпізнавання) або аудіозаписи (акустичне розпізнавання) та інші характеристики. Інженерне рішення такої задачі, є досить актуальною задачею в автоматизованому виробництві різних сфер та рівней [6]. Під час процесу розпізнавання об'єктів слід враховувати вплив перегріву, теплового режиму технічних засобів, які використовуються [20].

Образ – класифікаційне угруповання в системі класифікації, яка об'єднує (виділяє) певну групу об'єктів за певною ознакою. Образи мають характерною властивістю, що виявляється в тому, що ознайомлення з кінцевим числом явищ з одного і того ж безлічі дає можливість дізнаватися як завгодно велике число його представників. Образи мають характерні об'єктивними властивостями в тому сенсі, що різні люди, які навчаються на різному матеріалі спостережень, здебільшого однаково і незалежно один від одного класифікують одні і ті ж об'єкти. У класичній постановці задачі розпізнавання універсальне безліч розбивається на частини-образи. Кожне відображення будь-якого об'єкта на сприймаючі органи системи, що розпізнає, незалежно від його положення щодо цих органів, прийнято називати зображенням об'єкта, а безлічі таких зображень, об'єднані певними загальними властивостями, являють собою образи

Методика віднесення елемента до якого-небудь образу називається вирішальним правилом. Ще одне важливе поняття - метрика, спосіб визначення відстані між елементами універсальної множини. Чим менший цей період, тим більше схожими є об'єкти (символи, звуки та ін.) - те, що ми розпізнаємо. Зазвичай елементи задаються у виді набору чисел, а метрика - у виді функції. Від вибору уявлення образів і реалізації метрики залежить ефективність програми,

один алгоритм розпізнавання з різними метриками буде помилятися з різною частотою.

Навчанням зазвичай називають процес вироблення в деякій системі тієї чи іншої реакції на групи зовнішніх ідентичних сигналів шляхом багаторазового впливу на систему зовнішньої коректування. Таку зовнішню коригування в навчанні прийнято називати "заохоченнями" і "покараннями". Механізм генерації цієї коригування практично повністю визначає алгоритм навчання. Самонавчання відрізняється від навчання тим, що тут додаткова інформація про вірність реакції системі не повідомляється.

Адаптація - це процес зміни параметрів і структури системи, а можливо - і керуючих впливів, на основі поточної інформації з метою досягнення певного стану системи при початковій невизначеності і мінливих умовах роботи [6].

Навчання - це процес, в результаті якого система поступово набуває здатність відповідати потрібними реакціями на певні сукупності зовнішніх впливів, а адаптація - це підстроювання параметрів і структури системи з метою досягнення необхідної якості управління в умовах безперервних змін зовнішніх умов [6].

Приклади завдань розпізнавання образів:

- розпізнавання букв;
- розпізнавання штрих-кодів;
- розпізнавання автомобільних номерів;
- розпізнавання осіб та інших біометричних даних;
- розпізнавання мови.

Іншим завданням розпізнавання є виділення із вихідних зображень характерних ознак та/або властивостей. Це завдання можна віднести до попередньої обробки. Ознака повинна являти собою характерну властивість конкретного класу, при цьому загальну для цього класу. Міжкласові ознаки – це ознаки, що визначають відмінності між класами. Загальні ознаки, що властиві усім класам, не несуть корисної інформації, тому для задачі розпізнавання

об'єктів не розглядаються як характерні. Вибрати правильні ознаки - одна із важливих задач побудови систем розпізнавання.

1.2 Методи фільтрації візуальної інформації

Проблема автоматичного визначення заданих об'єктів на знімках, відома як розпізнавання образів, привертає останнім часом увагу дослідників. Одна з основних труднощів, пов'язаних з вирішенням цього завдання, - низька якість знімків, обумовлене різними спотвореннями. Наприклад, на аерофотографій, одержуваних за допомогою літальних апаратів, часто мають місце випадкові перешкоди типу хмарності, туману, імли [5], тому для задач розпізнавання об'єктів доцільно використовувати попередню обробку зображень. У більшості випадків для боротьби з такими випадковими перешкодами застосовують стандартні фільтри низьких частот або медіанний фільтри [5].

Однак ці фільтри не тільки частково усувають шуми, але і розмивають кордони об'єктів, що ускладнює їх розпізнавання. Розглянемо метод предобробки зображення за допомогою послідовного застосування білатерально фільтра і шок-фільтра. Проведемо дослідження впливу попередньої обробки з використанням даного методу на ефективність і швидкість розпізнавання.

Метод контурного аналізу. Для розпізнавання зображень, що не володіють внутрішніми деталями, застосовується контурний аналіз [6]. Цей метод дозволяє розпізнавати об'єкти зі зміненим розміром, а також з невеликим (до 15°) кутом повороту. Можлива модифікація для об'єктів, повернутих під будь-яким кутом. Під контуром розуміється межа об'єкта, тобто сукупність точок (пікселів), що відокремлюють об'єкт від фону. Контур кодується послідовністю, що складається з комплексних чисел. На контурі фіксується початкова точка. Потім контур обходиться в заданому напрямку (наприклад, за годинниковою стрілкою), і кожен вектор зміщення V записують комплексним числом $V = x + iy$,

де x – зміщення точки по осі X ; y – зміщення по осі Y . Зсув береться щодо попередньої точки контуру. Зсув першої точки визначається щодо точки $(0, 0)$. Модуль надає міру подібності контурів, а його аргумент - кут повороту контурів відносно один одного (при збереженні початкової точки контуру). Таким чином, можна знайти однакові контури на заданому зображенні.

Білатеральна фільтрація. Білатеральний фільтр дозволяє згладжувати зображення, усуваючи шуми. На відміну, наприклад, від медіанного фільтра білатеральний фільтр, згладжуючи зображення, залишає об'єкти досить чіткими. Однак при цьому контур зображення трохи розпливається. Основна ідея білатерально фільтра - врахувати близькість між пікселями різних кольорів і зробити перехід кольорів більш послідовним, а зображення більш плавним.

Шок-фільтр. Для виділення чіткої межі між зонами з різною інтенсивністю кольору використовується шок-фільтр [8]. Визначити близькість аналізованої точки до кордону можна, обчисливши градієнт функції інтенсивності кольору в цій точці, так як на кордоні він буде мати найбільше значення. Однак градієнта недостатньо, щоб дізнатися, до якої зони інтенсивності належить піксель, оскільки великі значення градієнта також характеризують окремі пікселі, що відрізняються від загального фону. Для того щоб визначити потрібну зону, використовується оператор Лапласа. Якщо позначити через $f(x, y)$ вихідне зображення, то вираз для шок-фільтра може бути представлено у виді

Алгоритм попередньої обробки знімків. Застосування окремо шок-фільтра або білатерально фільтра для предобработки недоцільно, так як шок-фільтр посилює шуми, а білатеральний фільтр робить контур складно виділеним за рахунок розмитості зображення. Тому в роботі запропоновано алгоритм, заснований на послідовному застосуванні спочатку білатерально, а потім шок-фільтра. Початкове зображення літака (рис. 1.1) має розмиті краю і зашумлено. Застосування білатерально фільтра призводить до згладжування шумів, але в той же час розмиває кордони фігури. Шок-фільтр робить над ділянкою чіткими, але не посилює шуми, так як вони були мінімізовані білатеральним фільтром.



Рисунок 1.1 – Демонстрація роботи фільтру

Результати, одержані при використанні алгоритму, і, відповідно, якість розпізнавання зображень залежать від констант d і r . Далі наведені результати дослідження впливу цих коефіцієнтів на якість розпізнавання. На рис. 1.2, показано вплив коефіцієнтів d і r білатерально фільтра на якість розпізнавання (крива 1). Значення НМП отримані в результаті контурного аналізу, описаного в розд. 1. Чим менше значення НМП, тим менше подібність контурів, а значить, можливість розпізнавання об'єктів. Для порівняння на графіках показаний результат, отриманий без застосування фільтрів взагалі (крива 2).

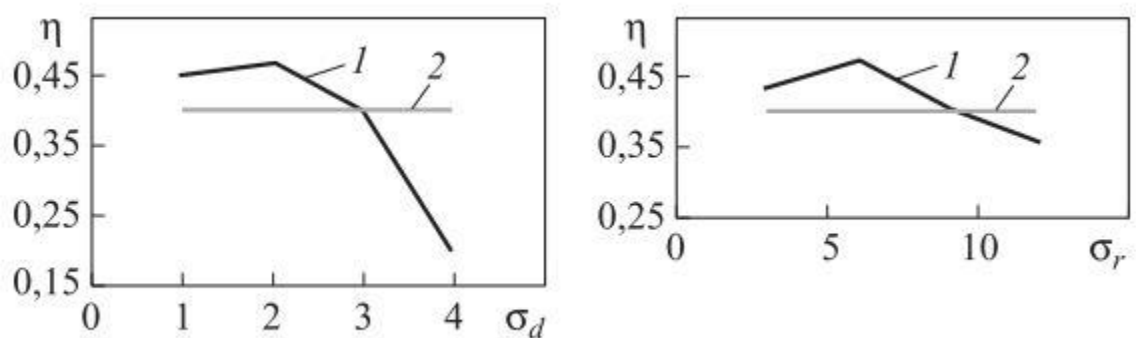


Рисунок 1.2 – Графік кривої застосування фільтрів

При збільшенні коефіцієнтів білатерально фільтра колір об'єктів наближається до кольору фону і кордони стають менш розбірливими, за рахунок чого при зчитуванні контуру виникають помилки. У вихідних прикладах це

особливо позначається на крилах літаків, так як вони оточені з двох сторін пікселями іншого кольору, кожен з яких впливає на колір результату [9]. Оскільки шуми, що додаються шок-фільтром, не псують над ділянкою, різкого зниження якості розпізнавання при збільшенні коефіцієнта не спостерігається. Однак в окремих місцях може бути порушена межа, за рахунок чого якість може бути знижено. З використанням підібраних коефіцієнтів була розглянута задача розпізнавання літаків на тестовому аерофотознімки. Результати були отримані при відсутності попередньої обробки і з застосуванням розглянутого алгоритму фільтрації.

1.3 Поняття штучної нейронної мережі

Штучна нейронна мережа - математична модель, яка має у собі декілька шарів, що виконують обчислення паралельно. Така архітектура була вперше створена за принципу дрібних обчислювальних елементів людського мозку - нейронами [8].

Нейронами у штучній нейронній мережі називаються мінімальні обчислювальні елементи. Нейронні мережі зазвичай складаються з трьох та більше шарів: вхідного шару, прихованого шару (або шарів) і вихідного шару (рис. 1.3), проте іноді вхідний і вихідний шари не враховуються, і тоді кількість шарів в мережі вважається за кількістю прихованих шарів. Такий тип нейронної мережі називається перцептрон.

В основі перцептрона лежить математична модель сприйняття інформації мозком. Різні дослідники по-різному його визначають. У найзагальнішому своєму виді (як його описував Розенблатт) він представляє систему з елементів трьох різних типів: сенсорів, асоціативних елементів і реагують елементів. Розглянемо принцип роботи перцептрона.

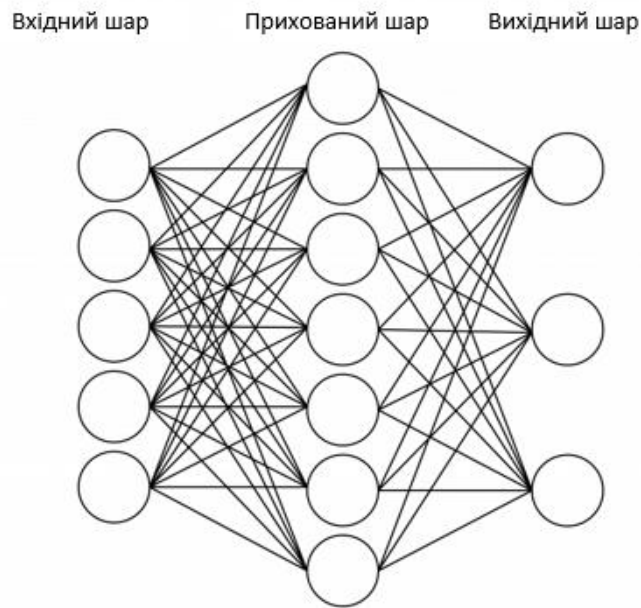


Рисунок 1.3 – Найпростіший перцептрон [6]

Першими в роботу включаються S -елементи. Вони можуть перебувати або в стані спокою (сигнал дорівнює 0), або в стані збудження (сигнал дорівнює 1). Далі сигнали від S -елементів передаються A -елементів по так званим S - A зв'язків. Ці зв'язки можуть мати ваги, рівні тільки -1, 0 або 1.

Потім сигнали від сенсорних елементів, які пройшли по S - A зв'язків потрапляють в A -елементи, які ще називають асоціативними елементами. Варто зауважити, що одному A -елементу може відповідати кілька S -елементів. Якщо сигнали, що надійшли на A -елемент, в сукупності перевищують деякий його поріг θ , то цей A -елемент збуджується і видає сигнал, рівний 1. В іншому випадку (сигнал від S -елементів не перевищив порогу A -елемента), генерується нульовий сигнал.

Одношаровий перцептрон. Ключова особливість полягає в тому, що кожен S -елемент однозначно відповідає одному A -елементу, всі S - A зв'язки мають вагу, рівний +1, а поріг A елементів дорівнює 1.

Багатошаровий перцептрон. Коли шари перцептрона об'єднані разом, вони утворюють багатошарову архітектуру, і це дає необхідну складність обробки

нейронної мережі. Багатошарові перцептрони (MLP) є найбільш широко використовуваної архітектурою для нейронних мереж (рис.1.4).

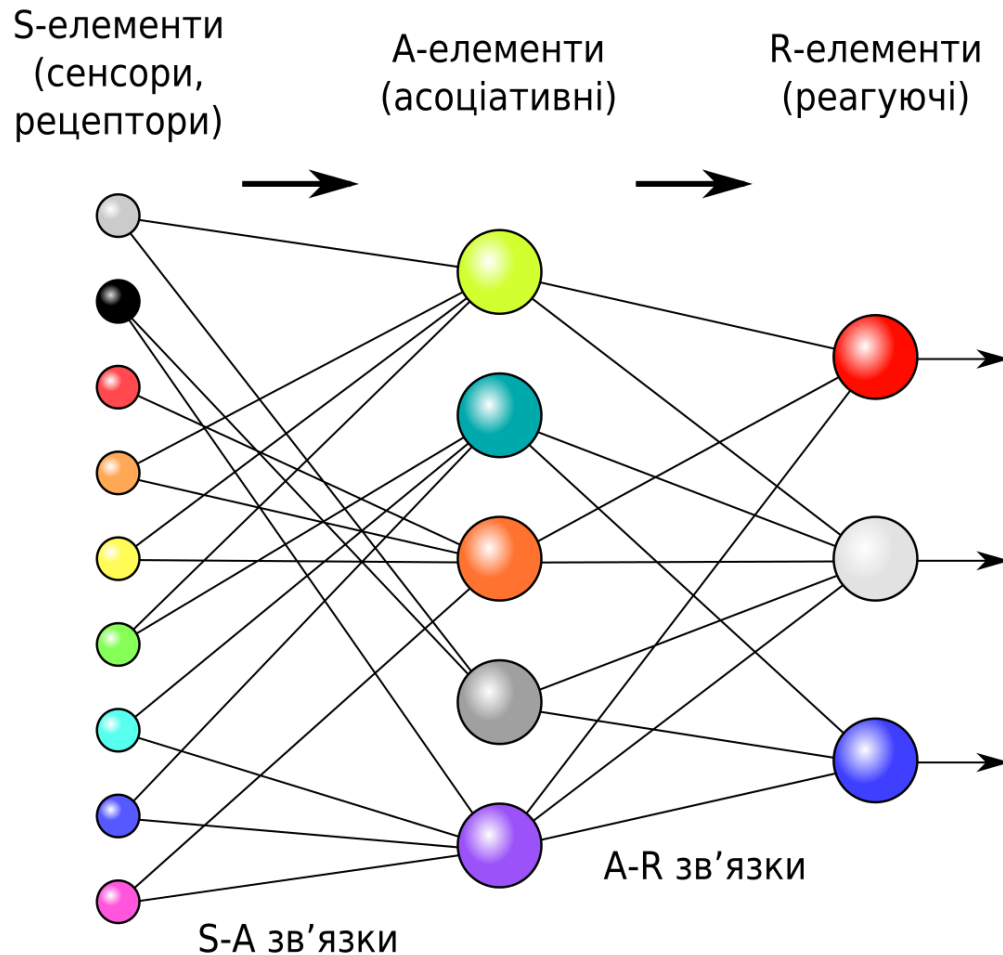


Рисунок 1.4 – Багатошаровий перцептрон

Під багатошаровим перцептроном розуміють два різних види: багатошаровий перцептрон за Розенблаттом і багатошаровий перцептрон за Румельхартом.

Багатошаровий перцептрон за Розенблаттом - перцептрон, у якого є більше 1 шару *A* елементів.

Багатошаровий перцептрон за Румельхартом - багатошаровий перцептрон за Розенблаттом, у якого навчання підлягають ще й S-A зв'язку, а також саме навчання проводиться за методом зворотного поширення помилки.

Обробка від вхідного рівня до прихованого шару(ів), а потім до вихідного рівня називається прямим поширенням. Сума (вхід * вага) + зміщення застосовується на кожному рівні, а потім значення функції активації поширюється на наступний рівень. Наступний шар може бути іншим прихованим шаром або вихідним шаром. У побудові нейронних мереж використовується велика кількість прихованих шарів для створення Deep Neural Network (DNN). Як тільки вихід досягнутий, на останньому шарі (вихідному шарі) ми обчислюємо помилку (передбачений вихід за вирахуванням вихідного виводу). Ця помилка необхідна для корекції ваг і зміщень, які використовуються при прямому поширенні. Тут використовується похідна функція. Кількість ваги, яке необхідно змінити, визначається градієнтним спуском. В процесі зворотного поширення використовується приватна похідна кожної функції активації нейрона для визначення нахилу (або градієнта) в напрямку кожного з вхідних ваг. Градієнт показує, як круто помилка буде зменшена або збільшена для зміни ваги. Зворотне поширення продовжує змінювати ваги, поки не буде сильно зменшено кількість помилок на, відоме як швидкість навчання. Швидкість навчання - це скалярний параметр, аналогічний розміром кроку в чисельному інтегруванні, який використовується для установки швидкості коригувань для більш швидкого зменшення помилок. Швидкість навчання використовується при зворотному поширенні при коригуванні ваг і зміщень.

Покрокова ілюстрація нейронної мережі і функції активації.

Вхідний шар має один нейрон, і вихід буде вирішувати двійкову завдання класифікації (передбачити 0 або 1) (рис. 1.5) [16].

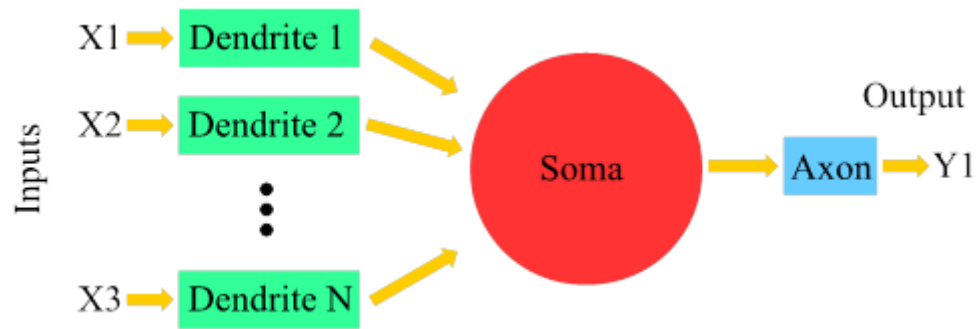


Рисунок 1.5 – Модель нейрона

Функція активації (активаційна функція, функція збудження) – функція, обчислює вихідний сигнал штучного нейрона. Як аргумент приймає сигнал, одержуваний на виході вхідного суматора. Найбільш часто користуються наступні функції активації:

- лінійний поріг або гістерезис;
- нескладна кусочно-лінійна функція;
- має два лінійних ділянки, де функція;
- активації тотожно дорівнює мінімально;
- допустимому і максимально допустимому;
- значенням і є ділянка, на якому функція;
- строго монотонно зростає.

Проста кусочно-лінійна функція. Якщо вхідне значення менше порогового, то значення функції активації одно мінімального допустимого, інакше – максимально допустимому.

Особливістю нейронної мережі є її вміння навчатися на прикладах, такий підхід називається навчанням з учителем. Нейронна мережа навчається по принципу великої кількості прикладів, вони складаються з пар Input/Output (відповідні один одному Input і Output). В завданні розпізнавання об'єктів такою парою буде вхідне зображення і відповіднай йому назва об'єкта. Навчання нейронної мережі - ітеративний процес, що зменшує відхилення виходу мережі від заданого «відповіді вчителя» - лейбла, яке відповідає даному зображенню

(рис. 1.6). Цей процес складається з кроків, які називаються епохами навчання (вони зазвичай обчислюються тисячами), на кожному з яких відбувається підгонка «ваг» нейронної мережі - параметрів прихованих шарів мережі. По завершенні процесу навчання якість роботи нейронної мережі зазвичай досить гарне для виконання завдання, під яку вона була навчена, хоча оптимальний набір параметрів, ідеально розпізнає всі зображення, часто підібрати неможливо.



Рисунок 1.6 – Процес навчання нейронної мережі

Для розпізнавання складних об'єктів створюють системи на основі нейронних мереж (НМ, neural network). Вони можуть мати топологію, орієнтовану на розв'язання конкретної задачі із врахуванням властивостей об'єкта – просторово-часову орієнтацію, масштаб, геометричні параметри об'єкта, включаючи координати, кутове положення, лінійний розмір, відстань, тощо. У той же час істотним недоліком типових НМ є відсутність ефективних засобів для розв'язання задач розпізнавання динамічних образів. Основною проблемою інтерпретації динамічних візуальних сцен є висока розмірність простору ознак, наявність геометричних перетворень над об'єктом. Стиск простору ознак виконують методом витягу інтегральних і інваріантних до

геометричних перетворень параметрів зображень. Метод геометричних та більш загальних алгебраїчних інваріантів відіграє значну роль у розв'язанні задач розпізнавання зображень. Так, наприклад, інваріанти, у тому числі інваріантні моменти, були успішно використані для розпізнавання профілів літаків і танків, друкованих і рукописних букв, параметрів стикувального вузла космічного апарата, а також багатьох інших об'єктів. Математичне обґрунтування інваріантних особливостей напівтонових зображень базується на теорії алгебраїчних інваріантів.

Сутність НМ полягає в тому, що мережа складається з елементів, котрі називаються формальними нейронами (formal neuron). Кожен нейрон приймає набір сигналів, що надходять на його входи від одної групи таких же нейронів, обробляє сигнали з врахуванням попередніх сигналів і адаптації до них на основі процедур навчання і передає результати обробки другій групі нейронів. Зв'язки між нейронами кодуються вагами, що відображають важливість їх інформації для визначення загального результату. Основний принцип настроювання нейронної мережі полягає в застосуванні процедур оптимізації та адаптації на основі певних критеріїв, здатності до перенавчання. Однією з переваг НМ є те, що всі елементи можуть функціонувати паралельно, тим самим істотно підвищуючи ефективність розв'язання задач, особливо при обробці зображень в реальному часі. Системи розпізнавання об'єктів зображення, що засновані на нейронних мережах, використовують ієрархічну архітектуру. Спочатку вектор ознак обробляється грубою з високим рівнем похибок, але швидкою, мережею, далі, якщо вектор не був класифікований як не об'єкт, алгоритм розв'язання коректується більш точною і більш повільною мережею [12].

Переважає кількість прикладних нейронних систем передбачає використання багат шарових перцептронів (назва „перцептрон” походить з англійського perceptron – сприйняття, оскільки перші зразки таких структур призначались для моделювання зору). Популярність перцептронів зумовлена широким колом доступних для них задач. Загалом вони вирішують задачу

апроксимації багатовимірних функцій, іншими словами – побудову багатовимірного відображення.

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

Розглянемо алгоритм навчання перцептрона на простій модельній задачі. Перцептрон навчають, подаючи сукупність (множину) зображень по одному на його вхід і змінюють ваги доти, доки для всіх зображень не буде досягнуто необхідний вихід. Припустимо, що вхідні зображення нанесено на демонстраційні карти. Кожну карту розбито на квадрати і від кожного квадрата на перцептрон подається вхідний сигнал. Якщо в квадраті є лінія, то від неї подається одиниця, у протилежному випадку – нуль. Сукупність квадратів на карті задає сукупність нулів і одиниць, котрі подаються на входи перцептрона. Мета полягає в тому, щоб навчити перцептрон вмикати індикатор за умови подавання на нього сукупності входів, що задають непарне число, і не вмикати у випадку парного. На рисунку 1.7 показана така перцептронна конфігурація.

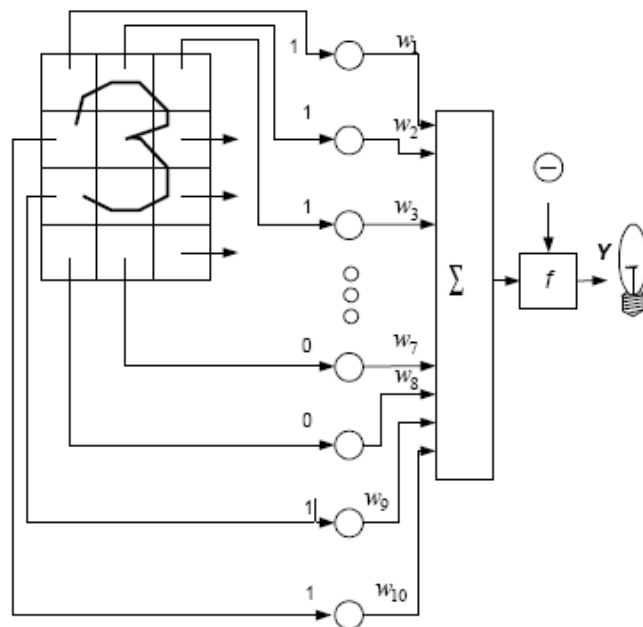


Рисунок 1.7 – Перцептронна система розпізнавання зображень

1.4 Методика навчання нейромережових систем

Перші нейромережі створювалися як структура кори головного мозку.

Базовий рівень мережі, перцептрон, є по суті математичним аналогом біологічного нейрона. Як і в головному мозку, в нейромережі можуть з'являтися пересічені один з одним перцептрони [19].

Перший шар нейромережі називається вхідним. Кожен вузол цього шару отримує на вхід будь-яку інформацію і передає її на наступні вузли в інших шарах. Найчастіше між вузлами одного шару немає зв'язків, а останній вузол ланцюжка виводить результат роботи нейромережі. Вузли посередині називаються прихованими, оскільки не мають з'єднань із зовнішнім світом, як вузли виведення і введення. Вони викликаються тільки в разі активації попередніх шарів (рис. 1.8) [17].

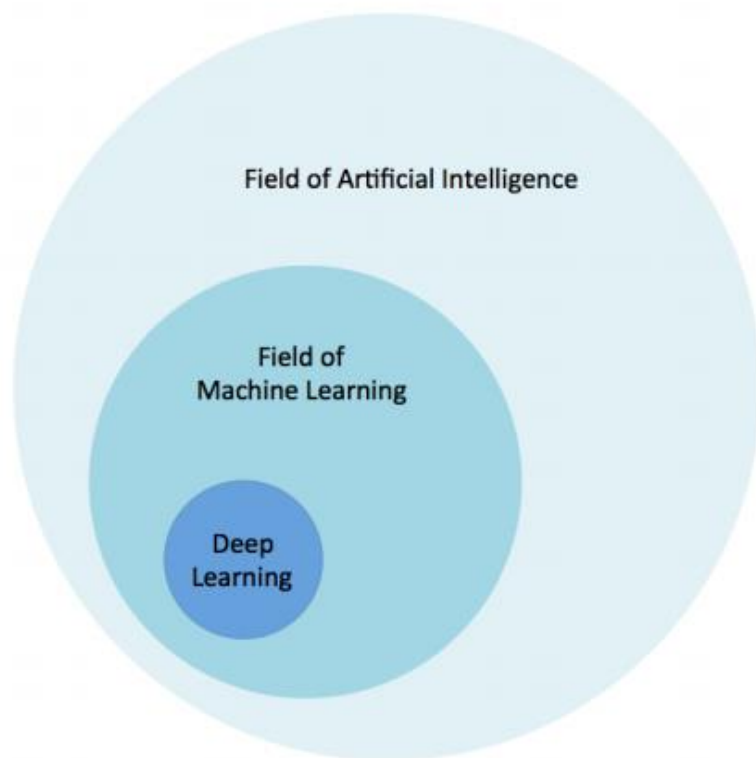


Рисунок 1.8 – Структура глибинного навчання нейронної мережі

Глибинне навчання складається з:

- штучного інтелекту в цілому (наприклад, комп'ютери);
- машинного навчання;
- глибинного навчання і штучні нейромережі.

Глибинне навчання – це по суті техніка навчання нейромережі, яка використовує безліч шарів для вирішення складних проблем (наприклад, розпізнавання мови) за допомогою шаблонів. У вісімдесятих роках більшість нейромереж були одношаровими в силу високої вартості і обмеженості можливостей даних.

1.5 Типи навчання нейромережевих систем

Навчання нейромережевих систем може бути з вчителем, без вчителя та з підкріпленням.

Навчання з вчителем. При навчанні з учителем дані організовані як сукупність розмічених зразків. Окремий елемент $\{(x_j, y_i)\}_{i=1}^N$ має назву вектор ознак. Вектор ознак - це вектор, в якому кожен вимір $j = 1, \dots, D$ містить значення, яке відображає деяку характеристику зразка. Це значення називається ознакою і позначається як $x^{(j)}$. Для всіх даних в наборі даних ознака в позиції j в векторі ознак завжди описує одну і ту ж характеристику. Це означає, що якщо описується характеристика деякого зразка x_i^n в умовних позначеннях, тоді також буде описувати характеристики умовних позначеннях кожного іншого зразка $x_k, k = 1, \dots, N$. Показник y_i може бути елементом кінцевої безлічі класів $\{1, 2, \dots, C\}$, дійсним числом або більш складною структурою, такий як вектор, матриця, дерево або граф.

Навчання без вчителя. У навчанні без вчителя набір даних представлений сукупність нерозмічених зразків. Метою алгоритму навчання без вчителя – створення моделі, яка приймає вектор ознак X на вході і перетворює його в інший вектор або в значення, яке можна використовувати для вирішення практичного

завдання. В завданнях кластеризації модель повертає ідентифікатор кластера для кожного вектора ознак в наборі даних. У завданнях зменшення розмірності модель повертає вектор ознак, який має менше елементів, ніж вхідний вектор X . У завданнях виявлення аномалій повертається дійсне число, яке вказує, наскільки x відрізняється від «типового» зразка в наборі даних.

Навчання з підкріпленням – це розділ машинного навчання, де передбачається, що машина «живе» в певному оточенні і здатна сприймати стан цього оточення як вектор характеристик. Машина може виконувати деякі дії в кожному стані. Різні дії приносять різні винагороди, а також можуть перевести машину в інший стан оточення. Мета алгоритму навчання з підкріпленням - вивчити лінію поведінки. Лінія поведінки – це стратегія (схожа на модель в навчанні з вчителем), яка приймає вектор ознак, що описує стан, і повертає оптимальне дію для виконання в цьому стані. Дія є оптимальним, якщо призводить до максимального очікуваного середнього винагороді. Навчання з підкріпленням вирішує особливий клас завдань, коли рішення приймаються послідовно, а мета є довгостроковою, наприклад, гра в відеогру, роботизація виробництва, управління ресурсами або логістика [13].

1.6 Багатошарова нейронна мережа

Багатошарова нейронна мережа (перцептрон) – це нейронна мережа, що складається з вхідного, вихідного і розташованих між ними одного (або кількох) прихованих шарів нейронів [12]. Щоб побудувати багатошаровий перцептрон, необхідно вибрати його параметри за наступним алгоритмом:

- визначити, який зміст вкладається в компоненти вхідного вектора X ;
- вхідний вектор повинен містити формалізоване умову задачі, тобто всю інформацію, необхідну для того, щоб отримати відповідь;
- визначити вихідний вектор Y таким чином, щоб його компоненти містили повну відповідь для поставленого завдання;

- визначити вид функції активації нейронів. При цьому бажано врахувати специфіку завдання, так як вдалий вибір збільшить швидкість навчання;
- визначити кількість шарів і нейронів у шарі;
- задати діапазон зміни входів, виходів, ваг і порогових рівнів на основі обраної функції активації;
- присвоїти початкові значення ваг і граничним рівням. Початкові значення не повинні бути великими, щоб нейрони не опинилися в насиченні (на горизонтальній ділянці функції активації), інакше навчання буде дуже повільним. Початкові значення не повинні бути і занадто малими, щоб виходи здебільшого нейронів були рівні нулю, інакше навчання теж сповільниться;
- провести навчання, тобто підібрати параметри мережі так, щоб задача вирішувалася найкращим чином. Після закінчення навчання мережа зможе вирішувати завдання того типу, яким вона навчена;
- подати на вхід мережі умови задачі у виді вектора X . Розрахувати вихідний вектор Y , який і дасть формалізоване рішення задачі.

Алгоритм зворотного поширення помилки є одним з методів навчання багат шарових нейронних мереж прямого поширення. Навчання алгоритмом зворотного поширення помилки передбачає два проходи по всім верствам мережі: прямого і зворотного. При прямому проході вхідний вектор подається на вхідний прошарок нейронної мережі, після чого поширюється по мережі від шару до шару. В результаті генерується набір вихідних сигналів, який і є фактичною реакцією мережі на даний вхідний образ. Під час прямого проходу все синаптичні ваги мережі фіксовані. Під час зворотного проходу все синаптичні ваги налаштовуються відповідно до правил корекції помилок, а саме: фактичний вихід мережі віднімається з бажаного, в результаті чого формується сигнал помилки. цей сигнал згодом поширюється по мережі в напрямку, протилежному напрямку синаптичних зв'язків. Звідси і назва – алгоритм зворотного поширення помилки. Синаптичні ваги налаштовуються з метою максимального наближення вихідного сигналу мережі до бажаного [19].

Введемо наступні позначення: X_i – вхідний вектор, Y_i – вихідний вектор, w_{ij}^k , j – i -ий ваговий коефіцієнт j -го нейрона k -го шару, b_i^k – поріг i -го нейрона k -го шару, d_i – еталонне вихідне значення i -го нейрона.

Вихідне значення j -го нейрона k -го шару обчислюється за формулою (1.1)

$$Y_j^k = F(\sum w_{ij}^k Y_i^{k-1} - b_j^k) . \quad (1.1)$$

Вихідне значення j -го нейрона вихідного шару обчислюється за формулою (1.2)

$$Y_j^k = F(\sum w_{ij} Y_i^{n-1} - b_j) . \quad (1.2)$$

Розглянемо систему з M лінійних нерівностей в N -вимірному просторі

$$A * x \leq b , \quad (1.3)$$

Алгоритм рішення має наступний вид:

$$x(0) = [0] , \quad (1.4)$$

$$y(k) = \psi(Ax(k) - b) , \quad (1.5)$$

$$x(k+1) = x(k) + HA^T y(k) , \quad (1.6)$$

де A – це матриця коефіцієнтів (входів), оскільки зображення є чорно-біле отже, вона є двійкової. Коли елемент матриці $a_{ij} = 0$ немає сенсу розглядати нерівність, тому будемо розглядати випадок, коли $a_{ij} = 1$, тоді отримуємо нерівність такого виду $x \leq b$. Використовуючи формулу для знаходження $x(k+1)$ отримуємо

$$x(k) + h * A^T * y(k) \leq b . \quad (1.7)$$

Функція активації

$$\psi(g) = \begin{cases} 1, & g \leq 0; \\ 0, & g > 0; \end{cases} \quad i = 1 \dots M , \quad (1.8)$$

де $y(k)$ – похибка, що може приймати два значення 0 або 1, слід розглядати випадок коли $y(k) = 1$. Елемент транспонованої матриці (АТ) може дорівнювати 0 або 1, у випадку коли елемент матриці дорівнює 1, тоді створюється нерівність такого виду:

$$x(k) + w_{i,j} \leq b, \quad (1.9)$$

$$x(k) + h \leq b. \quad (1.10)$$

Алгоритм навчання багатошарової НМ, полягає в наступному:

- задаються крок навчання α ($0 < \alpha < 1$) і бажана середньоквадратична;
- помилка мережі E_m ;
- ініціалізувалися випадковим чином вагові коефіцієнти порогові b_k^j значення НМ;
- подаються послідовно образи з навчальної вибірки на вхід нейронної мережі.

При цьому для кожного образу виконуються наступні дії:

- проводиться фаза прямого поширення вхідного образу по нейронній мережі. Обчислюється вихідне значення всіх нейронів Y_{jk} ;
- обчислюються помилки Y_j нейронів вихідного і прихованого шарів.
- проводиться зміна вагових коефіцієнтів і порогів нейронних елементів для кожного шару нейронної мережі;
- обчислюється сумарна помилка нейронної мережі E ;
- якщо $E > E_m$, то відбувається перехід до кроку 3, інакше виконання алгоритму завершується.

1.7 Висновки до першого розділу

Технологія створення машин, що мають здатність бачити, лежить в основі комп'ютерного зору. Ця дисципліна відноситься до штучних систем, які можуть виконувати завдання розпізнавання. У вищенаведеному розділі наведена

основна теорія розпізнавання об'єктів, що лежить в основі класифікації систем ідентифікації.

Огляд методів вирішення задач комп'ютерного зору за групами обґрунтував переваги нейромережевого методу над усіма запронованими. Принцип роботи нейронів штучної мережі дозволяє виявити його активність, що визначається його параметрами - вагами і його функцією активації.

Розглянуті основні принципи навчання нейронних мереж з їх особливостями показали, що ефективного навчання розпізнавання об'єктів можна досягти саме методом зворотнього поширення помилки.

2 РОЗРОБКА АРХІТЕКТУРИ НЕЙРОННОЇ МЕРЕЖІ

2.1 Види архітектур нейронної мережі

Нейромережі DNN це ANN з декількома прихованими шарами між вхідним і вихідним шарами. Подібно дрібним ANN, DNN можуть моделювати складні нелінійні відносини [19].

Глибокі нейронні мережі (DNN) з безліччю рівнів можуть бути ефективними в складних завданнях, таких як розпізнавання зображень або мови. Послідовні рівні використовуються для моделювання підвищення глибини семантики.

Зв'язок між вхідними та вихідними даними вивчена при навчанні нейронної мережі вхідних даних. Напрямок графа триває з вхідних даних через прихований шар і на вихідний шар. Всі вузли шару з'єднані по зваженим краях до вузлів наступного шару. Щоб обчислити вихідні дані мережі для певних вхідних даних, значення обчислюється на кожному вузлі в прихованих шарах і на вихідному шарі. Значення задається шляхом обчислення зваженої суми значень вузлів з попереднього шару. Потім функція активації застосовується до цієї зваженої суми.

Нейромережі CNN. Згорткові нейронні (рис 2.1) мережі працюють на основі фільтрів, які займаються розпізнаванням певних характеристик зображення (наприклад, прямих ліній) [19].

Фільтр - це сукупність кернелів (ядер) іноді в фільтрі використовується один кернел. Кернел - це звичайна матриця чисел, які називаються вагами, які "навчаються" (підлаштовуються, якщо вам так зручніше) з метою пошуку на зображеннях певних характеристик.

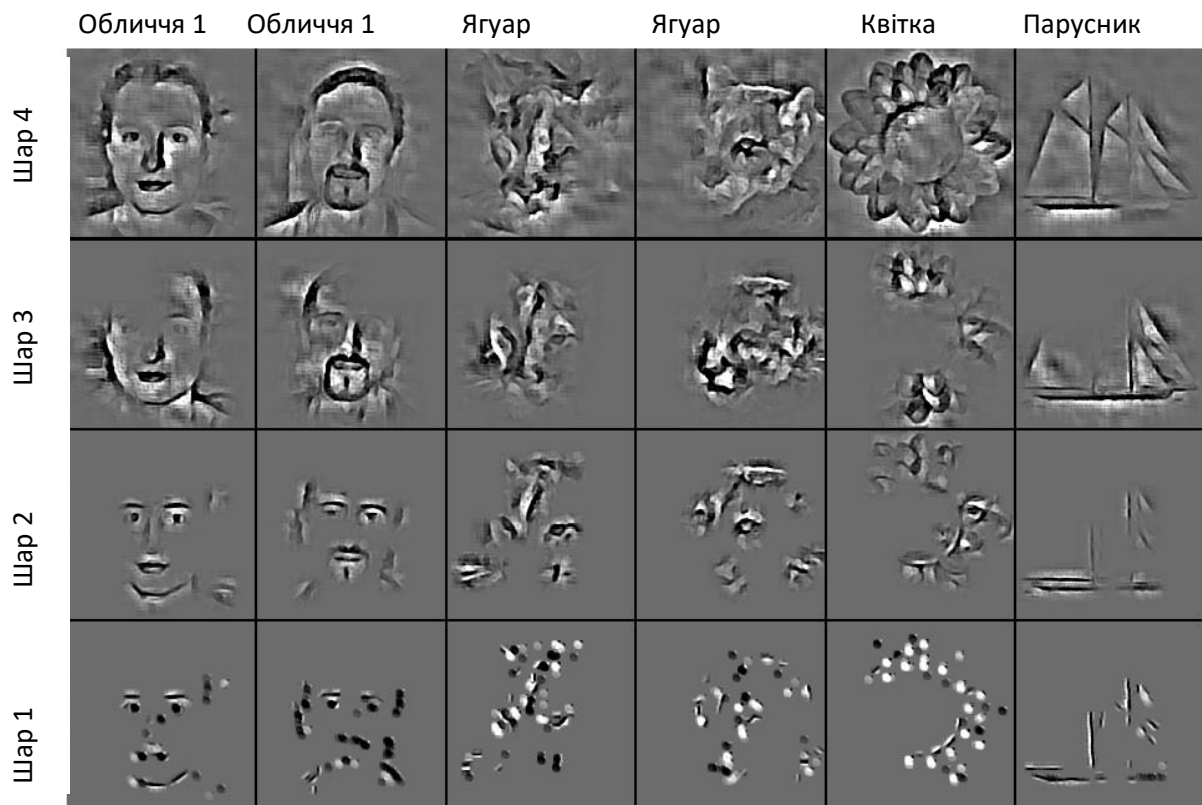


Рисунок 2.1 – Результат обробки мережі DNN

Фільтр переміщається уздовж зображення і визначає, чи присутня деяка шукана характеристика в конкретній його частині (рис. 2.2). Для отримання відповіді такого роду здійснюється операція згортки, яка є сумою добутків елементів фільтра і матриці вхідних сигналів.

Якщо деяка характеристика, пошук якої ведеться, присутня у фрагменті зображення, операція згортки на виході буде видавати число з відносно великим значенням. Якщо ж характеристика відсутня, вихідне число буде маленьким.

Кількість каналів фільтра повинна відповідати кількості каналів вихідного зображення, тільки тоді операція згортки буде виробляти належний ефект. Якщо вихідна картинка складається з трьох каналів (RGB: Red, Green, Blue), фільтр також повинен мати три канали.

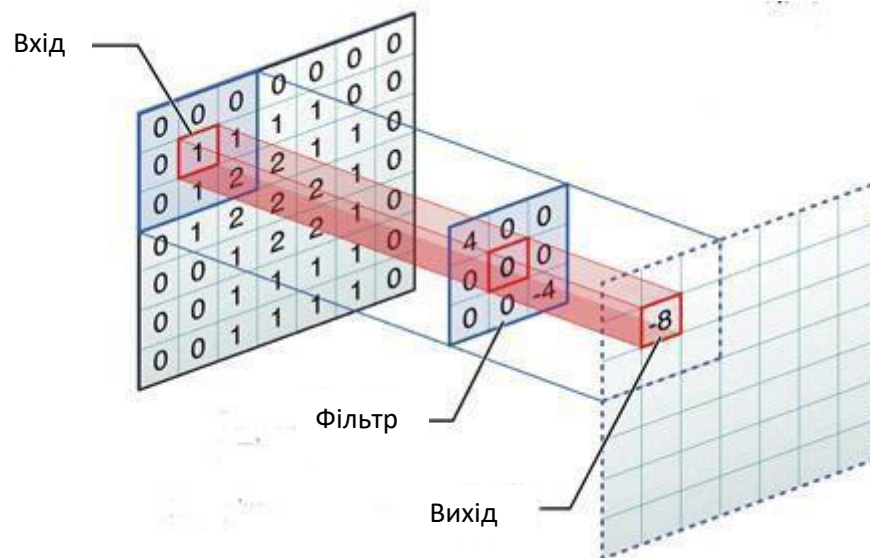


Рисунок 2.2 – Операції згортки

Фільтр може переміщатися вздовж матриці вхідних сигналів з кроком, відмінним від одиниці. Крок переміщення фільтра називається страйд (stride). Страйд визначає, на яку кількість пікселів повинен зміститися фільтр за одну ітерацію.

Для того, аби навчання ваг, укладених в Кернел, було ефективним, в результати згорток слід ввести деякий зсув (bias) і нелінійність.

Зсув - це статична величина, на яку слід "змістити" вихідні значення. За своєю суттю це звичайна операція додавання кожного елемента вихідний матриці з величиною зміщення. Якщо пояснювати дуже поверхнево, це потрібно для того, щоб вивести нейронну мережу з тупикових ситуацій, що мають суто математичні причини [18].

Завдяки нелінійності картина, яка формується за допомогою операції згортки, отримує певне перетворення, що дозволяє нейронній мережі більш якісно оцінювати ситуацію. Цю необхідність дуже грубо можна порівняти з необхідністю людям зі слабким зором носити контактні лінзи. А взагалі-то така необхідність пов'язана з тим, що вхідні дані за своєю природою нелінійні, тому потрібно навмисне спотворювати проміжні результати, щоб відповідь нейронної мережі був відповідним.

В багатьох випадках в якості функції активації використовують ReLU (Rectified Linear Unit).

Рівні введення, менші або рівні нулю, перетворюються в нуль; значення, що перевищують нуль, не змінюються.

2.2 Функція активації (ReLU)

Одним з етапів розробки нейронної мережі є вибір функції активації нейронів. Вид функції активації багато в чому визначає функціональні можливості нейронної мережі і метод навчання цієї мережі. Класичний алгоритм зворотного поширення помилки добре працює на двошарових і тришарових нейронних мережах, але при подальшому збільшенні глибини починає відчувати проблеми. Одна з причин - так зване загасання градієнтів. У міру поширення помилки від вихідного шару до вхідного на кожному шарі відбувається домноження поточного результату на похідну функції активації. Похідна у традиційній сігмоїдній функції активації менше одиниці на всій області визначення, тому після декількох шарів помилка стане близькою до нуля. Якщо ж, навпаки, функція активації має необмежену похідну (як, наприклад, гіперболічний тангенс), то може статися вибуховий збільшення помилки у міру поширення, що призведе до нестійкості процедури навчання [13].

Відомо, що нейронні мережі здатні наблизити як завгодно складну функцію, якщо в них досить шарів і функція активації є нелінійною. Функції активації начебто сігмоїдній або тангенціальній є нелінійними, але призводять до проблем з загасанням або збільшенням градієнтів. Однак можна використовувати і набагато простіший варіант - випрямлену лінійну функцію активації (rectified linear unit, ReLU), яка виражається формулою (2.1)

$$f(s) = \max(0, s) \quad (2.1)$$

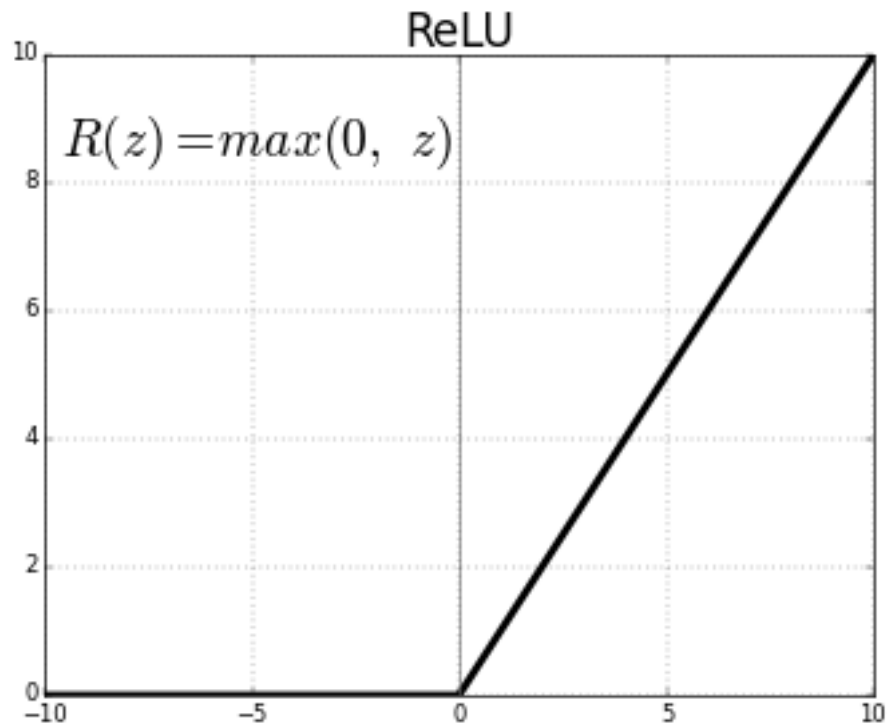


Рисунок 2.1 – Функція ReLU

Переваги використання ReLU. Її похідна дорівнює або одиниці, або нулю, і тому не може статися розростання або загасання градієнтів, тому що помноживши одиницю на дельту помилки отримаємо дельту помилки, якщо ж використовувати іншу функцію, наприклад, гіперболічний тангенс, то дельта помилки могла, або зменшитися, або зрости, або залишитися такою ж, тобто, похідна гіперболічного тангенса повертає число з різним знаком і величиною, що можна сильно вплинути на загасання або розростання градієнта. Більш того, використання даної функції приводить до проріджування ваг. Обчислення сигмоїд і гіперболічного тангенса вимагає виконання ресурсномістких операцій, таких як спорудження до рівня, в той час як ReLU може бути реалізований за допомогою простого порогового перетворення матриці активацій в нулі, відсікає непотрібні деталі в каналі при негативному виході.

З недоліків можна відзначити, що ReLU не завжди достатньо надійна і в процесі навчання може виходити з ладу («вмирати»). Наприклад, великий градієнт, що проходить через ReLU, може привести до такого оновлення ваг, що даний нейрон ніколи більше не активується. Якщо це станеться, то, починаючи з

даного моменту, градієнт, що проходить через цей нейрон, завжди буде дорівнює нулю. Відповідно, даний нейрон буде необоротно виведений з ладу. Наприклад, при дуже великій швидкості навчання (learning rate), може виявитися, що до 40% ReLU «мертві» (тобто, ніколи не активуються). Ця проблема вирішується за допомогою вибору належної швидкості навчання.

Leaky ReLU з «витоком» (leaky ReLU, LReLU) являє собою одну зі спроб вирішити описану вище проблему виходу з ладу звичайних ReLU. Звичайний ReLU на інтервалі $x < 0$ дає на виході нуль, в той час як LReLU має на цьому інтервалі невелике від'ємне значення (кутовий коефіцієнт близько 0,01). Тобто функція для LReLU має вид $f(x) = \alpha x$ при $x < 0$ і $f(x) = x$ при $x \geq 0$, де α - мала константа. Деякі дослідники повідомляють про успішне застосування даної функції активації, але результати не завжди стабільні.

Parametric ReLU Для параметричного ReLU (parametric ReLU, PReLU). Кутовий коефіцієнт на негативному інтервалі не ставить попередньо, а визначається на основі даних. Процес зворотного поширення помилки і поновлення для PReLU досить простий і подібний до відповідного процесу для традиційних ReLU.

Randomized ReLU Для рандомізованого ReLU (randomized ReLU, RReLU) кутовий коефіцієнт на негативному інтервалі під час навчання генерується випадковим чином із заданого інтервалу, а під час тестування залишається постійним.

2.3 Техніка dropout

Одна з головних проблем використання глибоких нейронних мереж, чи то багатoshарових звичайних, чи то згорткових – це перенавчання (англ. overfitting). Перенавчання проявляється у тому, що створена модель починає пояснювати тільки приклади з навчальної вибірки, адаптуючись до навчальних прикладів, замість того щоб вчитися класифікувати приклади, які не брали участі в навчанні

(втрачаючи здатність до узагальнення). За останні роки було запропоновано безліч рішень проблеми перенавчання, але одне з них перевершило інші, завдяки своїй простоті і досягненням у практичних результатах. Техніку Dropout можна описати як метод, який використовується для запобігання перенавчання та певної адаптації нейронів до вхідних даних шляхом встановлення вихідного сигналу будь-якого нейрона у значення нуля з ймовірністю p . Виключені нейрони не вносять свій внесок в процес навчання ні на одному з етапів алгоритму зворотнього поширення помилки (англ. *backpropagation*), який часто застосовується до навчань мереж; тому зб виключення хоча б одного з нейронів рівносильно навчанню ніби нової нейронної мережі. Графічне представлення техніки Dropout (рис. 2.4) в якій техніка була вперше представлена. Головна ідея техніки Dropout - замість навчання однієї нейромережі навчити ансамбль декількох нейромереж, а потім усереднити отримані результати [6].

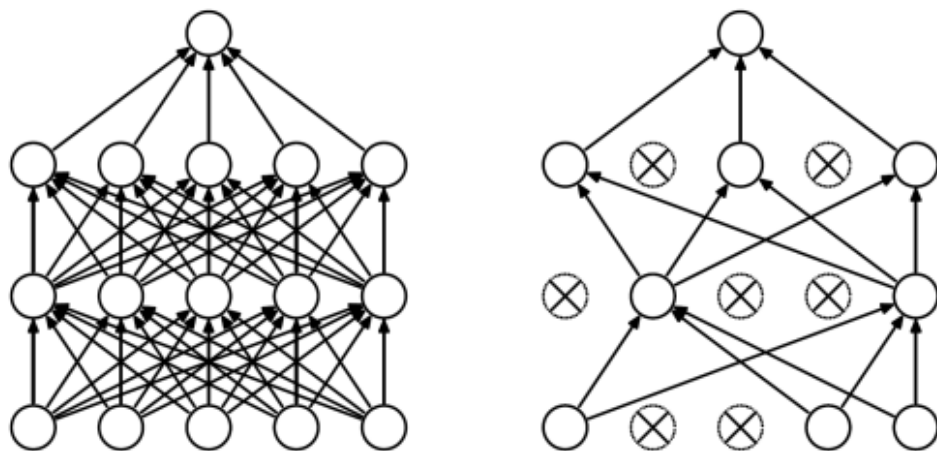


Рисунок 2.4 — Представлення техніки Dropout

Ліворуч на рисунку вид звичайної нейронної мережі, а праворуч – та сама мережа після застосування техніки Dropout. Мережа, зображена зліва, використовується при тестуванні, після навчання параметрами. У стандартній нейронній мережі похідна, отримана кожним параметром, повідомляє цьому ж параметру, як він повинен змінитися, щоб, з огляду на діяльність інших блоків,

мінімізувати функцію втрат. Тому блоки можуть змінюватися, виправляючи при цьому помилки інших блоків. Це може привести до надмірної спільної адаптації (англ. co-adaptation), що, призводить до перенавчання, оскільки ці спільні адаптації неможливо узагальнити для даних, які не брали участі в навчанні. Техніка Dropout запобігає спільній адаптації для кожного прихованого блоку, утворюючи невеликий відсоток надійності існування інших прихованих блоків. Тому прихований блок не може покладатися на інші блоки в виправленні власних помилок. Імовірність виключення кожного нейрона однакова. Застосування техніки Dropout на етапі навчання можна представити як змінену функцію активації

$$out = D \cdot a(h), \quad (2.1)$$

де $a(h)$ - функція активації, а символ $\cdot$ добуток Адамара [7].

2.4 Методи еволюції нейронних систем

ENS3 (Evolution of Neural Systems by Stochastic Synthesis, еволюція нейронних систем шляхом стохастичного синтезу) - метод, основу якого складає біологічна теорія коеволюції. Метод був розроблений в 2007 р. в інституті теоретичної математики імені Макса Планка, Френком Паземаном і Ульфом Дікманом. Коеволюція - це процес спільної еволюції, при якому пристосованість окремих особин залежить від інших особин популяції. Метод ENS3 застосуємо для ШНМ, що складаються з нейронів, використовують сигмоїдну функцію активації, але не накладає обмежень на кількість нейронів і топологію. Він одноразово оптимізує топологію нейромережі і її параметри: пороги активації, вагові коефіцієнти [17].

Метод заснований на поведінково-орієнтованому підході до проектування нейронних систем. Спочатку він був розроблений для вивчення складної динаміки в сенсорно-рухових системах автономних роботів і програмних агентів.

Метод ENS3 складається з 7 основних кроків:

- генерація початкової популяції $P(0)$, що складається з $n(0)$ нейромереж;
- створення наступної популяції $P'(0)$ шляхом репродукції початкової популяції $P(t)$;
- зміна структури і параметрів особин популяції $P'(t)$ з послідовним застосуванням стохастичних операторів S , E , V . Оператор V (Variation operator);
- стохастичний аналог мутації, який реалізує додавання і видалення нейронів, зв'язків і коригування параметрів. Результат його застосування залежить від ймовірностей, постійних для окремих нейронів і зв'язків. Оператор розрахунку E (Evaluation operator) визначається розв'язуваною завданням і в загальному випадку задається в термінах функції пристосованості. Оператор селекції S (Selection operator) визначає кількість особин, що потрапляють в нову популяцію;
- оцінка кожної особини вихідної і похідної популяцій $P(t)$ і $P'(t)$, розрахунок функцій пристосованості до їх особливостей;
- генерація популяції $P'(t+1)$ з використанням рангової селекції особин популяцій $P(t)$ і $P'(t)$ відповідно до їх значеннями пристосованості;
- перевірка досягнення критеріїв зупинки (по досягненню максимальної кількості епох або заданої точності нейросетевого виведення).

Якщо жоден з критеріїв не досягнутий, здійснюється перехід до кроку 2.

До переваг методу відносяться простота реалізації і висока продуктивність для нейромереж маленького і середнього розміру. Однак в методі ENS3 не реалізований оператор кросинговеру, що необхідно поррахувати його недоліком, так як при належній реалізації кросинговер здатний істотно скоротити час еволюції і прискорити формування оптимальної особини. Метод ENS3 також не допускає модифікацію виду функцій активації нейронів. Крім того, фіксовані ймовірності мутації для нейронів і зв'язків скорочують простір пошуку: багато ефективні конфігурації ШНМ не можуть бути сформовані в зв'язку з низькою ймовірністю зміни в відповідних вузлах ШНМ поточної популяції.

Стандартний метод нейроеволюції CNE (Conventional NeuroEvolution, стандартна нейроеволюція) – метод, що модифікує вагові коефіцієнти нейромережі і не зачіпає її структуру. У «стандартної нейроеволюція» є безліч реалізацій, але як правило під даним терміном розуміється в першу чергу сам метод CNE. Структура ШНМ, кількість нейронів і всі параметри за винятком вагових коефіцієнтів задаються в методі заздалегідь. Для оптимізації як правило вибираються повнозв'язні ШНМ.

Метод CNE складається з чотирьох основних кроків:

- генерування випадковим чином початкової популяції $P(0)$ з хромосом, що формуються шляхом послідовної конкатенації значень ваг. Вагові коефіцієнти можуть бути представлені двома способами: дійсними числами або у виді бінарного уявлення;
- оцінка пристосованості ШНМ. перетворення закодованого генотипу в фенотип з подальшою оцінкою ШНМ. Зупинка по досягненню умови оптимальності ШНМ;
- рекомбінирования: селекція (відбір кращих особин в ранжируваному ряду), мутація, кроссинговер;
- формування нової популяції $P(t + 1)$ і перехід на нову епоху еволюції $t + 1$.

До переваг методу відносяться: простота реалізації перетворень генотип-фенотип, висока швидкість роботи за рахунок компактного представлення генотипів. У методі CNE також є і недоліки. Так, існує висока ймовірність збіжності методу до локального оптимуму. Метод вимагає попереднього завдання топології нейромережі, кількості нейронів і зв'язків, а також має низьку масштабованість і не призначений для вирішення складних завдань: кількість параметрів, якими оперує метод, полиномиально зростає в залежності від розміру нейромережі.

2.5 Нормалізація нейронної мережі

Нормалізація вхідного шару нейронної мережі зазвичай виконується шляхом масштабування даних, що подаються в функції активації. Наприклад, коли є ознаки зі значеннями від 0 до 1 і деякі ознаки зі значеннями від 1 до 1000, то їх необхідно нормалізувати, щоб прискорити навчання. Нормалізацію даних можна виконати і в прихованих шарах нейронних мереж, що і робить метод пакетної нормалізації

Пакетна нормалізація зменшує величину, на яку зміщуються значення вузлів в прихованих шарах (т.зв. коваріантне зрушення (англ. Covariance shift)).

Коваріантне зрушення - це ситуація, коли розподілу значень ознак в навчальній і тестовій вибірці мають різні параметри (математичне очікування, дисперсія і т.д.). Коваріантність в даному випадку відноситься до значень ознак.

Використання пакетної нормалізації володіє ще кількома додатковими корисними властивостями:

- досягається більш швидка збіжність моделей, незважаючи на виконання додаткових обчислень;
- пакетна нормалізація дозволяє кожному шару мережі навчатися більш незалежно від інших верств;
- стає можливим використання більш високого темпу навчання, так як пакетна нормалізація гарантує, що виходи вузлів нейронної мережі не матимуть занадто великих або малих значень;
- пакетна нормалізація в якомусь сенсі також є механізмом регуляризації: даний метод привносить в виходи вузлів прихованих шарів деякий шум, аналогічно методу dropout;
- моделі стають менш чутливі до початкової ініціалізації ваг.

Навчання нейронних мереж з пакетною нормалізацією. Для навчання нейронних мереж необхідно обчислювати градієнт функції втрат.

У прямому напрямку, як і описано в алгоритмі методу, з входу x обчислюється середнє значення по кожній розмірності простору ознак. Потім

отриманий вектор середніх значення віднімається з кожного елемента навчальної вибірки. Далі обчислюється дисперсія, і за допомогою неї обчислюється знаменник для нормалізації. Потім отримане значення інвертується і множиться на різницю входу x і середніх значень. В кінці застосовуються параметри γ і β .

У зворотному напрямку обчислюються похідні необхідних функцій. У наступній таблиці докладніше зображені кроки обчислення градієнта функції втрат (де $N = m$ і $D = d$).

У прямому напрямку, з входу x обчислюється середнє значення по кожній озмірності простору ознак. Потім отриманий вектор середніх значення віднімається з кожного елемента навчальної вибірки. Далі обчислюється дисперсія, і за допомогою неї обчислюється знаменник для нормалізації. Потім отримане значення інвертується і множиться на різницю входу x і середніх значень. В кінці застосовуються параметри γ і β . У зворотному напрямку обчислюються похідні необхідних функцій.

2.6 Лінійний класифікатор

Як і в звичайних нейронних мережах, в згорткових мережах в повнозв'язну шари (класифікатор) нейрони мають зв'язок з усіма функціями активації з попереднього шару. Активації шарів класифікації можуть бути обчислені за допомогою множення матриць, що супроводжується зміщенням.

Відмінність між шаром класифікації і шаром згортки полягає у тому, що нейрони шару згортки з'єднані тільки з локальною областю на вході, і що нейрони цього шару можуть спільно використовувати параметри. Однак нейрони в обох шарах, незважаючи на свої особливості, підраховують скалярний добуток, тому їх функціональна форма ідентична. Більш того, можна виконати конвертацію між повнозв'язним і згортковим шарами. Класифікатор повинен запам'ятовувати всі навчальні дані та зберігати їх для подальших порівнянь із даними з тестового запуску. Це дуже ресурсозатратно, оскільки набори даних

можуть бути розміром у гігабайтах. Лінійний класифікатор дає оцінку класу як зважену суму всіх значень пікселів у трьох його кольорових каналах.

Залежно від того, які значення встановлено для цих ваг, функція класифікатору має здатність позитивно або негативно оцінювати (залежно від знаку кожної ваги) певні кольори у певних положеннях зображення. Наприклад, людина стверджуватиме, що клас "корабель" може бути більш імовірним, якщо з боків на зображенні є багато синього кольору (що може відповідати воді).

Ще один спосіб організації класифікатору – шаблонізація. Особливість способу стосується ваг W і полягає в тому, що кожен рядок W відповідає шаблону (або іноді також називається прототипом) для одного з класів. Оцінка кожного класу для зображення отримується шляхом порівняння кожного шаблону, один за одним, із зображенням за допомогою внутрішньої ознаки (або точки-ознаки), щоб знайти те, що "підходить" найкраще. За допомогою такого порівняння лінійний класифікатор знаходить відповідність шаблону.

2.7 Висновки до другого розділу

У розділі 2 було розглянуто загальну топологію згорткових нейронних мереж, також описано усі складові їх архітектури та значення параметрів і гіперпараметрів згорткових мереж. Також було описано суть техніки dropout, що має широке застосування при запобіганні перенавчань мереж.

Розглянуто методи еволюції нейронних систем а саме, ENS3 (Evolution of Neural Systems by Stochastic Synthesis, еволюція нейронних систем шляхом стохастичного синтезу) та метод нейроеволюції CNE (Conventional NeuroEvolution, стандартна нейроеволюція). Для системи ідентифікації слід використовувати метод еволюції нейронних систем шляхом стохастичного синтезу, саме тому, що переваги цього методу такі як, швидкість еволюції та простота реалізації, необхідні у динамічних системах розпізнавання.

У розділі пояснено суть нормалізації вхідних даних для шарів та важливість запобігати зниженню швидкості навчання мережі та кількості її параметрів.

3 ПРОЕКТУВАННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ

3.1 Особливості проектування системи ідентифікації персоналу

Розпізнавання осіб складається з безлічі взаємозалежних підзадач, в яких потрібно:

- проаналізувати зображення і знайти на ньому все обличчя;
- навчитися розпізнавати кожну особу, навіть якщо воно дивним чином повернуто, або якщо освітлення погане;
- вміти визначати унікальні риси обличчя, які відрізняють одну людину від інших, наприклад, розмір очей, форма обличчя і так далі;
- порівняти виявлені унікальні особливості цієї особи з усіма людьми, яких система вже знає, щоб зрозуміти, хто зображений на фото.

Мозок людини робить все миттєво і непомітно для нас. Можна сказати, що люди настільки звикли до процедури розпізнавання осіб, що бачать обличчя навіть в повсякденних об'єктах (психологічний феномен парейдолії) (рис. 3.1).



Рисунок 3.1 – Зображення феномену парейдолії

Комп'ютери не в змозі до таких узагальнень, тому нам доведеться вчити їх цьому крок за кроком.

Для цього потрібно побудувати свого роду конвеєр, на якому виконується кожен крок розпізнавання по-окремо, а потім передаємо результат поточного кроку наступного. Іншими словами, послідовно з'єднується кілька алгоритмів машинного навчання:

- знаходження обличчя на зображенні;
- аналіз рис обличчя;
- порівняння з вже відомими особами;
- зробити висновки.

3.2 Метод пошуку серед множини облич

Розпознавання осіб – це відмінна функція для камери. Якщо камера може автоматично виокремити обличчя, вона може переконатись, що всі особи будуть у фокусі, перш ніж робити знімок (рис. 3.2).

Немає необхідності використовувати цей алгоритм для іншої цілі – потрібно видати області на зображення, які передаються наступному кроку конвеєра.



Рисунок 3.2 – Приклад ідентифікації «серед множини обличч»

Щоб знайти обличчя на зображенні, спершу необхідно зробити зображення чорно-білим (рис. 3.3), тому що для пошуку осіб кольорові дані не потрібні.



Рисунок 3.3 – Перетворення на монохромне зображення

Потім розглядається кожен піксель на нашому зображенні, один за іншим (рис. 3.4). Але цікавить не стільки сам піксель, скільки пікселі, що оточують його:

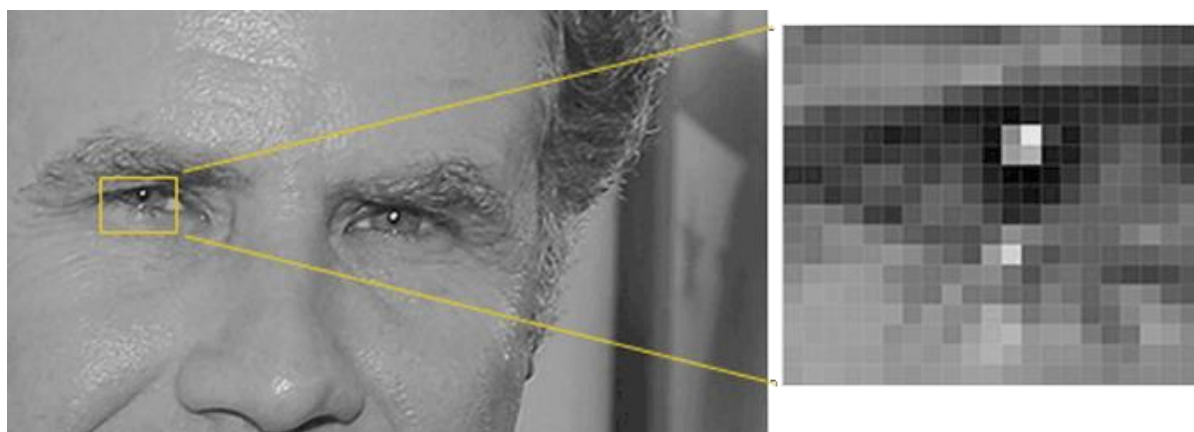


Рисунок 3.4 – Розглядання пікселів один за одним

Метою перетворення є – з'ясувати, наскільки темним є поточний піксель в порівнянні з сусідніми (рис.3.5). Потім зображується стрілка, яка показує, в якому напрямку зображення стає темніше:

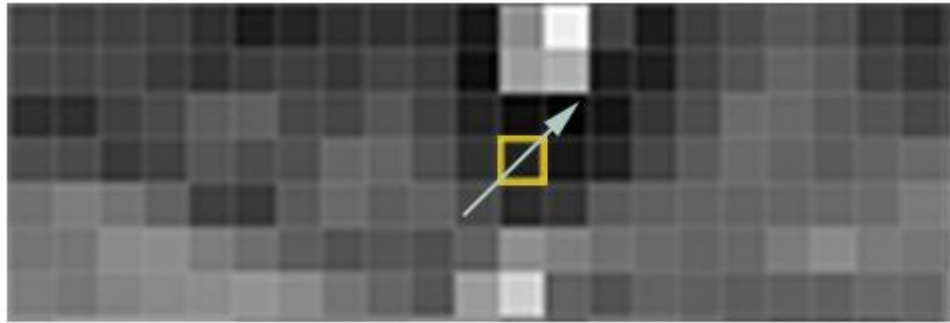


Рисунок 3.5 – Пошук найтимнішої ділянки зображення

Виконавши цю процедуру для кожного окремого пікселя зображення, можна замінити кожен піксель стрілкою (рис.3.6). Ці стрілки називаються градієнтами, і вони показують напрямок від світлих пікселів до темних по всьому зображенню:

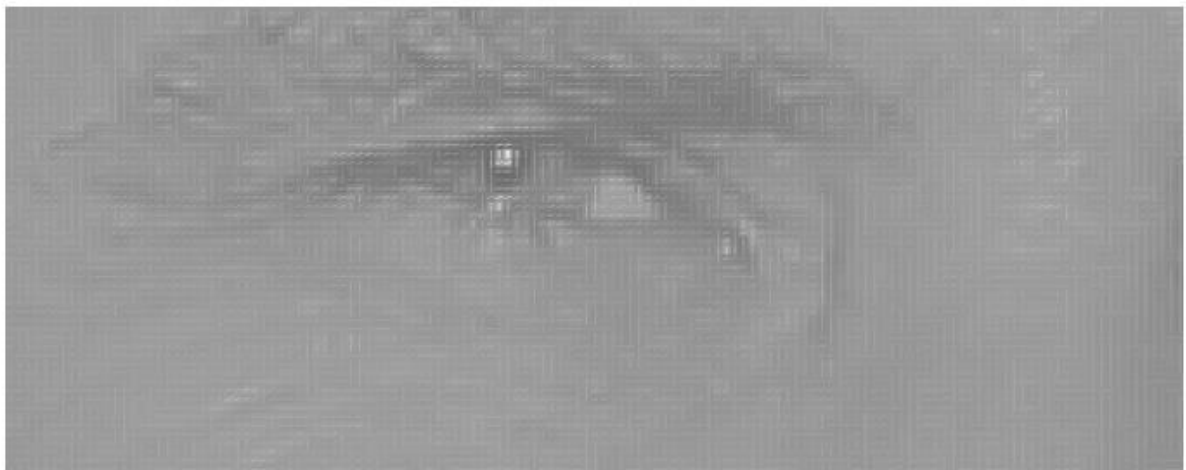


Рисунок 3.6 – Заміна пікселів на стрілки

Цей метод не обирається випадково, насправді є вагома причина замінювати пікселі градієнтами (рис.3.7). Якщо проаналізувати пікселі безпосередньо, то побачимо, що темні зображення і світлі зображення одного і того ж людини матимуть абсолютно різні значення яскравості пікселів.

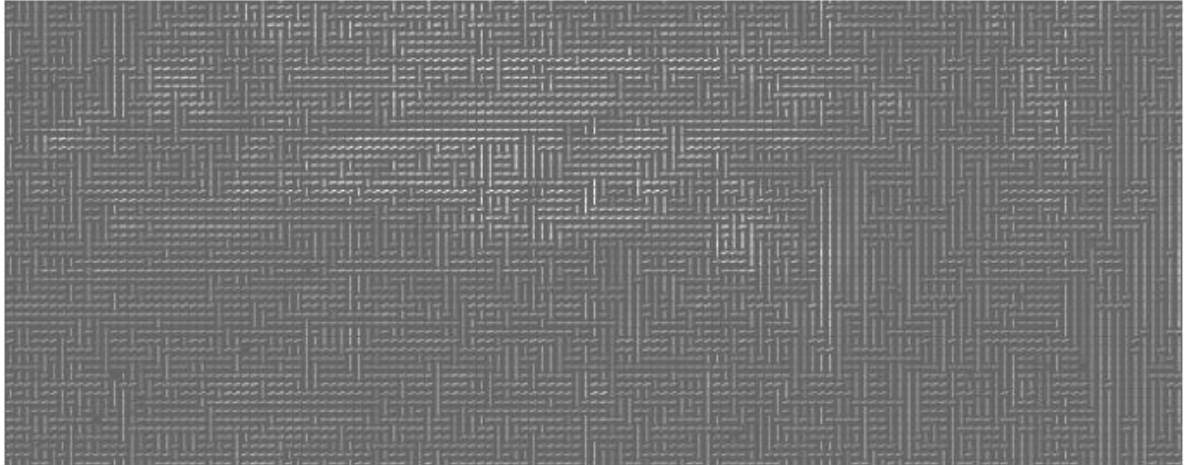


Рисунок 3.7 – Зображення градієнтів

Але, розглядаючи тільки напрямок зміни яскравості, створюється однакова картина незалежно від яскравості вихідного зображення. Це значно спрощує завдання. Однак, було б краще, якби можна було побачити основний напрямок зміни світла на більш високому рівні, отримуючи таким чином загальну канву зображення [19].

Для цього розділяють зображення на маленькі квадрати розміром 16 x 16 пікселів кожен (рис. 3.8). У кожному квадраті підраховується, скільки точок градієнта повернуто в кожному з основних напрямків (вгору, вгору, вправо, вправо і т.д.). Потім замінюється цей квадрат на зображенні стрілками, спрямованими туди ж, куди і більшість.

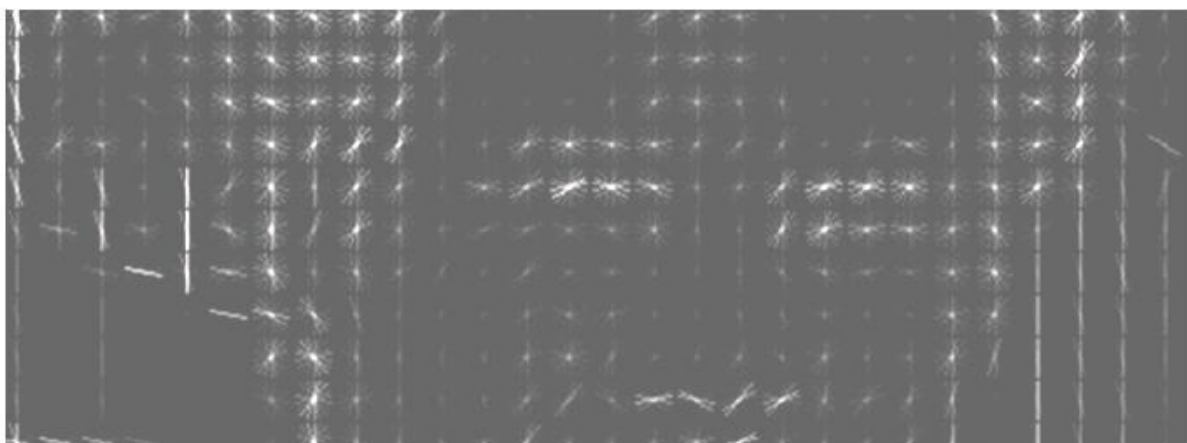


Рисунок 3.8 – Зображення методу розбивання зображення

Щоб знайти особу на HOG-зображенні, все, що нам потрібно зробити, це знайти частину нашого зображення, яка найбільш схожа на відомий малюнок HOG, отриманий з безлічі інших осіб в ході навчання (рис. 3.9).

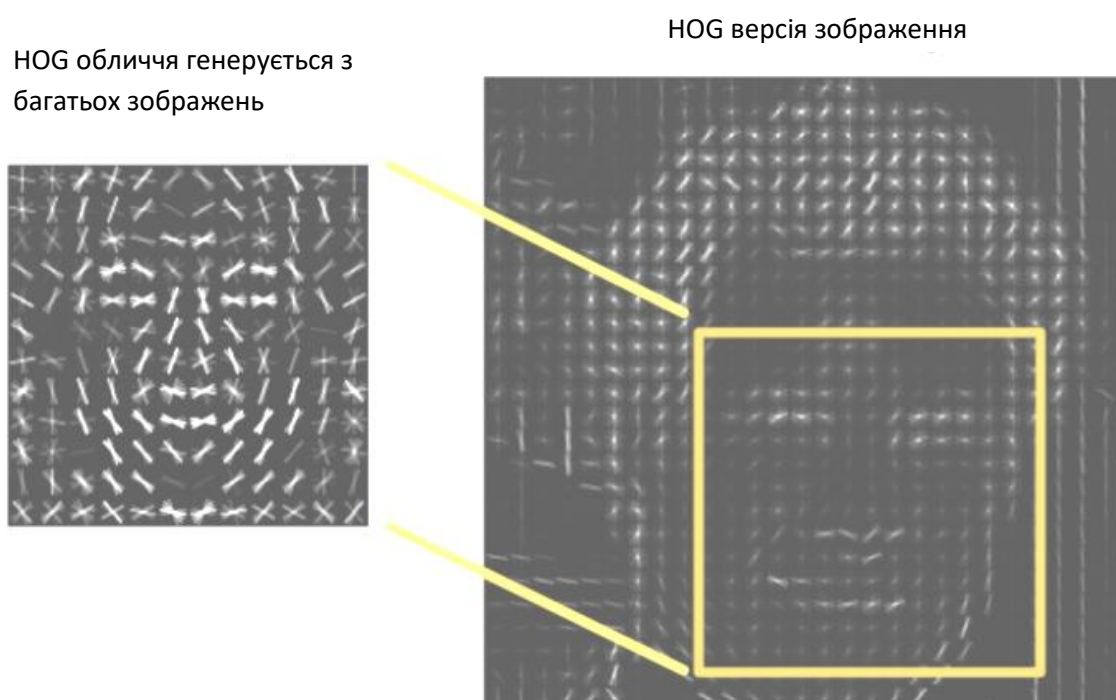


Рисунок 3.9 – Представлення HOG-зображення

Використовуючи цей метод, з легкістю можливо знайти особу на будь - якому зображенні.

3.3 Розташування облич

Для ідентифікації осіб можна використовувати модель ASM (Active Shape Models). Основна ідея полягає в обліку статистичних зв'язків між розташуванням антропометричних точок обличчя. На кожному зображенні особи точки пронумеровані в однаковому порядку (рис. 3.10). За їх взаємного розташування здійснюється порівняння осіб.

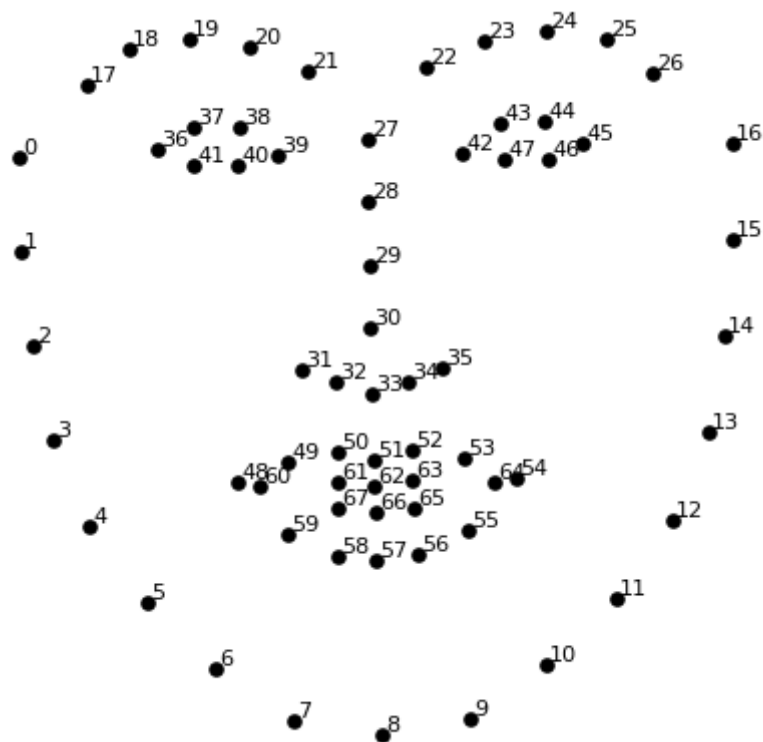


Рисунок 3.10 – Зображення 2D каркасу обличчя

Накладення каркасу виконується за допомогою (рис. 3.11)

`faceapi.nets.faceLandmark68Net.loadFromUri('/models/')`.

Існує проблема, коли обличчя, повернуті у різні сторони, тоді є верогідність того, що ці зображення система оцінить як різних осіб. Для вирішення проблеми перетворюються кожне зображення так, щоб очі і губи завжди знаходилися в якомусь конкретному місці. Для цього необхідно використовувати алгоритм під назвою оцінка орієнтирів особи (face landmark estimation) а саме метод Active Shape Models (ASM).

Суть методу ASM [16,19,19] полягає в обліку статистичних зв'язків між розташуванням антропометричних точок. На наявній вибірці зображень осіб, знятих в анфас, експерт розмічає розташування антропометричних точок. На кожному зображенні точки пронумеровані в однаковому порядку.

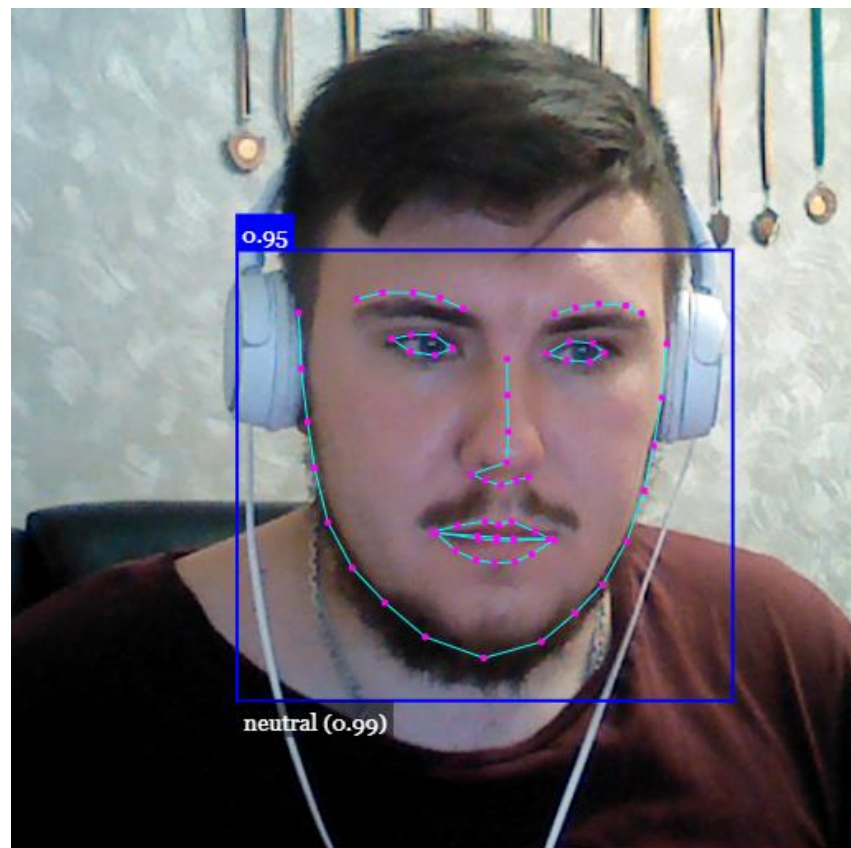


Рисунок 3.11 – 2D каркас у розробленій системі ідентифікації

Алгоритм Active Shape Model – це швидкий і надійний метод узгодження набору точок, керованих моделлю фігури, з новим зображенням. Параметри фігури для моделі, а також параметри, що визначають глобальну позу

(положення, орієнтацію та масштаб), визначають положення точок моделі на зображенні. Кожен крок в ітераційному підході до поліпшення напруженості точок, для анімації, включає спочатку вивчення області зображення навколо кожної поточної точки моделі, щоб знайти найкраще збіг поблизу, а потім оновити параметри для кращої моделі до нових знайдених точок. Це повторюється до збіжності. На практиці виконується пошук точки вздовж профілів, нормальних до межі моделі, через кожну точку моделі. Якщо очікується, що межа моделі відповідає ребру, можливо просто знайти найміцніший край (включаючи орієнтацію, якщо вона відома) уздовж кожного профілю і дати нове запропоноване місце для точки моделі.

3.4 Локальні моделі сірого рівня (Local Grey-level Models)

Точки моделей не завжди розташовуються на найсильнішому краю в місцевості – вони можуть повторювати слабший вторинний край або якусь іншу структуру зображення. Найкращий підхід – це навчитися з навчального набору, на що слід звертати увагу на цільовому зображенні. Це досягається шляхом вибірки вздовж профілю, нормального до межі в навчальному наборі, та побудови статистичної моделі структури рівня сірого [18]. Модель складається з оцінки середньої та коваріації рівнів сірого уздовж профілю. Під час пошуку робиться вибірка з пікселя профілю з будь-якої сторони поточної точки. Потім перевіряється якість відповідної моделі рівня сірого на кожному з можливих положень вздовж вибірки та вибираємо ту, яка дає найкращий збіг (найнижче значення). Це повторюється для кожної точки моделі, даючи запропоновану нову позицію для кожної точки. Потім оновлюються поточні параметри форми та форми, щоб найкращим чином відповідати моделі новим пунктам [15]. Для підвищення ефективності та надійності алгоритму ASM він реалізований в рамках множини з роздільною здатністю. До неї входить перший пошук об'єкта в крупному зображенні, потім фіксація місця розташування в серії послідовно

зображень з роздільною здатністю. Ефективність ASM, яка намагається зіставити модель кортикальних структур з новим зображенням. На ранніх стадіях цього розв’язання прийнято грубі рішення, що дозволяють великі рухи.

Моделі активних фігур здійснюють пошук навколо поточного місцезнаходження, уздовж профілів, тому можна очікувати, що вони матимуть більший діапазон захоплення, ніж AAM, який досліджує лише зображення безпосередньо під його поточною площею. Це чітко продемонстровано в результатах набору даних мозку (рис. 3.12).

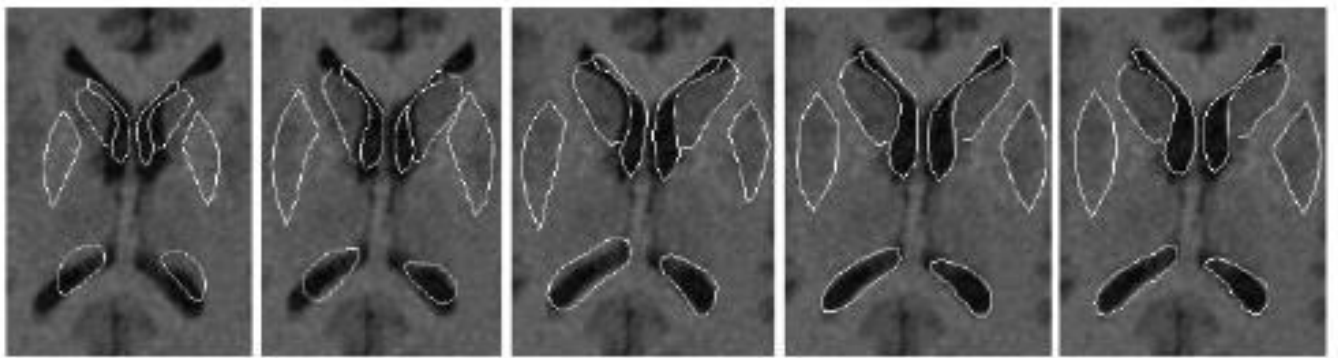


Рисунок 3.12 – Пошук із роздільною здатністю ASM на новому зображенні

ASM використовують лише дані навколо точок моделі і не користуються усією інформацією на рівні сірого, доступною в об'єкті, як це робить AAM. Таким чином, вони можуть бути менш надійними. Однак точки моделей, як правило, є цікавими місцями (межі або кути), де є найбільше інформації. Можна навчити AAM здійснювати пошук лише з використанням інформації в районах поблизу сильних кордонів – для цього знадобиться менше вибірки зображень під час пошуку, тому потенційно швидший алгоритм. Більш офіційний підхід полягає у вивченні з навчального набору, які пікселі є найбільш корисними для пошуку – це було досліджено в [14]. Результат пошуку є швидшим, але, як правило, менш надійним. Однією з переваг AAM є те, що можна створити переконливу модель із відносно невеликою кількістю орієнтирів. Будь-яка додаткова зміна форми виражається в додаткових режимах текстурної моделі.

ASM потрібні точки навколо кордонів, щоб визначити підходящі напрямки для пошуку. Через значну роботу, необхідну для отримання надійного маркування зображень, чим менше потрібно орієнтирів, тим краще. Алгоритм ААМ покладається на пошук лінійної залежності між зміщеннями параметрів моделі та індукованим вектором помилки текстури. Однак можливо збільшити вектор помилок за допомогою інших вимірювань, щоб надати алгоритму більше інформації. В конкретному методі комбінування ASM та ААМ передбачається пошук уздовж профілів у кожній точці моделі та збільшення вектора текстурної помилки з відстанню вздовж кожного профілю найкращого збігу. Як і помилка текстури, це має бути доведено до нуля для гарного збігу [18]. Цей підхід буде предметом подальшого дослідження та розробки системи ідентифікації, зображені на рис. 3.13, рис. 3.14.

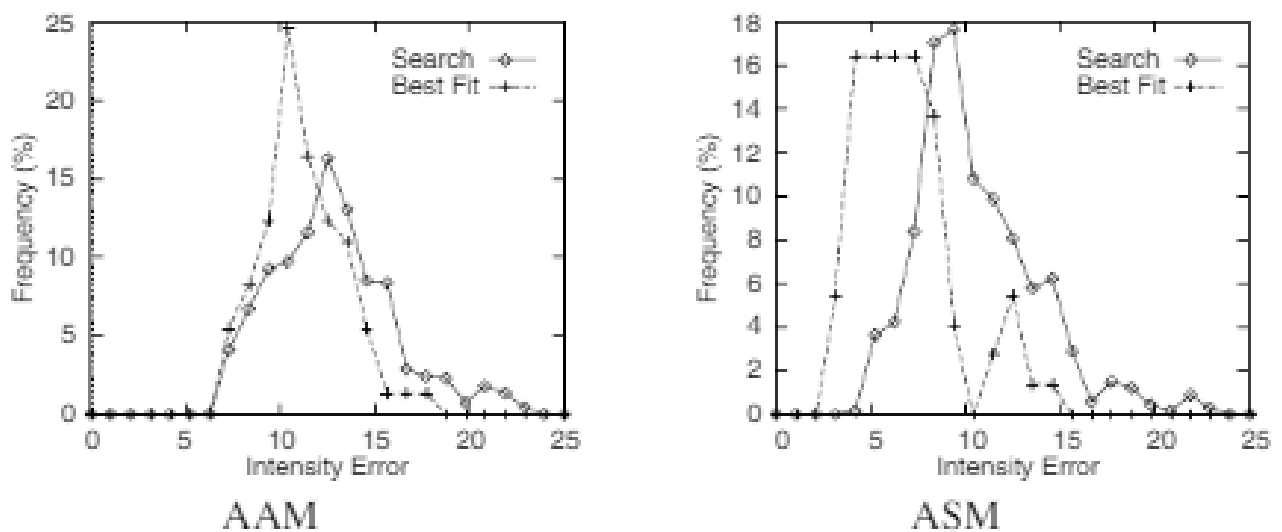


Рисунок 3.13 – Гістограми середньоквадратичних помилок текстури після пошуку зі зміщених позицій [15]

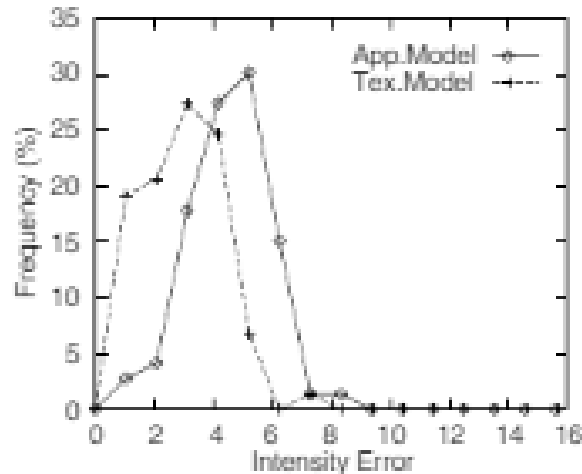


Рисунок 3.14 – Порівняння між фактурною моделлю найліпшого збігу та моделлю зовнішнього виду найліпшого збігу [15]

Точковий каркас особи відображається по різному в залежності від об'єктивних і суб'єктивних факторів.

Об'єктивні чинники - стан особи відносно камери.

Суб'єктивні чинники - нерівномірне або слабе освітлення, спотворення обличчя внаслідок емоцій, прикурювання ока і т.п. У цих випадках точковий каркас може бути некоректним, точки можуть навіть бути відірвані від імені.

Розпізнавач обличчя - це неймережа глибокого навчання, що використовує для обчислення унікальних дескрипторів осіб згадані вище моделі [6]. Навчити розпізнавач можливо на масиві тегірованих зображень осіб, після чого він зможе позначати особи на вхідному зображенні

3.1 Опис методів програмного забезпечення

Систему реалізовано за допомогою Visual Studio Code. Це кросплатформений редактор вихідного коду, розроблений Microsoft (рис.3.16). Позиціонується як «легкий» редактор кода для кросплатформенної розробки WEB та Cloud додатків.

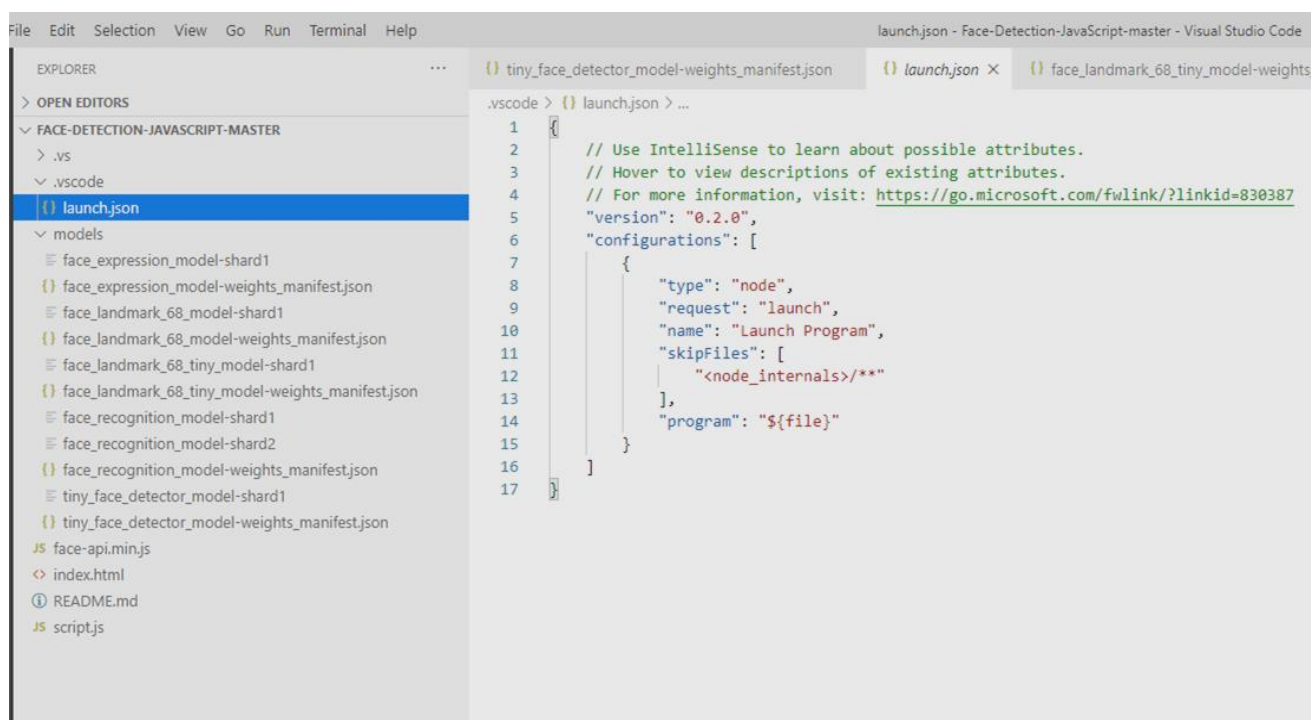


Рисунок 3.15 – Зображення робочої зони Visual Studio Code

Запуск програми здійснюється за допомогою компілятора Go Live.

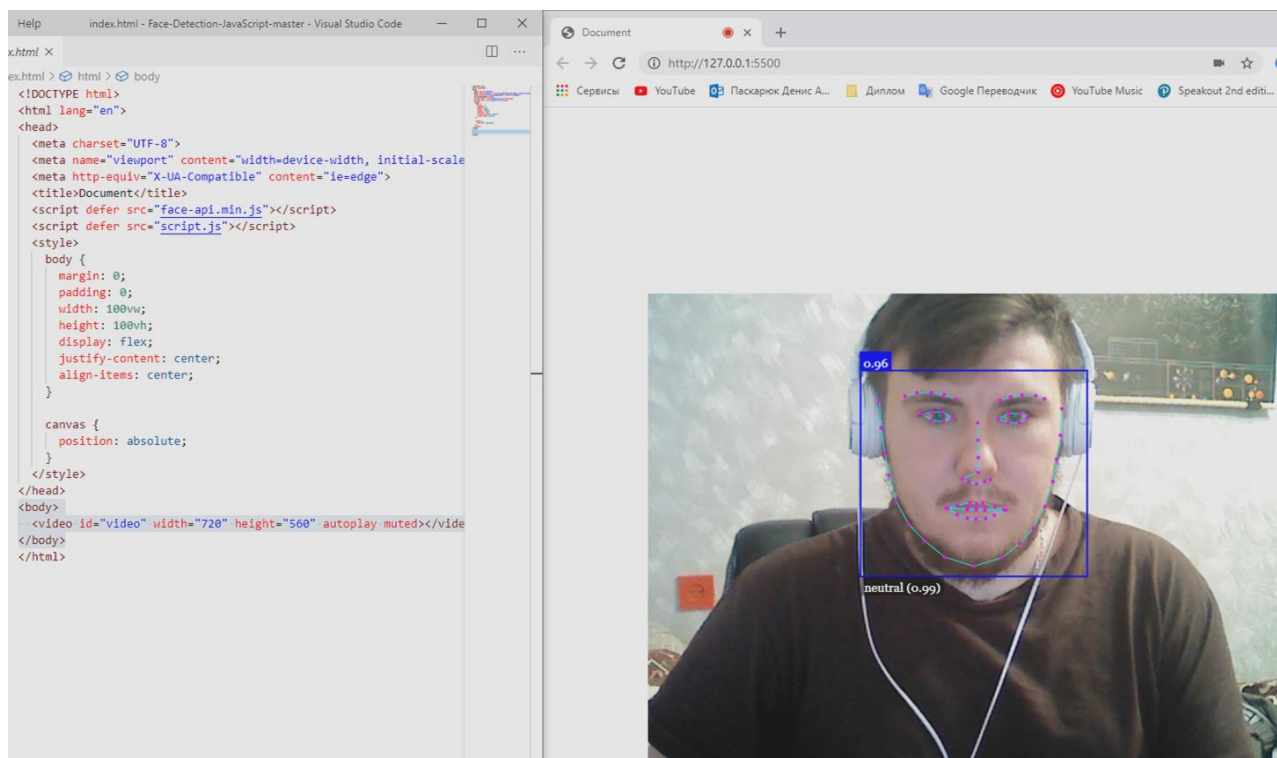


Рисунок 3.16 – Відображення запуску програми ідентифікації

Програмний додаток відкривається на відокремленому локальному серверу браузері, під назвою port 5500 та має вид браузерної сторінки (рис. 3.17).

Програмний додаток складається з таких бібліотек:

- .vs;
- .vscode;
- models;
- face-api.min.js;
- script.js.
- index.html.

Бібліотеки .vs та .vscode є стандартними бібліотками VS Code, та дозволяють робочому простору функціонувати за допомогою стандартних методів.

Бібліотека models зберігає у собі такі методи (рис.3.20):

- faceapi.nets.tinyFaceDetector.loadFromUri('/models');
- faceapi.nets.faceLandmark68Net.loadFromUri('/models');
- faceapi.nets.faceRecognitionNet.loadFromUri('/models');
- faceapi.nets.faceExpressionNet.loadFromUri('/models').

Бібліотека face-api.min.js зберігає у собі методи опису навчання розпізнавання нейромережею об'єктів у просторі камери.

Файл index.html є виконуючим файлом, який зберігає у собі структуру програми (рис. 3.17).

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8   <script defer src="face-api.min.js"></script>
9   <script defer src="script.js"></script>
10  <style>
11    body {
12      margin: 0;
13      padding: 0;
14      width: 100vw;
15      height: 100vh;
16      display: flex;
17      justify-content: center;
18      align-items: center;
19    }
20
21    canvas {
22      position: absolute;
23    }
24  </style>
25 </head>
26 <body>
27   <video id="video" width="720" height="560" autoplay muted></video>
28 </body>
29 </html>

```

Рисунок 3.17 – Зображення файлу index.html

Файл script.js у собі має усі методи, які необхідні при розпізнаванні об'єктів за допомогою нейромережі (рис. 3.18), а саме:

```

const video = document.getElementById('video')

Promise.all([
  faceapi.nets.tinyFaceDetector.loadFromUri('/models'),
  faceapi.nets.faceLandmark68Net.loadFromUri('/models'),
  faceapi.nets.faceRecognitionNet.loadFromUri('/models'),
  faceapi.nets.faceExpressionNet.loadFromUri('/models')
]).then(startVideo)

function startVideo() {
  navigator.getUserMedia(
    { video: {} },
    stream => video.srcObject = stream,
    err => console.error(err)
  )
}

video.addEventListener('play', () => {
  const canvas = faceapi.createCanvasFromMedia(video)
  document.body.append(canvas)
  const displaySize = { width: video.width, height: video.height }
  faceapi.matchDimensions(canvas, displaySize)
  setInterval(async () => {
    const detections = await faceapi.detectAllFaces(video, new faceapi.TinyFaceDetectorOptions()).withFaceLandmarks().withFaceExpressions()
    const resizedDetections = faceapi.resizeResults(detections, displaySize)
    canvas.getContext('2d').clearRect(0, 0, canvas.width, canvas.height)
    faceapi.draw.drawDetections(canvas, resizedDetections)
    faceapi.draw.drawFaceLandmarks(canvas, resizedDetections)
    faceapi.draw.drawFaceExpressions(canvas, resizedDetections)
  }, 100)
})

```

Рисунок 3.18 – Зображення файлу script.js

Метод опорних точки (Face Landmarks) – ідентифікує обличчя за допомогою набору точок, від 5 до 68.

Face-recognition.js. – метод дозволяє швидко і не надійно визначити осіб на зображенні використовуючи неймережу глибокого навчання або простий фронтальний розпізнавач.

Метод Promise.all відповідає за ідентифікацію обличчя, по моделям, які вказані у контейнерах Models (рис. 3.19).

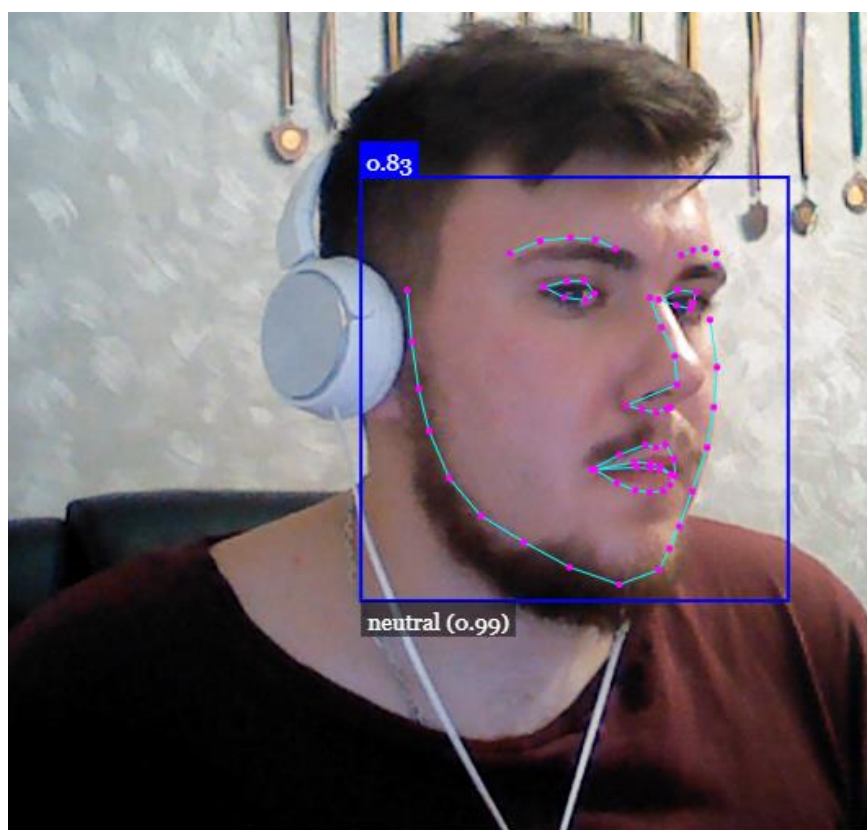


Рисунок 3.19 – Відображення точкового каркасу при впливі об'єктивних чинників

```
Promise.all([
  faceapi.nets.tinyFaceDetector.loadFromUri('/models'),
  faceapi.nets.faceLandmark68Net.loadFromUri('/models'),
```

```
faceapi.nets.faceRecognitionNet.loadFromUri('/models'),
    faceapi.nets.faceExpressionNet.loadFromUri('/models')
  ]).then(startVideo)

    faceapi.nets.faceRecognitionNet.loadFromUri('/models'),
```

використовується для встановлення положення обличчя, (рис. 3.20).

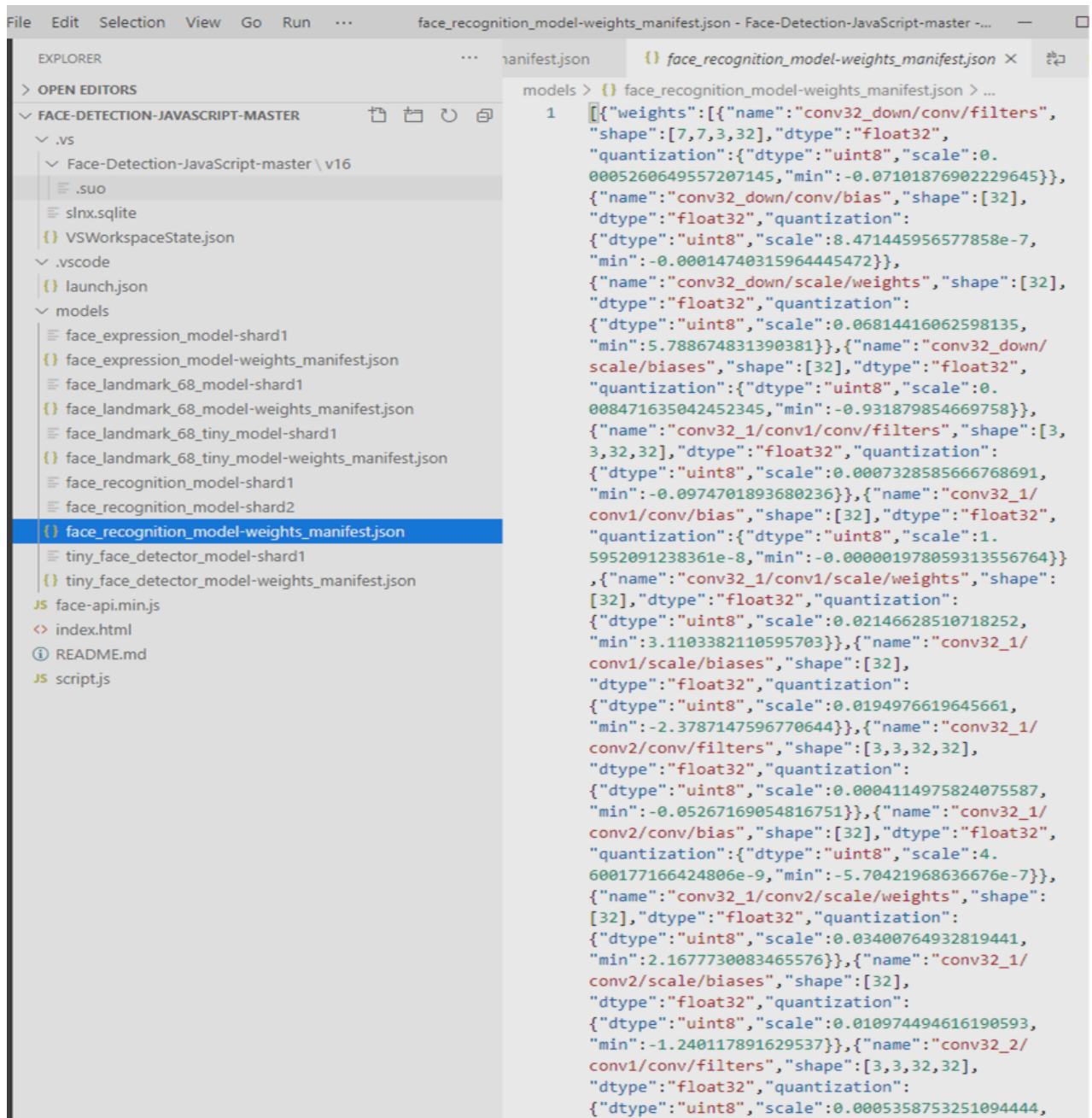


Рисунок 3.20 – Програмний опис реалізації 2D каркасу обличчя

`faceapi.nets.faceExpressionNet.loadFromUri('/models')` використовується для відстеження положення губ, та визначення емоцій.

Дані моделі ідентифікації дозволяють розпізнавати не лише одну людину, а усіх присутніх у об'єктиві камери.

Завантаження зображень у систему виконується за допомогою методу `const imageUpload = document.getElementById('imageUpload')`.

Після завантаження зображення, виконується порівняння моделей зображення, створюється пошук тестових (кінечних) моделей з їх початковим шаблонами. Це все описується у блоці `async function start()`.

Повний цикл навчання нейронної мережі описується в графічному представленні.

Завантаження зображень у систему виконується за допомогою `const imageUpload = document.getElementById('imageUpload')`.

Після завантаження зображення, виконується порівняння моделей зображення, створюється пошук тестових (кінечних) моделей з їх початковим шаблонами. Це все описується у блоці `async function start()`.

Папка `test_images` зберігає у собі тестові зображення, за допомогою яких наприкінці буде перевірка навчання системи.

Створення системи ідентифікації. Для того аби позначити обличчя на зображенні та ідентифікувати його, необхідно прописати метод `Canvas`

```
canvas {
```

```
  position: absolute;
```

```
  top: 0;
```

```
  left: 0;
```

```
}
```

Бібліотека `Face-api.min.js`

Цей репозиторій зберігає у собі усі необхідні дані, за допомогою яких відбувається ідентифікація особи на зображенні.

При відеозахоплення іноді проскакують такі зображення, на яких каркас з'їжджає, їх потрібно фільтрувати - як при навчанні так і розпізнаванні.

Деякі з точок є найбільш стабільними і інформативними. Вони жорстко прив'язані до лиця, незалежно від його положення щодо камери. Крім того, вони добре характеризують специфіку особи. Ці точки можуть бути використані як основа для моделювання системи ознак.

Для порівняння осіб можна використовувати точковий 2D каркас одного і того ж положення особи.

Всі ознаки (відстані) повинні бути безрозмірні (нормалізованості), тобто, співвіднесені до якогось розміру (відстані). Припускаю, що найбільш підходящий для цього розмір - відстань між центрами кутових точок очей.

Справа в тому, що кутові точки очей розсуваються (зближуються) при реагуванні на зміну кольору, вираженні подиву, моргання і т.п. Відстань між центрами очей нівелює ці коливання і тому більш переважно.

Програмний додаток відкривається на відокремленому локальному серверу браузері, під назвою port 5500 та має вид браузерної сторінки.

За основу необхідно взяти відстань від верхньої точки перенісся до нижньої точки підборіддя.

Перш ніж формувати ознаки для навчання і порівняння, необхідно відфільтрувати отримані відеозахоплення точкові каркаси осіб, які з суб'єктивних чи об'єктивних причин не є правильне фронтальне зображення особи (анфас).

Необхідно залишити тільки ті точкові каркаси, які проходять за такими ознаками (рис.3.21).

Пряма, яка проходить через крайні точки очей (лінія очей), перпендикулярна прямій, яка проходить через крайні точки носа (лінія носа).

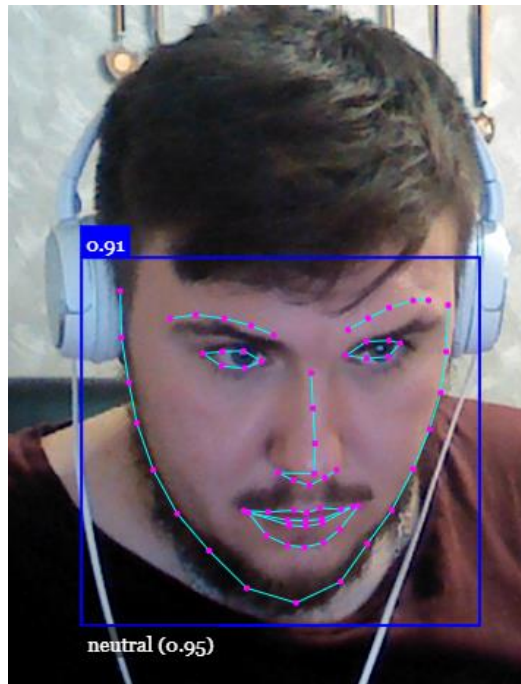


Рисунок 3.21 – Відображення точкового каркасу при впливі суб'єктивні чинників

Лінія очей паралельна прямій, яка проходить через точки куточків рота (лінія рота).

Дотримується симетрія зазначених вище точок щодо лінії носа.

Кутові точки очей (зовнішні і внутрішні) знаходяться на одній прямій.

3.6 Алгоритм роботи системи розпізнавання

Система ідентифікації реального часу реалізує свою роботи за такими етапами:

- система запускається у вигляді шаблону HTML сторінки у браузері, має роздільну здатність у 720 пікселей на 560 пікселей;
- для запуску відеопотоку викликається метод video;

```
<body>
```

```
<video id="video" width="720" height="560" autoplay muted></video>
```

```
</body>
```

- після спрацювання конструкції `video` вебкамера запускається автоматично;
- наступним кроком є спрацювання файлу основного файлу системи `script.js`;
- для передавання та аналізу дій об'єкта у камері реалізовано функцію `startVideo`.

Важливе зауваження, відео може мати затримку, це пов'язано з обчислювальною потужністю комп'ютера та вибором браузера.

```
function startVideo() {
  navigator.getUserMedia(
    { video: {} },
    stream => video.srcObject = stream,
    err => console.error(err)
  )
}
```

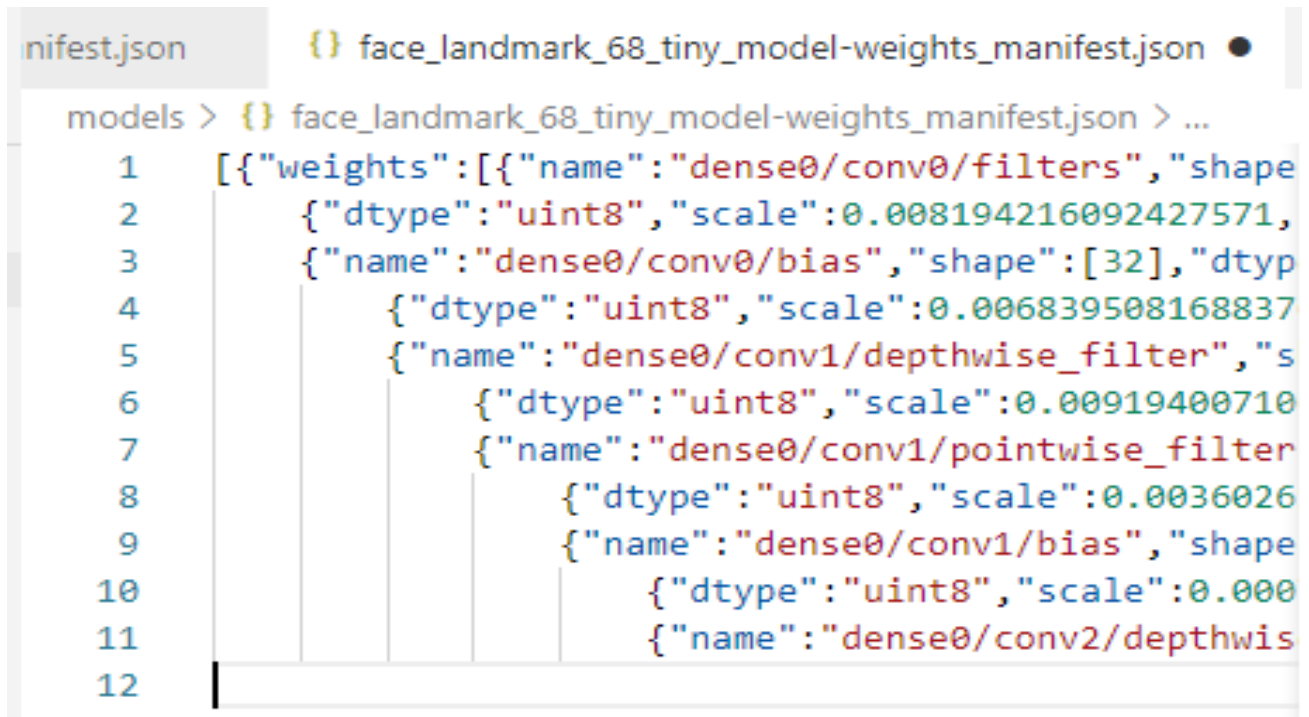
Завдяки цій функції ідентифікація об'єкта, а саме обличчя, створюється у `face-api.js`. Це процес є асинхронним, та виконується у методі `Promise`.

```
Promise.all([
  faceapi.nets.tinyFaceDetector.loadFromUri('/models'),
  faceapi.nets.faceLandmark68Net.loadFromUri('/models'),
  faceapi.nets.faceRecognitionNet.loadFromUri('/models'),
  faceapi.nets.faceExpressionNet.loadFromUri('/models')
]).then(startVideo)
```

Слід розглянути детальніше моделі розглянуті у методі `Promise`.

Модель `tinyFaceDeitector` ініціалізується перша та виконує аналіз об'єктів у просторі камери `faceapi.nets.tinyFaceDetector.loadFromUri('/models')`.

Модель `faceLandmark68Net` (рис. 3.22) реалізує оцінку відстані між ключовими точками обличчя за методом із 68 точок.



```

manifest.json  {} face_landmark_68_tiny_model-weights_manifest.json ●
models > {} face_landmark_68_tiny_model-weights_manifest.json > ...
1  [{"weights":[{"name":"dense0/conv0/filters","shape
2      {"dtype":"uint8","scale":0.008194216092427571,
3      {"name":"dense0/conv0/bias","shape":[32],"dtyp
4      {"dtype":"uint8","scale":0.006839508168837
5      {"name":"dense0/conv1/depthwise_filter","s
6      {"dtype":"uint8","scale":0.00919400710
7      {"name":"dense0/conv1/pointwise_filter
8      {"dtype":"uint8","scale":0.0036026
9      {"name":"dense0/conv1/bias","shape
10     {"dtype":"uint8","scale":0.000
11     {"name":"dense0/conv2/depthwis
12

```

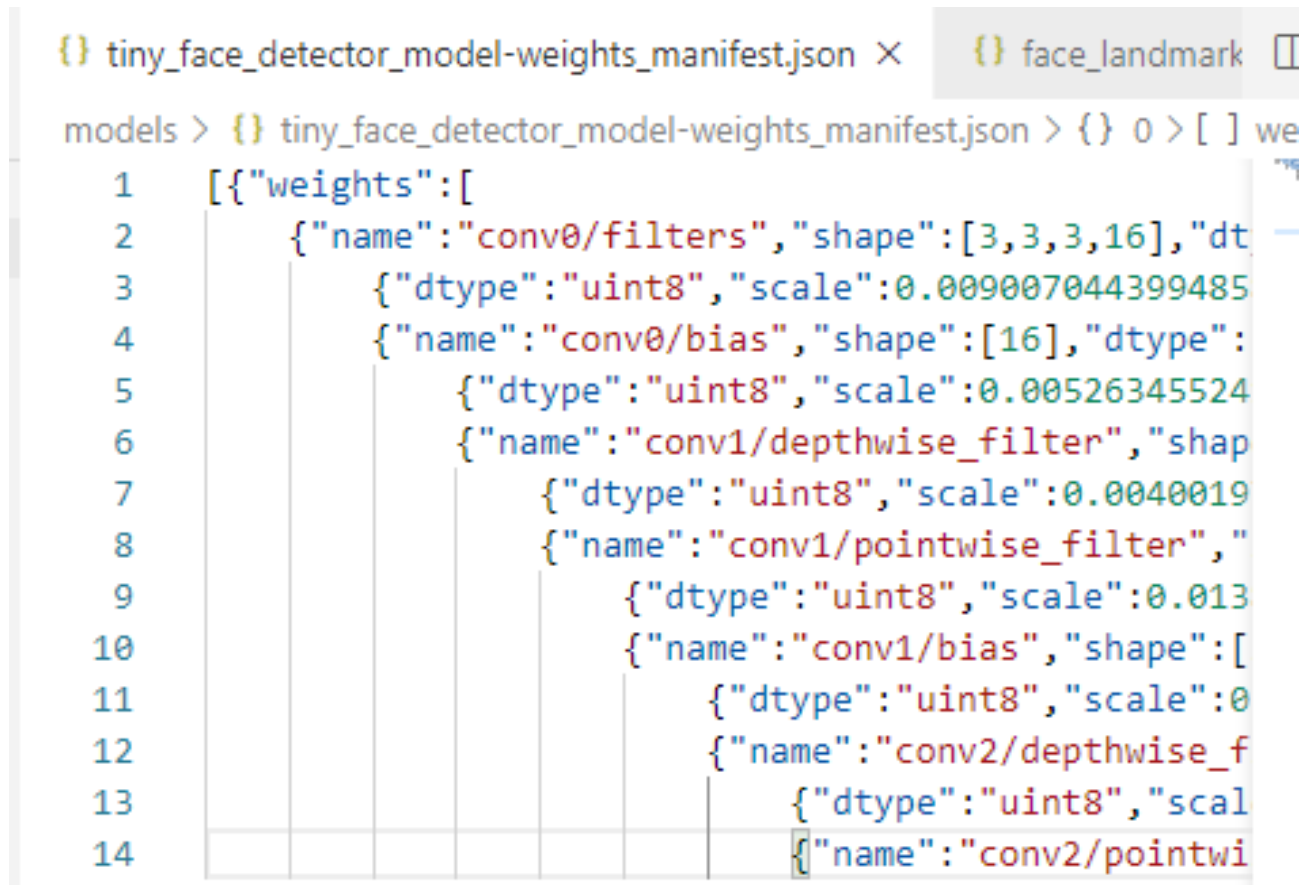
Рисунок 3.22 – Відображення вагів моделі faceLandmark68Net

Оцінки створюється за допомогою коефіцієнтів `weights` `faceapi.nets.faceLandmark68Net.loadFromUri('/models')`.

Метод `faceRecognitionNet` ініціалізується у режимі потоку відео (онлайн у момент включення послідовно двох попередніх моделей), та виконує аналіз положення обличчя у просторі та відповідно цього разом із обличчям переміщує шаблон із точок моделі `faceLandmark68Net` `faceapi`.

Метод `faceExpressionNet` виконує обчислення положення губ та виразу обличчя (поведінки та настрою), це виконується за допомогою вагів методу `faceapi.nets.faceExpressionNet.loadFromUri('/models')` (рис.3.23).

Останнім етапом є відображення рамки розпізнавання обличчя у виді синього квадрату, у якому є такі показники як, процент якості відображення (0,1 – 0,99) показник точності ідентифікації (0,1 – 0,99) необхідних параметрів, наприклад настрої (`neutral`, `happy`, `angry`, `adjusted`).



```

{} tiny_face_detector_model-weights_manifest.json × {} face_landmark {}
models > {} tiny_face_detector_model-weights_manifest.json > {} 0 > [ ] we
1 [{"weights": [
2   {"name": "conv0/filters", "shape": [3, 3, 3, 16], "dt
3     {"dtype": "uint8", "scale": 0.009007044399485
4     {"name": "conv0/bias", "shape": [16], "dtype":
5       {"dtype": "uint8", "scale": 0.00526345524
6       {"name": "conv1/depthwise_filter", "shap
7         {"dtype": "uint8", "scale": 0.0040019
8         {"name": "conv1/pointwise_filter", "
9           {"dtype": "uint8", "scale": 0.013
10          {"name": "conv1/bias", "shape": [
11            {"dtype": "uint8", "scale": 0
12            {"name": "conv2/depthwise_f
13              {"dtype": "uint8", "scal
14              {"name": "conv2/pointwi

```

Рисунок 3.23 – Відображення вагів моделі faceLandmark68Net

За останній рік, у режимі всесвітньої паднеї тенденцією та великою необхідністю є розпізнавання обличчя, яке скрито за маскою.

Розроблена система дозволяє розпізнавати обличчя з маскою, проте точність (рис. 3.24) та якість все ж залишаються питанням, яке у подальшому буде вирішено з модернізацією системи, та впровадженням до системи нових необхідних параметрів.

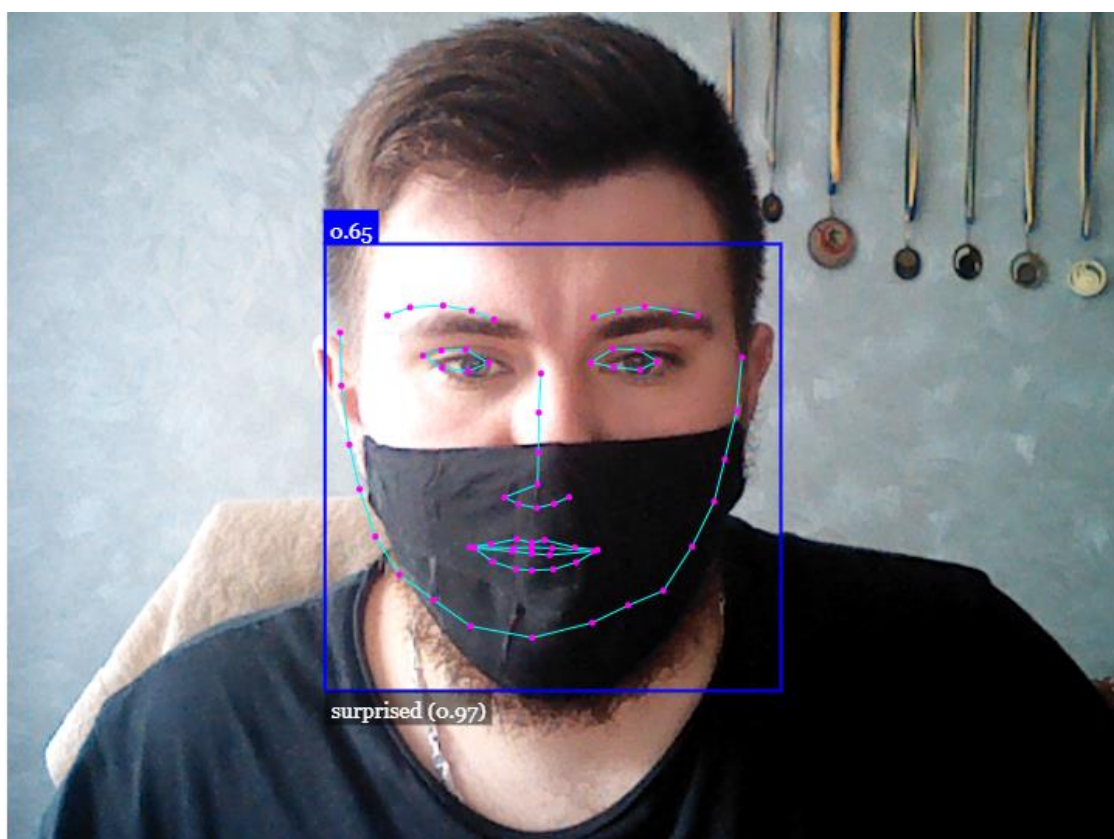


Рисунок 3.24 – Зображення ідентифікації обличчя у масці

3.7 Порівняння систем ідентифікації у реальному часі та роздільної ідентифікації

Слід розглядати як систему ідентифікації у реальному часі, та роздільної ідентифікації. У реальному часі система накладає 2D каркас на обличчя людини та «каркас розпізнавання» у якому міститься необхідна для системи інформація, та виконує пошук обличчя серед шаблонів навчання.

Кажучи про систему роздільної ідентифікації, мається на увазі, що, система бере зображення із бази даних або репозиторію (онлайн чи офлайн), проводить пошук по цим даним, та після цього виводить дані про об'єкт у виді сторінки браузера (рис. 3.25).

Обидві системи мають як переваги так і недоліки.

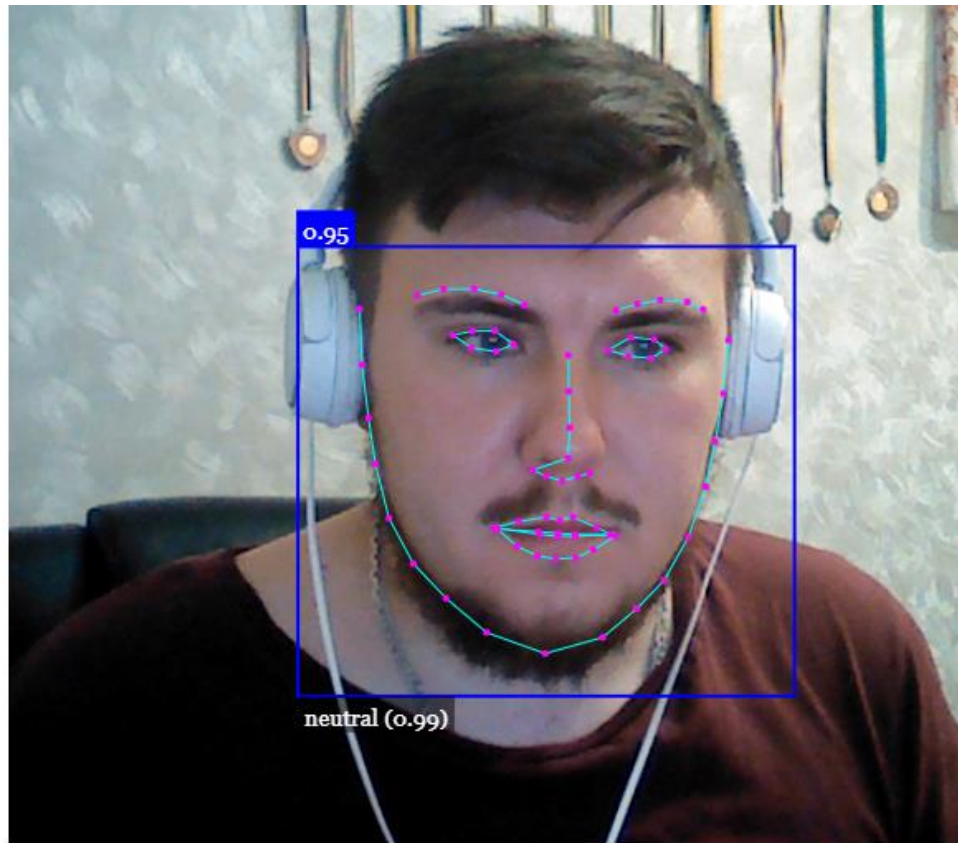


Рисунок 3.25 – Системи ідентифікації у реальному часі

Система ідентифікації у реальному часі має такі переваги:

- розпізнавання обличчя або об'єкта в момент його знаходження у полі зору системи;
- швидкість розпізнання;
- можливість ідентифікації навіть в умовах нечіткого положення об'єкта;
- визначення характеристик об'єкта у певний момент часу (настрій, температура, вид і т.д.).

Недоліки системи реального часу:

- великі втрати якості розпізнавання;
- похибки при обчисленні (у виді коефіцієнта від 0,1 до 0,99);
- хибна ідентифікація об'єкта та його характеристик;
- система повинна діяти в умовах чіткої ідентифікації (освітлення, положення відносно потоку об'єктів, інше).

Переваги системи роздільної ідентифікації:

- велика точність розпізнавання об'єктів завдяки глибокому навчанні (у порівнянні з системою реального часу);

- можливість ідентифікації навіть в умовах нечіткого положення об'єкта;
- немає залежності від умов середовища.

Недоліки системи роздільної ідентифікації:

- час виконання операцій;
- необхідність мати сховище шаблонів розпізнавання;
- на слабких комп'ютерах працює повільно.

В залежності від виду необхідної ідентифікації, слід обирати систему того чи іншого методу. Для рішення задач ідентифікації персоналу підприємств, слід вибрати систему ідентифікації у реальному часі. Тому що, в даному випадку, система повинна мати швидке реагування та можливість в момент часу показати максимальну кількість якісних характеристик об'єкта. Наприклад, ім'я об'єкта, температуру, настрій та інше.

3.8 Охорона праці під час виконання проекту

Базовий показник для визначення складових витрат праці розраховується за формулою (3.1)

$$T = T_0 + T_\delta + T_a + T_n + T_{\text{н}} + T_{\text{док}}, \quad (3.1)$$

де T – загальні витрати труда, люд. годин;

T_0 – витрати труда на опис завдання;

T_δ – витрати на розгляд предметної області;

T_a – витрати на розробку алгоритму рішення задачі;

T_n – витрати на програмування;

$T_{\text{н}}$ – витрати на налаштування програми;

$T_{\text{док}}$ – витрати на підготовку документації.

Всі складові визначаються через умовну кількість операторів Q :

$$Q = q \cdot c \cdot (1 + p), \quad (3.2)$$

де q – кількість операторів;

c – коефіцієнт складності завдання (приймається від 1,25 до 2);

p – коефіцієнт корекції програми, який враховує новизну проекту (для абсолютно нової програми дорівнює 0,1).

Візьмемо кількість операторів 23. З них 19 – кількість студентів, 4 – викладачів і інженерів-лаборантів, які будуть використовувати даний програмний продукт в навчанні (лабораторній практиці). Коефіцієнт складності програми приймемо за 1,5, тому що програмне забезпечення середньої складності.

Програма нова, тому коефіцієнт корекції програми візьмемо 0,1.

Підставивши отримані дані в формулу (3.2), отримаємо:

$$Q = 23 \cdot 1,5 \cdot (1 + 0,1) = 38. \quad (3.3)$$

Залежно від складності продукту T_0 візьмемо діапазон від 1 люд. год. до 5 люд. год. Програмний продукт середньої складності, тому T_0 візьмемо 3 люд. год.

T_∂ визначається за формулою (3.4):

$$T_\partial = (Q \cdot B) / (S_\partial \cdot k), \quad (3.4)$$

де B – коефіцієнт збільшення витрат роботи внаслідок недостатнього опису завдання (від 1,2 до 1,5);

S_o – кількість операторів, яка виділяється на 1 люд. год;

k – коефіцієнт кваліфікації працівника.

Таким чином, з огляду на стаж працівника до 2-х років, приймаємо коефіцієнт 0,8.

$$T_o = (38 \cdot 1,2) \div (1 \cdot 0,8) = 57 \text{ люд. год}, \quad (3.5)$$

T_a визначається за формулою (3.6):

$$\begin{aligned} T_a &= Q / (S_a \cdot k), \\ S_a &= 1 - 2, \end{aligned} \quad (3.6)$$

де S_a – кількість операторів, зайнятих розробкою блок-схеми, на 1 люд. год.

$$T_a = 38 / (2 \cdot 0,8) = 24 \text{ люд. год}. \quad (3.7)$$

Витрати на програмування розраховуються за формулою (3.8):

$$\begin{aligned} T_n &= Q / (S_n \cdot k), \\ S_n &= 1 - 2, \end{aligned} \quad (3.8)$$

де S_n – кількість операторів, зайнятих програмуванням, на 1 люд. год.

$$T_n = 38 / (2 \cdot 0,8) = 24 \text{ люд. год}. \quad (3.9)$$

Витрати на настройку програми розрахуємо наступним чином:

$$T_n = Q/(S_n \cdot k), \quad (3.10)$$

$$S_n = 1 - 2,$$

де S_n – кількість операторів, зайнятих налаштуванням, на 1 люд. год.

Витрати на підготовку документації знаходять за формулою (3.11):

$$T_{док} = T_{др} + T_{док}, \quad (3.11)$$

де $T_{др}$ – витрати праці на підготовку матеріалу рукописи знаходять за формулою (3.12):

$$T_{др} = Q/(S_{др} \cdot k), \quad (3.12)$$

$$S_{др} = 1 - 2,$$

де $S_{др}$ – кількість операторів, зайнятих підготовкою матеріалів в рукописі, на 1 люд. год.

$$T_{др} = 38/(1 \cdot 0,8) = 24 \text{ люд. год}, \quad (3.13)$$

$T_{док}$ – витрати праці на редагування, роздруківку і оформлення документів.

$T_{док}$ визначається за формулою (3.14):

$$T_{док} = 0,75 \cdot T_{др} = 0,75 \cdot 24 = 18 \text{ люд. год}. \quad (3.14)$$

$$T = 38 + 57 + 24 + 24 + 24 + 18 = 185 \text{ люд. год}. \quad (3.15)$$

Отримане значення загальної трудомісткості необхідно відредагувати з урахуванням рівня мови програмування:

$$T_{кор} = T \cdot k_{кор} = 185 \cdot 0,9 = 167 \text{ люд. год.} \quad (3.16)$$

де $k_{кор}$ – коефіцієнт, який враховує рівень мови програмування (від 0,8 до 1).

Час роботи персонального комп'ютера (фонд часу) під час створення програмного продукту визначається за формулою (3.18):

$$\Phi_v = 1,15 \cdot (T_{\Pi} + T_{Д} + T_{Н}) \cdot k_{кор}, \quad (3.17)$$

$$\Phi_v = 1,15 \cdot (24 + 24 + 24) \cdot 0,9 = 75 \text{ люд. год.},$$

де 1,15 – коефіцієнт, який враховує витрати часу на профілактичні роботи.

3.9 Висновки до третього розділу

Під час проектування системи ідентифікації слід враховувати підзадачі аналізу зображення на приступність об'єктів, навчання розпізнавання окремої особи та її параметрів.

Реалізація розпізнавання створюється за допомогою HOG перетворення, за принципом перетворення зображення на монохром та заміни пікселей на градієнти.

У даному розділі розглянуто проектування системи ідентифікації по методу накладення каркасу з 68 ключових точок обличчя. Описаний метод реалізовано за допомогою javascript та алгоритму Active Shape Models (ASM). Даний алгоритм дозволяє нейромережі швидко та надійно навчатися, розпізнавати та керувати положенням обличчя, ініціалізовувати параметри, які необхідні під час ідентифікації персоналу підприємства.

ВИСНОВКИ

Здатність до навчання є невід’ємною частиною властивості нейронних мереж. Навчання посилює працездатність системи у відповідності до задач та правил, які покладені у основу мережі. Нейромережа навчається за принципом інтерактивного процесу корегування синаптичних ваг нейронів. Робота мережі має такий вид: вона повинна отримувати знання із навколишнього середовища під час кожної ітерації процесу свого навчання.

У магістерській роботі розглянуто особливості актуальних методів ідентифікації, а також принципів навчання нейромереж, докладно описано головні частини згорткових нейромереж.

Проаналізовано актуальні архітектури нейромережових систем, методи навчання та засоби покращення характеристик систем.

Розроблена програмна реалізація нейронної мережі ідентифікації на основі технології нейромережової системи реального часу. Для цього використано Face-API.js та бібліотеки, що дозволяють інженерам проектувати архітектури мереж та здійснювати їх налаштування.

Результатом розробки є нейромережова система реального часу. Ця система була розглянута разом із системою роздільної ідентифікації. Під час порівняння проаналізовані характеристики обох систем та у результаті для поставленої задачі було зроблено висновок та обрано систему нейромережової системи реального часу, за такими критеріями:

- розпізнавання обличчя або об’єкта в момент його знаходження у полі зору системи;
- швидкість розпізнавання;
- можливість ідентифікації навіть в умовах нечіткого положення об’єкта;
- визначення характеристик об’єкта у певний момент часу.

Слід зазначити, що система може бути модифікована у разі появи нових завдань ідентифікації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. - К.: ДП «УкрНДНЦ», 2016. - 31 с.
2. Невлюдов, І.Ш. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» [Текст]: навч. посіб. / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. – Київ – 58, пр. Космонавта Комарова, 1, 2016. – 320 с.
3. Методичні вказівки «З розробки й оформлення магістерської атестаційної роботи для студентів другого (магістерського) рівня вищої освіти галузі знань 15 Автоматизація та приладобудування за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології освітні програми: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» [Текст] / Упоряд. І.Ш. Невлюдов, В.В. Косенко, В.В. Євсєєв. – Харків: ХНУРЕ, 2019. – 55 с.
4. Положення про протидію академічному плагіату в ХНУРЕ [Електронний ресурс]. – Режим доступу: https://nure.ua/wp-content/uploads/Main_Docs_NURE.html. – (дата звернення: 29.09.2020).
5. Невлюдов, І.Ш. Основи наукових досліджень [Текст]: навч. посібник / І.Ш. Невлюдов, Ю.М. Олександров, А.О. Андрусевич, О.О. Чала. – Кривий Ріг: Криворізький коледж НАУ, 2019. – 396 с.
6. Невлюдов, І.Ш. Техніко – економічне обґрунтування інженерних рішень в автоматизованому виробництві [Текст]: підручник / І.Ш. Невлюдов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 448 с.
7. Комарцова, Л.Г. Нейрокомпьютеры [Текст] / Л.Г. Комарцова, А.В. Максимов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с.

8. Srivastava, N. Dropout: a simple way to prevent neural networks from overfitting. [Text] / N. Srivastava, G. E. Hinton // Journal of Machine Learning Research. – 2014. – Vol. 15. – № 1. – P. 1929–1958.
9. Методи робастного, нейро-нечеткого и адаптивного управления [Текст] / Под ред. Н.Д. Егупова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 744 с.
10. Бодянський, Є. В. Нейро – фаззі моделі в системах штучного інтелекту [Текст] / Є. В. Бодянський, Є. І. Кучеренко. – Харків: ХНУРЕ, 2006. – 177 с.
11. Руденко, О. Г. Основы теории искусственных нейронных сетей [Текст] / О.Г. Руденко, Е. В. Бодянский. – Харьков: Телетех, 2002. – 317 с.
12. Невлюдов, І.Ш. Розпізнавання голосових команд за допомогою багатопарового перцептрона [Текст] / І.Ш. Невлюдов, О.М. Цимбал, С.С. Мілютіна // Східно-європейський журнал передових технологій, Харків. – 2006. – № 3/2 (16). – С. 13–16.
13. Цимбал, А.М. Использование искусственной нейронной сети в подсистеме ввода голосовой информации САПР ТП роботизированного производства [Текст] / И.Ш. Невлюдов, А.М. Цимбал, С.С. Милютина // Радиоэлектроника и информатика, Харьков. – 2007. – № 1. – С. 56-61.
14. Каллан, Р. Основные концепции нейронных сетей [Текст] // Р. Каллан. – New York: Williams, 2001. – С. 288.
15. Bergstra, J. Random search for hyper-parameter optimization [Text] / J. Bergstra, Y. Bengio // Journal of Machine Learning Research. – 2012. – Vol. 13. – № 1. – P. 281–305.
16. Cootes, T. Active appearance models [Text] / T. Cootes, G. J. Edwards, C. J. Taylor, H. Burkhardt, B. Neumann // 5 th European Conference on Computer Vision, Springer, 1998. – Vol. 2, P. 484–498.
17. Aparna C Shastri Convolutional Neural Networks [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/convolutional-neural-networks-part-1-2aeb17fc208c.html>. – (дата звернення: 29.09.2020).

18. Гиниятуллин, В.М. Реализация бинарных решающих функций в нейросетевом базисе [Текст] / Ю.И. Зозуля, В.А. Муртазина, М.В. Фазылова // Нейрокомпьютеры. – 2006. – № 10. – С. 3–8.

19. Паскарюк, Д.О. Розпізнавання образів за допомогою нейронних мереж [Текст] / Збірник студентських наукових статей / Харківський національний університет радіоелектроніки; [редкол.: І.Ш. Невлюдов та ін.]. – 2020. – Вип. 2. – С. 74-77.

20. Семенець, В.В. Исследование зависимости максимального перегрева радиоэлектронного аппарата от его параметров [Текст] / В.В. Семенец, А.М. Синотин, Т.А. Колесникова // Системи обробки інформації. – 2018. – №4. – С. 29 –34.