

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ

Кваліфікаційна робота

другий (магістерський)

(рівень вищої освіти)

на тему Нейромережна система для розпізнавання емоцій на обличчі
людини

Виконав: студент 2 курсу, групи 8.1219-пзс
спеціальності 121 Інженерія програмного
забезпечення

(код і назва спеціальності)

освітньої програми Інженерія програмного
забезпечення

(код і назва освітньої програми)

А. О. Бублик

(ініціали та прізвище)

Керівник доцент, к. т. н. Ю. О. Лимаренко
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Ай Ті Діменшн»

В. С. Тряпичко

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2020

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Інженерний навчально-науковий інститут

Кафедра програмного забезпечення автоматизованих систем

Рівень вищої освіти другий (магістерський)

Спеціальність 121 Інженерія програмного забезпечення
(код та назва)

Освітня програма Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри В.Г. Вербицький
" 01 " вересня 2020 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Бублик Анна Олександрівна

(прізвище, ім'я, по батькові)

1. Тема роботи Нейромережна система розпізнавання емоцій на обличчі людини

керівник роботи Лимаренко Юлія Олексіївна, доцент, к. т. н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від "25" травня 2020 року № 600-с

2. Строк подання студентом кваліфікаційної роботи 30.11.2020

3. Вихідні дані магістерської роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

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

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження проблеми розпізнавання мов та розробка методів її вирішення;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- дослідження поставленої проблеми та розробка висновків та пропозицій.

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


6. Консультанти розділів магістерської роботи

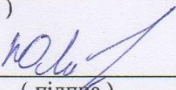
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.09.2020

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	02.09-09.09.20	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	10.09-11.09.2020	виконано
3	Аналіз існуючих методів рішення	12.09-15.09.19	виконано
4	Дослідження області розпізнавання обличчя та емоцій людини	16.09-23.09.19	виконано
5	Узгодження подальших дій з науковим керівником	23.09-24.09.19	виконано
6	Аналіз теоретичних відомостей	24.09-13.10.19	виконано
7	Проектування інтерфейсу системи розпізнавання емоцій	14.10-18.10.19	виконано
8	Узгодження інтерфейсу з науковим керівником	19.10-20.10.19	виконано
9	Реалізація функціоналу розпізнавання емоцій на обличчі людини	21.10-01.11.19	виконано
10	Представлення отриманих результатів науковому керівнику і узгодження плану подальшого дослідження	02.11-03.11.19	виконано
11	Реалізація функціоналу веб-застосунку	03.11-10.11.19	виконано
12	Проведення аналізу можливостей розробленого програмного застосунку	11.11-14.11.20	виконано
13	Оформлення звіту	14.11-29.11.20	виконано

Студент  Бублик А. О.
(підпис) (прізвище та ініціали)

Керівник роботи  Ю.О. Милосеренко
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер  І.А. Скрипник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок: 108

Рисунків: 36

Таблиць: 8

Джерел: 39

Бублик А. О. Нейромережна система розпізнавання емоцій на обличчі людини.

Кваліфікаційна робота для здобуття ступеня вищої освіти магістра за спеціальністю 121 – Інженерія програмного забезпечення, науковий керівник Ю. О. Лимаренко. Інженерний інститут ЗНУ. Факультет енергетики, електроніки та інформаційних технологій, 2020.

Мета роботи полягає у дослідженні та вивченні методів розпізнавання обличчя людини та підходів до аналізу емоцій людини, порівняння їх особливостей, перевірка можливостей застосування і створення нейромережної системи для розпізнавання емоцій людини в режимі реального часу.

Досліджено методи і конкуруючі сучасні системи розпізнавання емоцій з обличчя людини, їх проблематику і можливості розробки і використання системи. Порівняно методи виділення обличчя людини і методи розпізнавання емоцій людини. Спроектовано та реалізовано згорткову нейронну мережу на мові програмування Python. Створено веб-застосунок, що дозволяє розпізнавати емоції на обличчі людини в режимі реального часу завдяки створеній нейронній мережі.

Ключові слова: *ВИДІЛЕННЯ ОБЛИЧЧЯ, РОЗПІЗНАВАННЯ ЕМОЦІЙ, НЕЙРОННА МЕРЕЖА, ШТУЧНИЙ ІНТЕЛЕКТ, ГЛИБИННЕ НАВЧАННЯ, PYTHON, DJANGO, ВЕБ-ЗАСТОСУНОК*

SUMMARY

Pages: 109

Figures: 35

Tables: 8

Sources: 39

Bublyk A. O. Neural network system for recognizing emotions on a person's face.

Qualification work for higher master's degree in specialty 121 - Software Engineering, supervisor J. O. Lymarenko. Engineering Institute ZNU. Faculty of Energy, Electronics and Information Technology, 2020.

The aim of the research is to study the methods of human face recognition and approaches to the analysis of human emotions, comparing their features, testing the application and creating a neural network system for recognizing human emotions in real time..

Methods and competing modern systems of emotions recognition from the person facial expressions, their problems and possibilities of development and use of system are investigated. The methods of detecting a person's face and methods of recognizing human emotions are compared. A convolutional neural network in the Python programming language was designed and implemented. Created a web application that allows user to recognize emotions in real time due to the created neural network.

Keywords: *FACIAL DETECTION, EMOTION RECOGNITION, NEURAL NETWORK, ARTIFICIAL INTELLIGENCE, DEEP LEARNING, PYTHON, DJANGO, WEB APPLICATION*

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ	
ЛЮДИНИ	14
1.1 Комп'ютерний зір.....	14
1.2 Типові задачі комп'ютерного зору	16
1.3 Розпізнавання образів	17
1.3.1 Класифікація.....	17
1.4 Системи розпізнавання образів	18
1.5 Методи розпізнавання образів.....	19
1.6 Розпізнавання обличчя людини.....	20
1.7 Розпізнавання емоцій людини.....	21
1.7.1 Емоції людини	22
1.7.2 Розпізнавання емоцій з зображення	23
1.7.3 Розпізнавання емоцій у відео-поточі	24
1.7.4 Основні проблеми, пов'язані з розпізнаванням облич та емоцій	24
1.8 Аналіз існуючих систем розпізнавання емоцій	25
1.8.1 FaceReader.....	25
1.8.2 EmoDetect.....	27
1.8.3 FindFace Security.....	28
1.9 Результати аналізу існуючих рішень	30
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЗАСОБІВ РОЗПІЗНАВАННЯ ЕМОЦІЙ	
ЛЮДИНИ	31
2.1 Теорія розпізнавання обличчя людини	31
2.2 Методи розпізнавання облич	33
2.2.1 Метод Віюли-Джонса.....	33
2.2.2 Метод гнучкого порівняння на графах	36
2.2.3 Приховані Марковські моделі	39
2.2.4 Метод головних компонент	39
2.2.5 Активні моделі зовнішнього вигляду.....	41

2.3	Методи розпізнавання емоцій людини	43
2.3.1	Метод ключових точок	44
2.3.2	Згорткові нейронні мережі.....	46
2.3.3	Рекурентні нейронні мережі	48
2.4	Згорткові нейронні мережі для розпізнавання об'єктів з зображень	50
2.4.1	Архітектура AlexNet.....	50
2.4.2	Архітектура VGG16	52
2.4.3	Архітектура GoogLeNet	53
2.4.4	Архітектура ResNet-50	55
2.5	Датасети для розпізнавання емоцій на обличчі	57
2.6	Фреймворки глибинного навчання.....	59
2.7	Обґрунтування вибраних засобів для реалізації системи	64
РОЗДІЛ 3 ПРОЕКТ ПРОГРАМНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ НА ОБЛИЧЧІ ЛЮДИНИ		66
3.1	Архітектура системи	66
3.2	Засоби реалізації.....	68
3.2.1	Мова програмування Python	69
3.2.2	Редактор коду Visual Studio Code	69
3.2.3	HTML, CSS, JavaScript	70
3.2.4	Фреймворк Django.....	72
3.2.5	Протокол WebSockets.....	76
3.2.6	Бібліотека Django Channels	77
3.2.7	Бібліотека OpenCV	79
3.2.8	Фреймворк TensorFlow та Keras API.....	81
3.3	Модулі програмної системи	82
3.3.1	Клієнтська частина системи	82
3.3.2	Серверна частина системи	85
3.3.3	Архітектура нейронної мережі для розпізнавання емоцій людини ..	87
3.4	Проект інтерфейсу.....	92
3.5	Вимоги до апаратного забезпечення	93

3.6	Опис функціональних можливостей	94
РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ.....		95
4.1	Навчання та перевірка побудованої нейронної мережі.....	95
4.2	Вимірювання точності розпізнавання емоцій людини в залежності від статі.....	96
4.3	Вимірювання точності розпізнавання емоцій людини в залежності від віку	98
4.4	Вимірювання точності розпізнавання емоцій людини на частково закритому обличчі.....	100
ВИСНОВКИ		103
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		104

ВСТУП

Актуальність теми

З розвитком технологічного прогресу, термін «комп'ютерний зір» дуже часто зустрічається в сучасних дослідженнях. Комп'ютерний зір – це область науки, яка займається завданнями, пов'язаними з аналізом зображень і відео. За допомогою комп'ютерного зору можуть бути вирішені наступні проблеми: сегментація зображень, виділення об'єктів, класифікація зображень, відстеження рухомих об'єктів в часі, генерація зображень, розпізнавання облич та навіть емоцій.

Розпізнавання емоцій — важлива тема у сфері комп'ютерного зору та штучного інтелекту завдяки значному академічному та комерційному потенціалу. Сплеск суспільного і наукового інтересу до теми детекції та розпізнавання емоцій відбувся у другій половині 2000 – початок 2010-х років, пікових значень ці процеси досягли в 2017-2018 роках і їх показники продовжують збільшуватись й сьогодні.

Великий інтерес до теми розпізнавання емоцій зумовлений тим, що системи розпізнавання емоцій можуть бути застосовані в багатьох сферах людської діяльності. Особливо актуальне використання подібних систем для визначення стану людини у випадках, пов'язаних з небезпекою для її життя. В якості одного з прикладів можна привести системи розпізнавання втоми людини, якими оснащуються деякі сучасні автомобілі. Подібні системи дозволяють в багатьох випадках уникнути аварій, викликаних неуважністю, сонливістю або поганим самопочуттям водія.

Ще однією сферою застосування методів автоматичного розпізнавання емоцій є забезпечення безпеки людей за допомогою автоматизованих охоронних систем. Сучасні охоронні системи часто мають в своєму складі засоби реєстрації та аналізу відеоданих. Однак, як правило, в таких системах рішення приймається людиною-оператором. Це може привести до зниження точності і оперативності реагування на різні ситуації, пов'язані з поведінкою

людей на охоронюваних територіях. Підвищити ефективність охоронних систем можна за рахунок підвищення ступеня автоматизації процедур, що забезпечують аналіз зображень.

Розпізнавання емоцій застосовується також в цілому ряді інших областей, таких як телекомунікації, продажі й реклама, відеоігри, анімація, психіатрія, автоматизоване навчання, криміналістика і т.д.

Мета дослідження

Мета дослідження полягає в аналізі методів виділення обличчя людини, розпізнавання емоцій і також створення власної системи для розпізнавання емоцій та контролю емоційного стану користувача. Необхідно дослідити підходи виявлення емоцій на обличчі людини, їх збір та аналіз. Розроблена система повинна досліджувати емоції користувача в режимі реального часу та аналізувати його стан і транслювати дані аналізу.

Об'єкт дослідження

Відео потік, який система отримує з включеної веб-камери користувача.

Предмет дослідження

Аналіз та розпізнавання емоцій людини з відео потоку в режимі реального часу.

Методи дослідження

Для розв'язання поставлених задач використовуються такі методи:

- Вивчення джерел про нейронні мережі, способи розпізнавання та виділення обличчя людини за допомогою комп'ютерного зору.
- Аналіз існуючих методів розпізнавання міміки обличчя та систем, що використовують такі методи.
- Узагальнення отриманих результатів дослідження.

Наукова новизна одержаних результатів

Одержанні результати дослідження є відображенням актуальності проблеми розпізнавання емоцій та систем, які можуть виконувати такі процеси. В ході дослідження було виявлено переваги та недоліки використання нейронних мереж для побудови систем аналізу людських емоцій.

На основі отриманих даних можлива розробка повноцінного застосування, який буде розпізнавати емоції людини, аналізувати їх та робити висновки щодо стану здоров'я людини.

Практичне значення одержаних результатів

За допомогою одержаних результатів можна оцінити доцільність використання нейронних мереж для розроблення систем розпізнавання емоцій людини. Також зробивши аналіз створеної системи можна зрозуміти найкращий тип і архітектуру нейронної мережі для вирішення задач розпізнавання емоцій. Дослідити найпоширеніші проблеми у реалізації подібних систем, встановити основні емоції, які система може розпізнати без помилок.

Також у результаті дослідження було отримано прототип, який допомагає розпізнавати емоції людини в режимі реального часу.

Апробація результатів

Результати досліджень, викладені у кваліфікаційній роботі магістра були представлені на XXV науково-технічній конференції студентів, магістрантів, аспірантів, молодих вчених та викладачів [39] і опубліковані в збірнику наукових праць студентів, аспірантів і молодих вчених «Молода наука -2020» [38].

Глосарій

Глибинне навчання — це галузь машинного навчання, що ґрунтується на наборі алгоритмів, які намагаються моделювати високорівневі абстракції в даних, застосовуючи глибинний граф із декількома обробними шарами, що побудовано з кількох лінійних або нелінійних перетворень.

Машинне навчання — це підгалузь штучного інтелекту в галузі інформатики, яка часто застосовує статистичні прийоми для надання комп'ютерам здатності «навчатися» (тобто, поступово покращувати продуктивність у певній задачі) з даних, без того, щоби бути програмованими явно.

Нейронна мережа — математична модель, а також її програмне або апаратне втілення, побудована за принципом організації та функціонування біологічних нейронних мереж.

Згорткова нейронна мережа — спеціальна архітектура штучних нейронних мереж, запропонована Яном Лекуном в 1988 році і націлена на ефективне розпізнавання образів, входить до складу технологій глибокого навчання.

Комп'ютерний зір або Комп'ютерне бачення — теорія та технологія створення машин, які можуть проводити виявлення, стеження та визначення об'єктів.

Теорія розпізнавання образів — розділ кібернетики, що розвиває теоретичні основи й методи класифікації і ідентифікації предметів, явищ, процесів, сигналів, ситуацій і т. п. об'єктів, які характеризуються кінцевим набором деяких властивостей і ознак.

Емоції — це психічний процес, складний стан організму середньої тривалості, що відображає суб'єктивне оціночне ставлення до існуючих або можливих ситуацій та об'єктивного світу, призводить до дії симпатичної нервової системи і підвищення життєдіяльності організму.

Міміка — рухи м'язів обличчя, що виражають почуття і психічний стан людини.

Задача класифікації — формалізована задача, яка містить множину об'єктів (ситуацій), поділених певним чином на класи. Задана скінченна множина об'єктів, для яких відомо, до яких класів вони належать.

Розпізнавання облич — практичне застосування теорії розпізнавання образів, в завдання якого входить автоматична локалізація обличчя на фотографії і, в разі необхідності, ідентифікація персони по обличчю. Функцію ідентифікації людей на фотографіях вже активно використовують в програмному забезпеченні для керування фотоальбомами.

Відеопотік — це тимчасова послідовність кадрів певного формату, закодована в потік бітів.

Кадр — фрагмент відеоряду, окреме зображення або відрізок кіноплівки.

Сервер — комп'ютер, виділений з групи персональних комп'ютерів (або робочих станцій) для виконання будь-якої сервісної завдання без безпосередньої участі людини.

Клієнт — це апаратний або програмний компонент обчислювальної системи, який посилає запити серверу.

Шаблон проектування або патерн — повторювана архітектурна конструкція, що представляє собою рішення проблеми проектування в рамках деякого часто виникає контексту.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ

1.1 Комп'ютерний зір

Комп'ютерний зір — це автоматична фіксація і обробка зображень, як нерухомих, так і рухомих об'єктів за допомогою комп'ютерних засобів [1].

Як наукова дисципліна, комп'ютерний зір відноситься до теорії і технології створення штучних систем, які отримують інформацію з зображень. Відео дані можуть бути представлені безліччю форм, наприклад відеопослідовність, зображення з різних камер або тривимірні дані, наприклад з пристрою Kinect або медичного сканера.

Як технологічна дисципліна, комп'ютерний зір прагне застосовуючи теорії та моделі комп'ютерного зору створити системи комп'ютерного зору.

Перші спроби змусити комп'ютер «бачити» відносяться до початку 60-х років 20 століття [1]. Однак лише в останні роки в зв'язку з підвищенням обчислювальних потужностей і швидкодії процесорів, обсягів пам'яті, підвищенням роздільної здатності та інших параметрів камер, розвитком смуги пропускання каналів зв'язку, а також з появою таких технологій, як глибинне навчання, штучний інтелект, технології комп'ютерного зору стали знаходити все більше застосувань в різних галузях промисловості і повсякденному житті людей.

До появи глибокого навчання завдання, які міг виконувати комп'ютерний зір, були дуже обмежені і вимагали великої кількості ручного програмування і зусиль з боку розробників і людей-операторів. Наприклад, для вирішення задачі розпізнавання обличчя потрібно було виконати наступні кроки:

1. **Створення бази даних:** потрібно було зробити окремі зображення всіх об'єктів, які необхідно відстежувати, в певному форматі.

2. **Анотація зображень:** потім для кожного окремого зображення потрібно ввести декілька ключових точок даних, таких як відстань між очи-

ма, ширина перенісся, відстань між верхньою губою і носом і десятки інших вимірів, які визначають унікальні характеристики кожної людини.

3. Створення нових зображень: Потім потрібно створити нові зображення, будь то з фотографій або відеоконтенту. А потім необхідно знову пройти процес вимірювання, відзначаючи ключові точки на зображенні.

Після всієї цієї ручної роботи система, нарешті, зможе порівняти вимірювання на новому зображенні з тими, які зберігаються в її базі даних, і сказати вам, чи відповідає воно якомусь з відслідковуваних профілів. Фактично, автоматизації було дуже мало, і велика частина роботи виконувалася вручну. І похибка залишалася великою.

Машинне навчання надало інший підхід до вирішення проблем комп'ютерного зору. Завдяки машинному навчанню розробникам більше не потрібно вручну кодувати кожне правило в своїх системах машинного зору. Замість цього вони запрограмували «функції», невеликі додатки, які могли виявляти певні закономірності на зображеннях. Потім вони використовували алгоритм статистичного навчання, такий як лінійна регресія, логістична регресія, дерева рішень або допоміжні векторні машини, для виявлення шаблонів і класифікації зображень і виявлення в них об'єктів.

Глибоке навчання надало принципово інший підхід до машинного навчання. Глибоке навчання спирається на нейронні мережі - універсальну функцію, яка може вирішити будь-яку проблему, яку можна уявити на прикладах. Коли необхідно надати нейронній мережі безліч помічених прикладів певного типу даних, вона зможе виділити загальні закономірності між цими прикладами і перетворити їх в математичне рівняння, яке допоможе класифікувати майбутні фрагменти інформації.

Системи, в яких застосовується комп'ютерний зір:

- **Безпілотні автомобілі.** Комп'ютерне зір дозволяє безпілотним автомобілям розбиратися в навколишньому середовищі. Камери знімають відео під різними кутами навколо автомобіля і передають його в програмне забезпечення комп'ютерного зору, яке потім обробляє зображення в режимі реаль-

ного часу, щоб знаходити край доріг, зчитувати дорожні знаки, виявляти інші автомобілі, об'єкти і пішоходів.

- Системи для розпізнавання людей. Комп'ютерне зір також грає важливу роль в додатках для розпізнавання людей - технології, яка дозволяє комп'ютерам зіставляти зображення облич людей з їх зображеннями з бази даних.

- Системи доповненої реальності і змішаної реальності. Використовуючи комп'ютерне зір, пристрої доповненої реальності виявляють об'єкти в реальному світі, щоб визначати місця на дисплеї пристрою для розміщення віртуального об'єкта.

- Системи охорони здоров'я. Комп'ютерне зір також став важливою частиною досягнень в галузі охорони здоров'я. Алгоритми комп'ютерного зору можуть допомогти автоматизувати такі завдання, як виявлення злоякісних родимок на зображеннях шкіри або виявлення симптомів на рентгєнівських знімках і МРТ.

1.2 Типові задачі комп'ютерного зору

Задачі комп'ютерного зору полягають, головним чином, в отриманні корисної інформації з фото- або відеозображень. Найбільш вживаними завданнями можуть бути [2]:

- Калібрування камер і оптичних систем, як складаються з однієї камери, так і набору камер.
- Визначення руху по зображеннях.
- Сегментація зображень.
- Детектування об'єктів.
- Класифікація зображень.
- Розпізнавання облич.
- Оптичне розпізнавання символів.
- Генерація зображень.

- Визначення перешкод по ходу руху.
- Просторова реконструкції сцени.
- Аналіз відмінності в наборі зображень.

1.3 Розпізнавання образів

Розпізнавання - це здатність живих організмів виявляти в потоці інформації, що надходить від органів чуття, певні об'єкти, закономірності, явища. Воно може здійснюватися на основі зорової, слуховий, тактильної інформації. Розвиток і поширення комп'ютерної обробки інформації привели до виникнення в середині ХХ століття потреб в технологіях, що дозволяють машинам здійснювати розпізнавання в оброблюваній ними інформації. Розробка методів машинного розпізнавання дозволяє розширити коло виконуваних комп'ютерами завдань і зробити машинну переробку інформації більш інтелектуальною.

Люди часто плутають задачі класифікації зображень та виявлення конкретних об'єктів. Загалом, якщо необхідно віднести зображення до певної категорії, використовують класифікацію зображень [5]. З іншого боку, якщо необхідно ідентифікувати розташування об'єктів на зображенні і, наприклад, підраховувати кількість екземплярів об'єкта, можливо використовувати виявлення об'єкта.

Однак між цими двома сценаріями є деякий збіг. Якщо треба віднести зображення до певної категорії, може трапитися так, що об'єкт або характеристики, необхідні для виконання категоризації, занадто малі в порівнянні з повним зображенням. В цьому випадку більш високої продуктивності буде досягнуто з виявленням об'єкта замість класифікації зображень, навіть якщо не цікавить точне місце розташування або кількість об'єктів.

1.3.1 Класифікація

Виділяють наступні види класифікації об'єктів при їх розпізнаванні [3]:

1) Класифікація за формою.

Коли постає завдання виділити об'єкти за формою, повинні бути задані класифікаційні примітиви. У більшості методів пошуку в якості примітивів використовуються круглі, еліптичні, прямокутні або прямолінійні об'єкти. Також поширена практика пошуку об'єктів за формою, що має аналітичний опис.

2) Класифікація за положенням.

Одним з найбільш наповнених евристикою напрямків в теорії розпізнавання образів є методи пошуку по положенню. Зокрема, при пошуку осіб або інших фрагментів тіл в області кадру приймається допущення, що шукані області представляють собою довгасті, найчастіше витягнуті у вертикальному напрямку сукупності пікселів близьких за яскравістю. Тобто, якщо існує стійкий метод виділення в кадрі, наприклад, очей людини або носа, то можна по цих деталям зробити припущення, де знаходиться все інше. Винятки становлять атипові випадки, коли об'єкт в кадрі має нетривіальним поєднанням цих деталей в невідповідних для розпізнавання положеннях.

3) Класифікація за кольором.

Багато об'єктів можна класифікувати в залежності від їх кольору: вони або постійно мають певне забарвлення, або в деякі моменти їх забарвлення може бути регламентовано досить чітко. Більш того, в зв'язку з тим, що існує безліч базисів уявлення колірних компонент (RGB, YUV, YCrCb, HSV і т.д.), нерідкі випадки, коли в тому чи іншому базисі даний об'єкт можна класифікувати практично безпомилково.

1.4 Системи розпізнавання образів

Багато задач аналізу зображень можна успішно вирішити за допомогою класичної системи розпізнавання образів. Системи розпізнавання образів призначені для класифікації вхідних зображень або їх частин на кілька категорій. Схема класичної системи розпізнавання образів показана на рисунку 1.

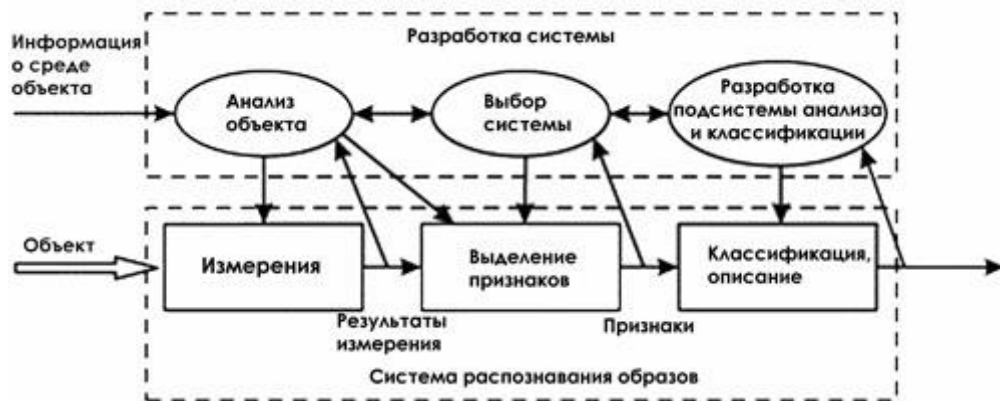


Рис. 1 Схема класичної системи розпізнавання образів

Кожен фрагмент вхідного зображення можна розглядати як точку в деякому просторі всіх можливих фрагментів. Для фрагмента розміром $N*N$ елементів, в якому кожне значення елемента квантоване на V рівнів, число різних можливих фрагментів V^{N*N} може виявитися дуже великим навіть для відносно невеликих фрагментів і при грубому квантуванні [4]. Тому зазвичай виникає необхідність скоротити це число за допомогою виділення ознак. Типовими ознаками є контурні точки, текстурні параметри і коефіцієнти Фур'є. Ці ознаки також можна розглядати як точки в деякому просторі. Для вироблення алгоритму розпізнавання образів (класифікації) вимірюються ознаки спеціально підібраних фрагментів зображень, точна класифікація яких відома [4].

1.5 Методи розпізнавання образів

Як правило, структура виявлення об'єктів складається з трьох кроків [5].

По-перше, модель або алгоритм використовуються для створення регіонів інтересів або пропозицій регіонів. Ці пропозиції області представляють собою великий набір обмежувальних рамок, що охоплюють всі зображення (тобто компонент локалізації об'єкта).

На другому етапі візуальні ознаки витягуються для кожної з обмежувальних рамок, вони оцінюються і визначається, чи присутні і які об'єкти в пропозиціях, на основі візуальних ознак (тобто компонента класифікації об'єктів).

На заключному етапі пост обробки рамки об'єднуються в одну обмежувальну рамку.

1.6 Розпізнавання обличчя людини

Розпізнавання обличчя — це автоматичний пошук обличчя людини на зображенні або відео і, при потребі, розпізнавання та співвідношення знайденого обличчя на основі наявних баз даних.

Існує декілька підходів для вирішення проблеми пошуку обличчя на зображенні.

Один з найперших методів розпізнавання обличчя — емпіричний підхід. Його ціль заключається в використанні деяких правил, які використовує людина для виділення обличчя. Наприклад, лоб зазвичай яскравіше, ніж центральна частина обличчя, яка, в свою чергу, однорідна по яскравості і кольору [6]. Цей методи легко реалізувати, але він практично непридатний, якщо на зображенні знаходиться велика кількість сторонніх об'єктів або декілька людей.

Наступний підхід використовує інваріантні ознаки, характерні для зображення особи. Метод виявляє характерні частини обличчя, його кордон, зміну форми, контрастності і т.д., об'єднує всі ці ознаки і верифікує. Даний метод може використовуватися навіть при повороті голови, але при наявності інших осіб або неоднорідному тлі розпізнавання стає неможливим.

Наступний алгоритм — це детектування осіб за допомогою шаблонів, які задає розробник. Особа бачиться певним шаблоном або стандартом, і мета алгоритму провести перевірку кожного сегмента на наявність цього шаб-

лону, причому перевірка може проводитися для різних ракурсів і масштабів. Така система вимагає безліч трудомістких обчислень [6].

Всі сучасні технології розпізнавання осіб використовують системи, які навчаються за допомогою тестових зображень. Для навчання використовуються бази з зображеннями, що містять особи, і не містять особи. Кожен фрагмент досліджуваного зображення характеризується як вектор ознак, за допомогою якого класифікатори (алгоритми для визначення об'єкта в кадрі) визначають, є дана частина зображенням особи чи ні.

1.7 Розпізнавання емоцій людини

Розпізнавання емоцій за виразом обличчя — об'єктивний метод оцінки реакції людини на деяке явище або іншу людину. Людина використовує безліч способів для вираження своїх емоцій: жести, мова тіла, інтонації. Але найбільш експресивним засобом служить, вираз обличчя: подивившись на зморщений ніс, насуплені брови або широку посмішку, можна багато чого сказати про людину й її емоційний стан.

Популярність наявності рішення задачі розпізнавання емоцій людини складається в тому, що отриманні дані можуть бути використані в багатьох галузях, наприклад аналіз стану водія, маркетингові дослідження, системи відеоаналітики для розумних міст, людино-машинне взаємодія і т. д.

Алгоритм роботи більшості методів, використовуваних в рамках вирішення завдання розпізнавання емоцій, складається з наступних етапів: збір і попередня обробка даних, вилучення та вибір ознак, класифікація та прийняття рішення [7].

При використанні деяких сучасних методів машинного навчання, таких як глибоке навчання, на вхід алгоритму подається саме зображення, а не набір ознак: вибір ознак і подальше визначення переважної на знімку емоції відбувається безпосередньо всередині нейронної мережі. В інших підходах витяг ознак і подальша класифікація є окремими, не пов'язаними між собою за-

вданнями, кожна з яких може бути вирішена одним з можливих способів. Так, для отримання ознак в подібних підходах можуть використовуватися такі методи як: Active Shape Model, Local Binary Pattern, вейвлети Габора [7]. Рішення завдання класифікації же може бути здійснено за допомогою методу опорних векторів.

1.7.1 Емоції людини

Емоція — це особливий вид психічних процесів, які висловлюють переживання людини до навколишнього світу і до самої себе. Згідно з однією з теорій, автором якої є російський фізіолог П.К. Анохін, здатність відчувати емоції була вироблена в процесі еволюції як засіб більш успішної адаптації живих істот до умов існування. Емоція виявилася корисною для виживання і дозволила живим істотам швидко і найбільш економно реагувати на зовнішні впливи.

Емоції відіграють величезну роль в житті людини і міжособистісному спілкуванні. Вони можуть бути виражені різними способами: мімікою, позою, руховими реакціями, голосом і вегетативними реакціями (частота серцевих скорочень, артеріальний тиск, частота дихання). Однак найбільшою виразністю володіє обличчя людини.

Кожна людина висловлює емоції дещо по-різному. Відомий американський психолог Пол Екман, досліджуючи невербальну поведінку ізольованих племен в Папуа-Новій Гвінеї в 70-х роках минулого століття, встановив, що ряд емоцій, а саме: гнів, страх, сум, відраза, презирство, подив і радість є універсальними і можуть бути зрозумілі людиною, незалежно від його культури [8].

Але правильно інтерпретувати емоцію людини не завжди просто. Як показали дослідження нейробіолога Лізи Фельдман Барретт, при розпізнаванні емоцій людина активно використовує контекстну інформацію: голос, дії, ситуацію. Використовуючи тільки область обличчя, правильне передбачення зробити буде складно. У зв'язку з цим для вирішення цього завдання

необхідно використовувати як додаткові модальності, так і інформацію про зміну сигналів з плином часу.

1.7.2 Розпізнавання емоцій з зображення

Сучасні алгоритми машинного навчання демонструють точність в 55% для розпізнавання виразу обличчя на зображеннях.

Основний алгоритм автоматичного розпізнавання емоцій складається з наступних етапів [9]:

1. Отримання зображення з обличчям людини.
2. Обробка отриманого зображення та виділення людини на зображенні.
3. Розпізнавання елементів обличчя людини.
4. Виділення ключових точок на обличчі.
5. Класифікація емоцій.

Для виділення обличчя на зображенні найбільш ефективним є метод Віоли-Джонса, який забезпечує досить хорошу швидкість і високу точність виявлення заданих об'єктів на зображеннях [6]. Після того як обличчя було виділено, необхідно виділити його елементи. Людина проявляє емоції за допомогою брів, очей і рота. Для виділення цих елементів на зображенні обличчя використовується все той же метод Віоли-Джонса. В якості вхідного зображення подається зображення особи, виділене на попередньому етапі [9]. Наступним кроком є знаходження ключових точок виділених елементів обличчя. Визначити емоції можна на основі аналізу декількох ключових точок. Після виділення ключових точок можна виконати класифікацію емоції [9].

Однак емоції людини мають не статичний, а динамічний характер, вони досить швидко змінюються, і сама швидкість їх зміни теж може служити показником емоційного стану. Окрім цього в сучасних системах використовуються технології глибокого навчання для розпізнавання емоцій людини. Наприклад, згорткові нейронні мережі, рекурентні нейронні мережі.

1.7.3 Розпізнавання емоцій у відео-поточі

Точність розпізнавання емоцій людини у відео-поточі, тобто в режимі реального часу, трохи менша, ніж точність розпізнавання емоцій на зображенні, вона складає 46%.

Модель розпізнавання емоцій на відео схожа на розпізнавання емоцій на зображеннях, але має свої особливості. По-перше, вона починається з отримання необхідної інформації з відео: спочатку отримуються кадри, а потім розпізнаються обличчя і виконується корегування для кожного окремого кадру. Крім того, пропонується об'єднувати кадри з плином часу, оскільки тимчасова коваріація захоплює необхідну динаміку руху обличчя.

Отримавши кадр з відео-поточі, може використовуватись той самий алгоритм аналізу емоцій, що використовувався при розпізнаванні емоцій з зображення, бо окремий кадр вважається окремим зображенням. Тобто будуть виконані наступні етапи: 1) реєстрація зображення; 2) первинна обробка зображення; 3) виділення обличчя на зображенні; 4) виділення ключових точок на обличчі; 5) класифікація емоцій [9].

1.7.4 Основні проблеми, пов'язані з розпізнаванням облич та емоцій

Проблеми, пов'язані з розпізнаванням і ідентифікацією осіб по відеофіксації в режимі реального часу, можна класифікувати на наступні [11]:

1. Проблеми, пов'язані з навчанням.

Навчання систем розпізнавання та ідентифікації осіб по відеофіксації в режимі реального часу реалізується за допомогою поділу і класифікації окремих об'єктів із зазначенням їх приналежності того чи іншого способу. В результаті навчання розпізнає система набуває здатності реагувати відповідними реакціями на об'єкти одного або різних образів.

2. Проблеми, пов'язані з розпізнаванням.

За навчанням йде процес розпізнавання нових об'єктів, який характеризує дії вже навченої системи. Автоматизація цих процедур і становить про-

блему навчання розпізнавання образів. У тому випадку, коли людина сама розгадує або придумує, а потім нав'язує системі правило класифікації, проблема розпізнавання вирішується частково, так як основну і головну частину проблеми (навчання) людина бере на себе.

Для того, щоб система ефективно розпізнавала і ідентифікувала людей по відеофіксації в режимі реального часу необхідно забезпечити максимальну інваріантність до різних зображень осіб в умовах впливу зовнішніх факторів [11]. Зовнішніми факторами можуть виступати: освітлення, положення голови у просторі, фон, масштаб зображення обличчя, часткове закриття обличчя (окуляри, волосся, вуса, борода).

Рекомендації щодо вдосконалення існуючих алгоритмів розпізнавання в умовах незмінності архітектури системи відеофіксації і обчислювальних потужностей:

- Звуження вибірки розпізнавання за допомогою аналізу рудиментарних особливостей схематичних зображень облич.
- Оптимізація шаблонних зображень в частині освітленості.
- Оптимізація шаблонних зображень в частині підвищення динамічних властивостей осіб.
- Підвищення значущості властивостей пігментації шкіри.
- Підвищення значущості кольору шкіри або волосся.

1.8 Аналіз існуючих систем розпізнавання емоцій

1.8.1 FaceReader

Цей застосунок може вірно розпізнавати по виразу обличчя людини такі емоції, як «щастя», «смуток», «злість», «здивування», «страх», «відраза» і «спокій». Крім того, FaceReader здатний визначити вік, стать і расу людини. Додаток не має потреб в навчанні і додатковому налаштуванні.

У програмі реалізовані технології комп'ютерного зору. Зокрема, це метод Active Template, що полягає в накладенні деформованого шаблону на зо-

браження обличчя. Інтерфейс застосунку FaceReader продемонстровано на рисунку 2.

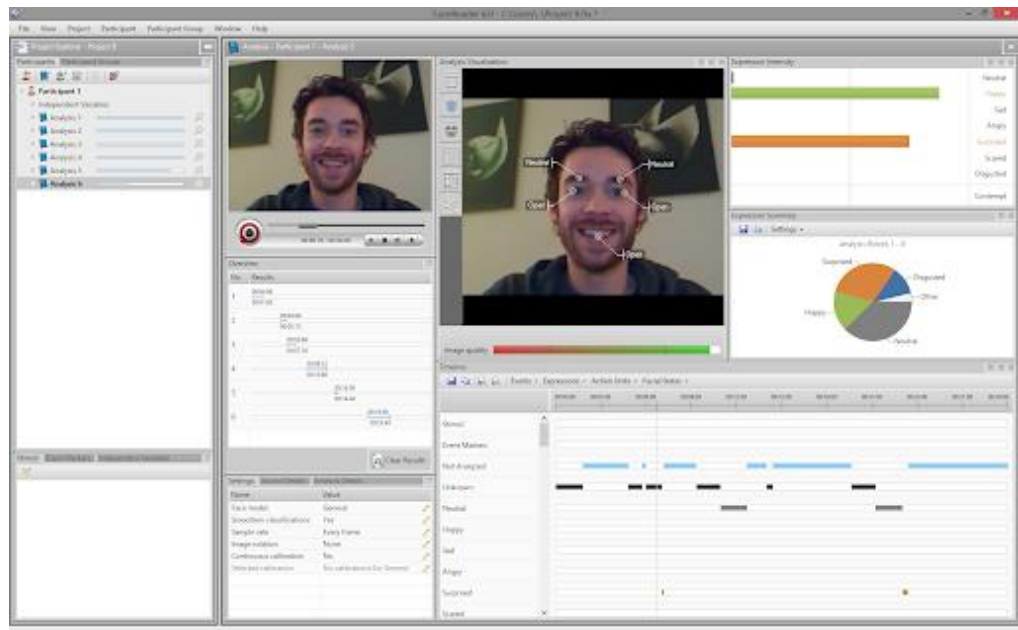


Рис. 2 Інтерфейс програми “FaceReader”

Переваги даної системи:

- Середній відсоток розпізнавання емоцій дорівнює 89% і не залежить від нахилу й повороту площині обличчя.
- Програма працює з великою кількістю форматів відеофайлів з кодеками MPEG1, MPEG2, DivX4, DivX5, DivX6, DV-AVI.
- Додаток не обмежується роботою з відео, які необхідно завантажити, FaceReader працює і зі статичними зображеннями і в реальному часі, зчитуючи дані з камери користувача.
- Можливості візуалізації: наявність гістограм, діаграм, процентне відображення аналізованих емоцій.

Недоліки системи:

- Відсоток розпізнавання емоцій і віку сильно падає при роботі з обличчями дітей віком менше 5 років.
- Визначення може відбуватися неточно, якщо людина носить окуляри або має бороду.
- Нейтральний стан людини не може бути визначений.

1.8.2 EmoDetect

Програмне забезпечення EmoDetect дозволяє визначити психоемоційний стан людини по виборці зображень (відео або набір файлів). Розроблений класифікатор емоцій дозволяє визначити 6 базових емоцій: радість, здивування, смуток, злість, страх, презирство. Аналіз емоцій відбувається за допомогою нейронних мереж: виконується знаходження опорних точок на обличчі людини, і класифікатор будує загальну картину на основі їх становища щодо одне одного.

Розрахунок рухових одиниць і їх класифікація відбувається за системою кодування лицьових рухів П. Екмана (FACS Action Units). Інтерфейс програми EmoDetect проілюстровано на рисунку 3.

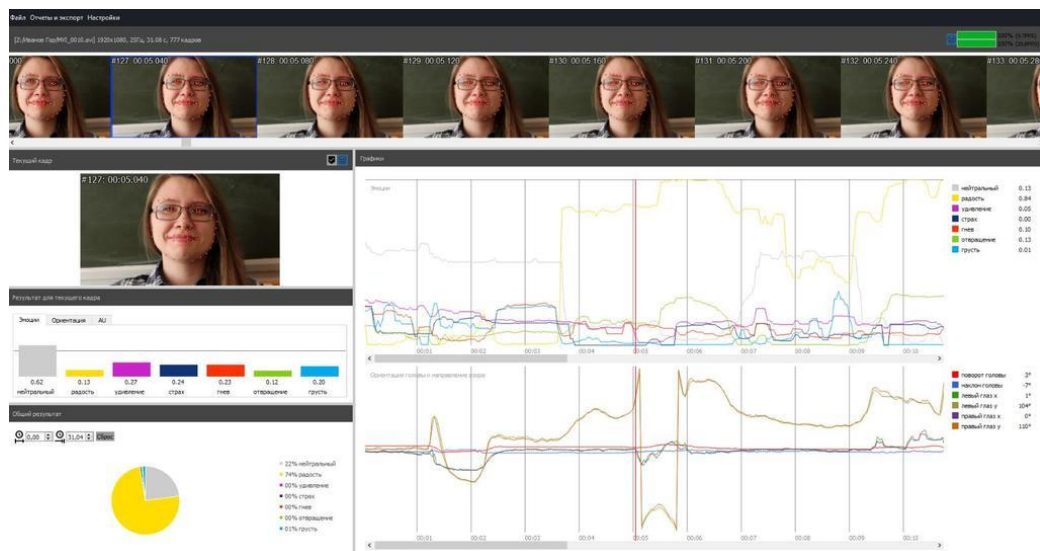


Рис. 3 Інтерфейс програми “EmoDetect”

Переваги системи:

- Можливість визначення нейтрального виразу обличчя користувача, адаптивність алгоритму до індивідуальних особливостей кожної людини.
- Побудова графіка динамічної зміни інтенсивності емоцій випробуваного в часі і формування звіту про результати обробки відео.

- Можливість запису відео-файлів з веб-камери, аналіз всіх підтримуваних операційною системою форматів.

- Режим вкладок для проведення порівняльної оцінки.

- Ідентифікація по обличчю в режимі реального часу.

Недолікі системи:

- Обличчя, що повернуті в профіль, не можуть бути проаналізовані.

- Складність аналізу емоцій людей різної раси.

1.8.3 FindFace Security

Цей програмний застосунок сканує потік вхідних даних і, при виявленні в кадрі осіб, порівнює їх з заведеною базою даних на предмет наявності збігів. У разі знаходження збігів система відправляє повідомлення операторам в реальному часі. Цей додаток широко використовується в системі безпеки європейських країн для виявлення підозрілих осіб в громадських місцях. Великі компанії ж використовують його для ідентифікації важливих клієнтів. Рядові фірми використовують дану систему для розпізнавання облич своїх співробітників і виявлення незареєстрованих відвідувачів, використовуючи бази даних з фотографіями своїх підлеглих. Також дана розробка допомагає аналізувати стан людини й повідомляє в разі отримання підозрілого результату. Інтерфейс системи FindFace Security можна побачити на рисунку 4.

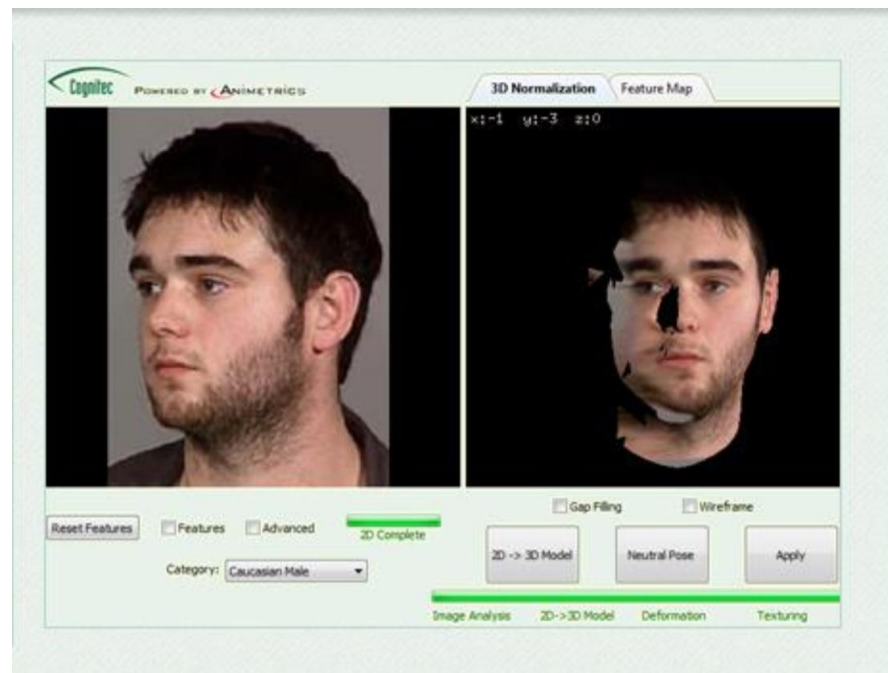


Рис. 4 Інтерфейс програми “FaceSecurity”

Система використовує такі етапи розпізнавання облич:

- Детектування особи і силуету на зображенні.
- виправлення візуальних спотворень.
- витяг характеристик особи.
- Верифікація або ідентифікація особи.

Переваги програми:

- Робота в реальному часі, можливість стеження за особою на декількох потоках даних одночасно.
- У режимі реального часу особа порівнюється з еталонами, занесеними в базу даних.
- Реєстрації при нерухомому зображенні або при живому потоці відео в ручному і автоматичному режимі.
- Застосування C++ API і Web Services API.
- Можливість масштабованості в межах «контрольного списку»: розмір, кількість відеопотоків і число видимих осіб на зображенні.

Недоліки програми:

- Неможливість проведення обчислень для осіб, що мають відхилення від фронтального положення більш ніж 15 градусів.
- Світлочутливість. Неможливість розпізнавання особи, при попаданні на нього тіней і відблисків

1.9 Результати аналізу існуючих рішень

Результат аналізу існуючих рішень розпізнавання облич і емоцій людини показав, що проблема ідентифікації емоцій людини може бути вирішена, з використанням сучасних методів розпізнавання на основі згорткових мереж. Система може аналізувати емоції не тільки з зображення людини, а й досліджувати емоційний стан людини з відео-потoku.

Також аналіз створених систем показав, що не всі емоції можуть бути ідентифіковані точно, але процент правильного аналізу достатньо великий. Деякі системи витрачають дуже багато часу на аналіз емоцій, тому не завжди відбувається розпізнавання емоцій в режимі реального часу. Ще одним недоліком таких систем є те, що особливості обличчя людини не завжди враховуються і в зв'язку з цим, не всі обличчя можуть бути проаналізовані. Тому було прийнято рішення сфокусуватися на вирішенні проблеми розпізнавання емоцій людини в режимі реального часу з врахуванням особливостей обличчя людини (окуляри, борода, веснянки, чуб) за допомогою використання згорткової нейронної мережі.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЗАСОБІВ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ

2.1 Теорія розпізнавання обличчя людини

Люди розпізнають людей і їх обличчя автоматично кожен день і практично без зусиль.

Незважаючи на те, що для людей це звучить як дуже проста задача, вона виявилася набагато складнішою для комп'ютера, так як в ній є багато змінних, які можуть погіршити точність методів, наприклад: варіація освітлення, низька роздільна здатність, оклюзія та багато іншого.

У комп'ютерній науці розпізнавання обличчя — це, по суті, завдання розпізнавання людини по його зображенню. Воно стало дуже популярним в останні два десятиліття, в основному завдяки новим розробленим методам і високій якості сучасних відео / відеокамер.

Необхідно відзначити, що розпізнавання обличчя відрізняється від детектування обличчя [6]:

Детектування обличчя: мета цього процесу — знайти обличчя людини (місце розташування та розмір) на зображенні та, швидше за все, отримати їх для використання алгоритмом розпізнавання обличчя.

Розпізнавання обличчя: оскільки зображення обличчя вже отримано, обрізані, змінені в розмірах і зазвичай перетворені в градації сірого, алгоритм розпізнавання обличчя відповідає за пошук характеристик, які найкраще описують зображення.

Процес виявлення та розпізнавання обличчя складається з наступних етапів (Рис. 5):

1. Детектування обличчя.

Для розпізнавання обличчя на зображенні або відео з використанням опорних точок на обличчі, таких як очі, ніс, рот і т.д.

2. Отримання ознак.

Вирівнювання, нормалізація обличчя для кращої точності розпізнавання.

3. Розпізнавання обличчя.

Для розпізнавання конкретної людини на зображенні або відео по базі даних.

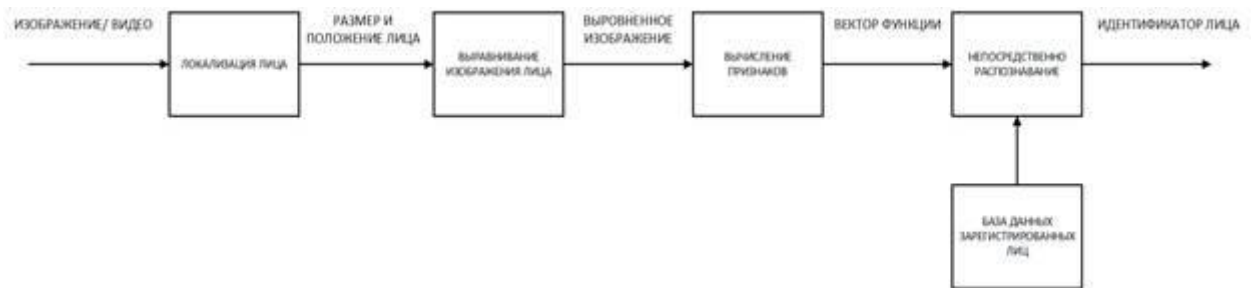


Рис. 5 Загальний алгоритм розпізнавання обличчя

Коли система виявляє обличчя, вона визначає стан, розмір і позу голови. Використовуючи глибину і вісь вимірювання, на які не впливає освітлення, тривимірне розпізнавання особи може навіть використовуватися в темряві і має можливість розпізнавати об'єкт під різними кутами зору з можливістю розпізнавання до 90 градусів (особа в профілі). У разі роботи з 2D, голова повинна бути повернута не менше ніж на 35 градусів до камери. На етапі безпосереднього розпізнавання система порівнює обчислені ознаки із закладеними в базу даних еталонами. Тому в якості основного відмінності серед алгоритмів буде обчислення ознак і порівняння їх сукупностей між собою

Системи розпізнавання обличчя можуть працювати в основному в двох режимах:

- Перевірка або автентифікація зображення обличчя: в основному виконується порівняння вхідного зображення особи із зображенням обличчя, що пов'язане з відповідним користувачем, якому потрібно автентифікація. В основному це порівняння 1x1.
- Ідентифікація або розпізнавання обличчя: в основному виконується порівняння вхідного зображення обличчя з усіма зображеннями обличчя з

набору даних з метою знайти користувача, який відповідає цій особі. В основному це порівняння $1 \times N$.

2.2 Методи розпізнавання облич

Методи розпізнавання облич можна класифікувати по таким категоріям [6]:

1. Методи, засновані на зовнішньому вигляді.
2. Методи зіставлення на основі ознак.
3. Гібридні методи.

Класифікація методів розпізнавання облич і алгоритмів, що використовуються в описаних вище методах, показано на рисунку 6.

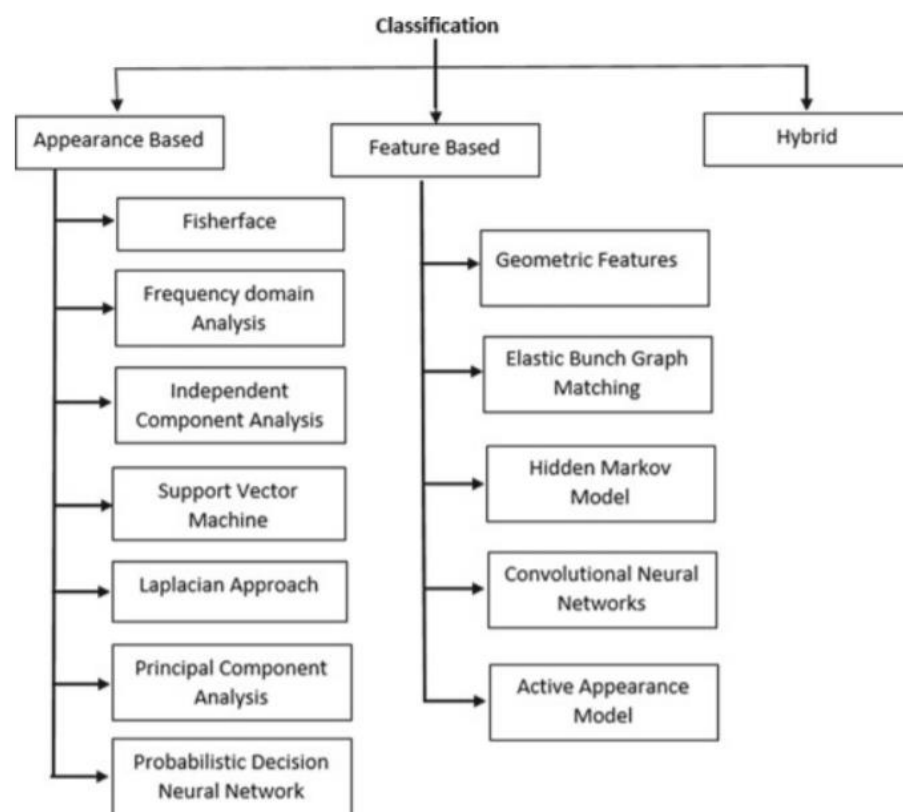


Рис. 6 Класифікація методів розпізнавання обличчя

2.2.1 Метод Віоли-Джонса

У 2001 році Пол Віола і Майкл Джонс запропонували алгоритм для розпізнавання обличчя, який став проривом в області розпізнавання облич

людей. Метод використовує технологію скануючого вікна. Тобто рамка, розміром, меншим, ніж вихідне зображення, рухається з деяким кроком по зображенню, і за допомогою каскаду слабких класифікаторів визначає, чи є в даному вікні обличчя. Метод скануючого вікна ефективно використовується в різних завданнях комп'ютерного зору і розпізнавання об'єктів [10].

Метод складається з 2-х під алгоритмів: алгоритм навчання і алгоритм розпізнавання. На практиці швидкість роботи алгоритму навчання не важлива. Вкрай важлива швидкість роботи алгоритму розпізнавання.

Метод має такі переваги:

- можливе виявлення більше однієї особи на зображенні;
- використання простих класифікаторів показує хорошу швидкість і дозволяє використовувати цей метод у відео-потоці.

Спочатку алгоритм був запропонований для розпізнавання тільки осіб, але його можна використовувати для розпізнавання інших об'єктів. Одним з вкладів Віюлі і Джонса було застосування таблиці сум, яку вони назвали інтегральним зображенням.

В якості ознак для алгоритму розпізнавання авторами були запропоновані ознаки Хаара, на основі вейвлетів Хаара [12]. Вони були запропонований угорським математиком Альфредом Хааром в 1909 році.

У задачі розпізнавання осіб загальне спостереження, що серед всіх обличчя області очей темніше області щоби. Розглянемо маски, що складаються зі світлих і темних областей (Рис. 7).



Рис. 7 Приклад масок в алгоритмі Віюлі-Джонса

Кожна маска характеризується розміром світлої і темної областей, пропорціями, а також мінімальним розміром. Спільно з іншими спостереження-

ми були запропоновані наступні ознаки Хаара, як простір ознак в задачі розпізнавання для класу осіб (Рис. 8).

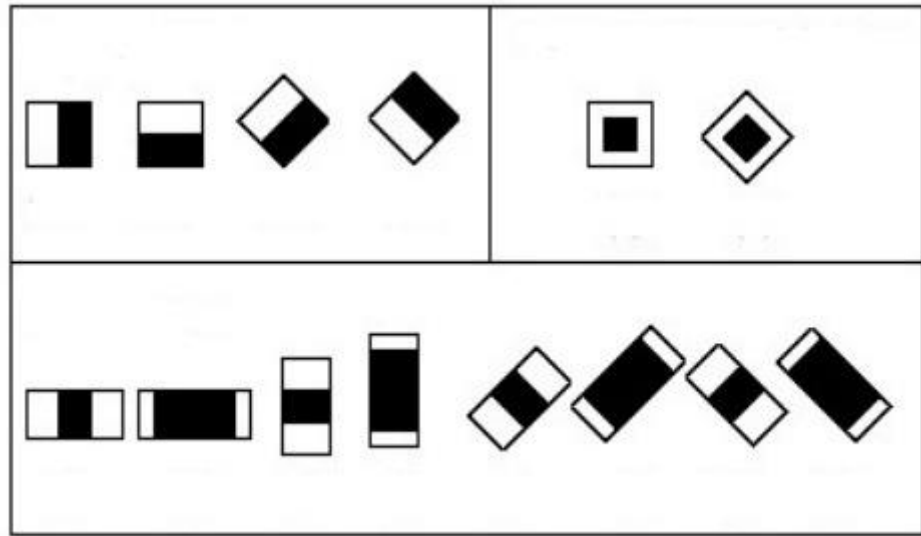


Рис. 8 Набір ознак Хаара з методу Віоли-Джонса

Ознаки Хаара дають точкове значення перепаду яскравості по осі X і Y відповідно. Тому загальна ознака Хаара для розпізнавання облич представляє набір двох суміжних прямокутників, які лежать вище очей і на щоках. Значення ознаки обчислюється за формулою: $F = X - Y$, в якій X - сума значень яскравості точок, що закриті світлою частиною ознаки, а Y - сума значень яскравості точок, що закриті темної частиною ознаки [12].

Видно, що якщо вважати суми значень інтенсивностей для кожної ознаки це зажадає значних обчислювальних ресурсів. Віолою і Джонсом було запропоновано використовувати інтегральне представлення зображення. Інтегральне уявлення дозволяє швидко розраховувати сумарну яскравість довільного прямокутника на зображенні й час розрахування не залежить від площі прямокутника. Інтегральне уявлення – це матриця, розміри якої є точно такими, як і розміри вихідного зображення. У кожному елементі такої матриці зберігається сума інтенсивностей всіх пікселів, які знаходяться лівіше и вище цього елемента.

Пошук обличчя відбувається за допомогою скануючого вікна, в оригінальному алгоритмі розміри якого становлять 24x24 пікселі. Вікно переміщується по зображенню з кроком в 1 піксель і для кожного його положення обчислюються ознаки Хаара з різним масштабом і положенням у вікні. Також сканування проводиться і для різних масштабів скануючого вікна. Класифікатором передаються знайдені ознаки і визначається за їх значенням чи є область зображення, що дорівнює вікну, обличчям чи ні [10].

2.2.2 Метод гнучкого порівняння на графах

Метод гнучкого порівняння на графах (Elastic Graph Matching) – це алгоритм комп'ютерного зору для розпізнавання об'єктів або класів об'єктів на зображенні на основі графічного представлення, витягнутого з інших зображень. Він широко використовується при розпізнаванні осіб і аналізі, а також для жестів і інших класів об'єктів.

Візуальні об'єкти в EGM представлені у вигляді позначених графів, де вузли представляють локальні текстури на основі вейвлетів Габора, а краї є відстані між місцями розташування вузлів на зображенні. Таким чином, зображення об'єкта представляється як набір локальних текстур в певному просторовому розташуванні. Якщо новий об'єкт на зображенні повинен бути розпізнаний, помічені графи збережених об'єктів, так звані модельні графи, зіставляються з зображенням. Для кожного графа моделі розташування вузлів в зображенні оптимізовані таким чином, щоб локальна текстура зображення відповідала локальній текстурі моделі, а відстані між місцями розташування відповідали відстаням між вузлами моделі. Граф моделі з найкращим відповідністю є розпізнаний об'єкт, і з його розташуванням вузлів на зображенні можна створити граф зображення.

У подібних системах розпізнавання графи можуть являти собою як прямокутну решітку (Рис. 9), так і структуру, утворену характерними (антропометричними) точками особи (Рис. 10).

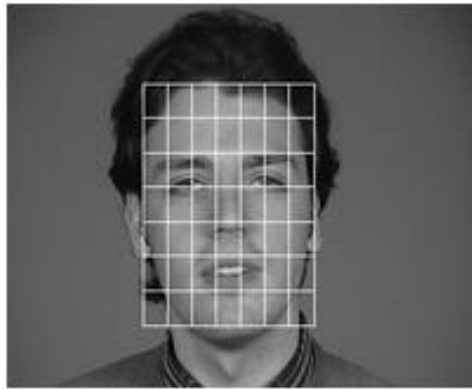


Рис. 9 Приклад структури графа для розпізнавання обличчя – регулярна решітка

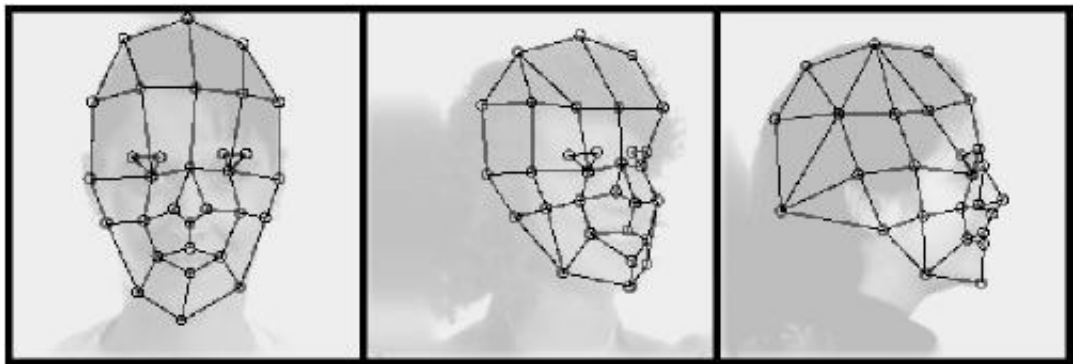


Рис. 10 Приклад структури графа для розпізнавання обличчя - граф на основі антропометричних точок обличчя.

Припустимо, дано 1000 зображень людей ідентичною пози. Це являє собою модельну галерею. Для розпізнавання обличчя використовуються наступні дії:

1. Побудова графа обличчя. Першим кроком до завантаження системи є визначення структури графа для даної пози. Таким чином, використовується перше зображення і вручну визначається розташування вузлів на обличчі, які легко локалізувати, наприклад, куточки очей або рота, центр очей, кінчик носу. Також визначаються ребра між вузлами. Це становить перший граф обличчя.

2. Побудова графа групи обличч. Певний вище однолінійний граф можна розглядати як груповий граф з одним екземпляром в ньому. Його можна порівняти з другим зображенням обличчя, але якщо перші два зобра-

ження обличчя не дуже схожі, збіг буде поганої якості. Груповий граф з двома екземплярами потім зіставляється з третім зображенням, і після деякої ручної корекції отримується третій примірник для зв'язкового графа. Повторюючи цей процес, груповий граф зростає, і в міру його зростання відповідність новим зображенням стає все більш надійною.

3. Створення модельної галереї графів. Оскільки отримано груповий граф, який забезпечує достатню якість для пошуку місць розташування вузлів на новому обличчі, можливо обробити зображення, що залишились, повністю автоматично.

4. Побудова тестового графа. Припустимо, отримано нове зображення і необхідно знайти зображене обличчя в галереї. Спочатку потрібно створити граф для пробного зображення. Цей процес працює точно так само, як і для модельних зображень, тільки використовується пробне зображення.

5. Порівняння з усіма модельними графами. Граф зображення порівнюється з усіма модельними графами, в результаті отримується значення подібності. Вони складають основу рішення про розпізнавання.

6. Розпізнавання. Для розпізнавання очевидно, що граф моделі з найбільшою подібністю з графом зображення є кандидатом на розпізнавання. Однак, якщо оптимальне значення подібності відносно низьке, система може вирішити, що людину на пробному зображенні взагалі немає в галереї моделі; і якщо є кілька дуже високих значень подібності, система може вирішити, що людина, швидше за все, присутня в галереї моделей, але є кілька можливих кандидатів.

На рисунку 11 поетапно показано деформацію графа у вигляді регулярної решітки.

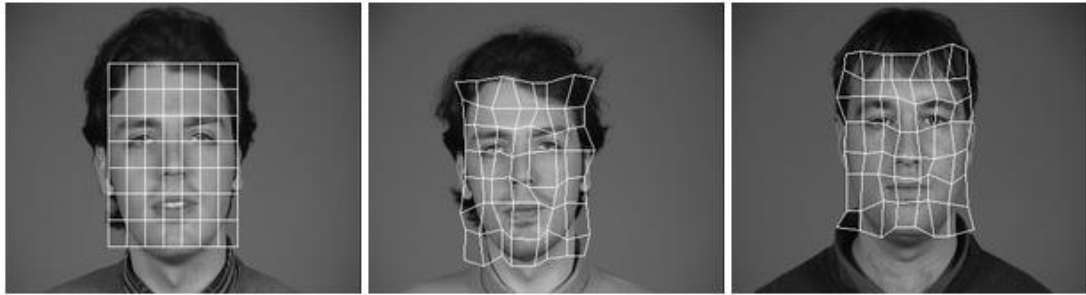


Рис. 11 Приклад деформації графа у вигляді регулярної решітки

2.2.3 Приховані Марковські моделі

Приховані Марковські моделі використовуються для опису зображень обличчя на основі тільки статистичних даних. Вони дозволяють враховувати просторово-часові характеристики сигналу, тому широко використовуються для розпізнавання мови.

У розпізнаванні обличчя приховані Марковські моделі застосовуються наступним чином. Кожному класу зображень обличчя відповідає своя модель. Зображення, що розпізнається, представляється у вигляді послідовності подій з імовірностями переходу a_{mn} . Для кожної моделі обчислюється ймовірність генерації представленої послідовності. Модель, що отримала максимальну ймовірність генерації, представляє клас розпізнаваного обличчя.

Метод прихованих Марківських моделей добре працює при варіюванні умов освітлення, зміні виразу обличчя або його орієнтації. Недоліком є низька розрізняюча здатність класів, що може обмежити використання при великих кількостях користувачів.

2.2.4 Метод головних компонент

Один з найпростіших і ефективних підходів методу головних компонент (Principal Component Analysis, PCA) в системах розпізнавання обличчя використовується так званий підхід власного обличчя (eigenfaces). Цей підхід перетворює обличчя в невеликий набір суттєвих характеристик, власних обличчя, які є основними компонентами початкового набору навчальних зображень (навчального набору).

Розпізнавання виконується шляхом проектування нового зображення в підпростір власного обличчя, після чого людина класифікується шляхом порівняння його положення в просторі власних облич з положенням відомих облич [17]. Перевага цього підходу перед іншими системами розпізнавання облич полягає в його простоті, швидкості і нечутливості до невеликих або поступових змін на обличчі. Проблема обмежена файлами, які можна використовувати для розпізнавання особи. А саме зображення повинні бути вертикальними фронтальними видами людських облич.

Для проведення класифікації за допомогою методу будь-яке зображення представляється як вектор в n -вимірному просторі, де n - кількість пікселів зображення. Теоретичне підґрунтя методу полягає в перекладі вихідного зображення в простір ознак з набагато меншою розмірністю, в якому його можна максимально повно охарактеризувати.

Застосування методу до набору векторів лінійного простору дозволяє перейти до такого базису, що основна дисперсія буде спрямована вздовж кількох перших осей базису, званих головними осями [17]. Це дозволяє представити унікальність векторів тренувального набору за допомогою декількох компонент (Рис. 12). Отриманий простір максимально інформативно описує об'єкт.

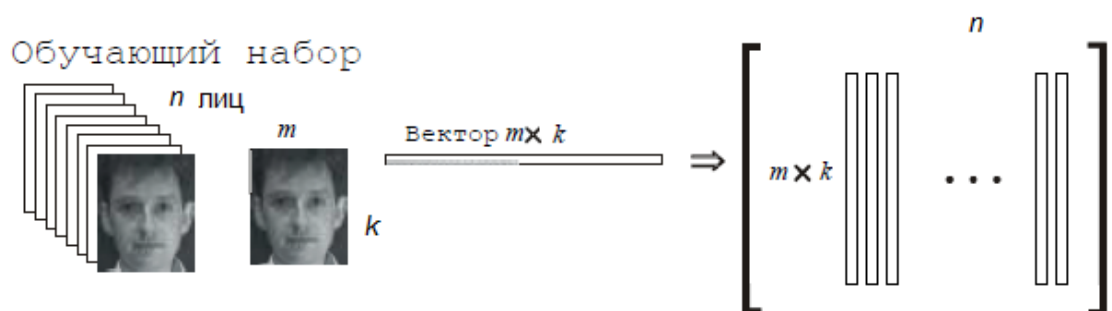


Рис. 12 Перетворення навчального набору облич в матрицю X

При розпізнаванні обличчя за допомогою методу головних компонент обчислюються дві характеристики: відстань від проекції тестового вектора до середнього вектора тренувального набору і відстань від тестового вектора до його проекції в підпростір головних компонент. Виходячи з цих параметрів,

вноситься рішення про приналежність тестового об'єкта класу, що отриманий завдяки тренувальному набору. До недоліків методу слід віднести високу критичність до варіацій масштабу обличчя (40% помилок), просторової орієнтації (15% помилок) і освітлення (5% помилок) [17]. Метод головних компонент можна розглядати як окремий випадок факторного аналізу, коли всі специфічні фактори прийняті рівними нулю, а загальні фактори ортогональні.

2.2.5 Активні моделі зовнішнього вигляду

Активні моделі зовнішнього вигляду (Active Appearance Models, ААМ) — це алгоритм комп'ютерного зору для зіставлення статистичної моделі форми об'єкта і зовнішнього вигляду об'єкта нового зображення. Вони будуються на етапі навчання. Набір зображень разом з координатами орієнтирів, які з'являються на всіх зображеннях, надається керівнику навчання.

Модель була вперше представлена Тімом Кутсом і Крісом Тейлором в контексті аналізу осіб на 3-й Міжнародній конференції з розпізнавання осіб і жестів в 1998 році [19]. Далі автори описали цей підхід як загальний метод комп'ютерного зору на Європейській конференції з комп'ютерного зору в тому ж році [19]. Цей підхід широко використовується для зіставлення і відстеження облич, а також для інтерпретації медичних зображень.

Алгоритм використовує різницю між поточною оцінкою зовнішнього вигляду і цільовим зображенням для управління процесом оптимізації. Використовуючи переваги методу найменших квадратів, він може дуже швидко відповідати новим зображенням. Розмітка зображень виробляється вручну. Кожна мітка має свій номер і визначає характерну точку, яку повинна буде знаходити модель під час адаптації до нового зображення (Рис. 13).

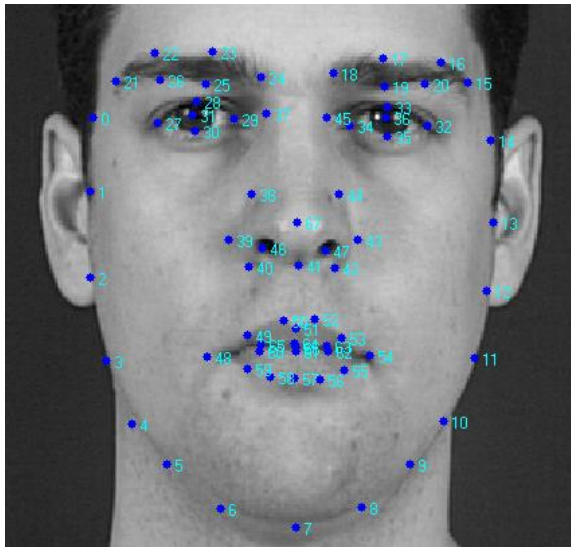


Рис. 13 Приклад розмітки зображення обличчя

Активні моделі зовнішнього вигляду використовуються для моделювання осіб, схильних до як жорсткої (поворот, масштабування), так і нежорсткої (зміна виразу обличчя) деформації. При розпізнаванні обличчя моделі бази піддаються деформації для знаходження активної моделі, ступінь близькості якої з розпізнаваним обличчям була б максимальна (Рис. 14).

Одночасно з цим змінюється текстура активної моделі. Найбільш передовим прикладом методу активних моделей зовнішнього вигляду є розпізнавання на основі підгонки деформованих 3D моделей.



Рис. 14 Розпізнавання обличчя на основі підгонки

В ході роботи методу на оригінальному зображенні виділяються стандартні елементи обличчя, зображенню обличчя ставиться у відповідність 3D модель середнього обличчя з усіх обличч бази і відбувається ряд трансформацій, щоб підігнати модель до оригіналу.

Спочатку змінюється орієнтація і розмір середньої моделі, потім її висвітлення. Автоматизований алгоритм, шляхом зміни декількох параметрів для підгонки форми моделі, знаходить найбільш схожу форму. Обчислюючи відстань між обличчями з бази і підігнаною 3D моделлю, можна знайти модель найбільш відповідного обличчя бази.

При порівнянні моделей використовується метод головних компонент, при цьому сама форма обличчя і його текстура порівнюються окремо і кореляції між ними не враховуються.

Підгонка деформованих 3D моделей показує багатообіцяючі результати і дозволяє отримати інформацію не тільки про приналежність особи, а й про стать його власника, емоційне вираження.

2.3 Методи розпізнавання емоцій людини

Аналіз розпізнавання лицьової експресії є складним і комплексним завданням, тому існує величезна кількість методів і алгоритмів, розглянутих в роботах різних авторів. Деякі методи дуже спірні і мають суттєві недоліки, так як автори намагаються знайти все більш витончений метод, який доведе свою спроможність і виявить недоліки попередніх алгоритмів. Є деякі особливості обличчя людини, через які не існує на даний момент еталонного способу вирішення завдання. Дуже часто багато методів не враховують аксесуарів і лицьової рослинності, типу бороди, вусів, нерівномірних брів і об'ємних зачісок, що закривають або заважають розпізнаванню. Плюс до того, емоція дуже абстрактний процес, який не є однаковим для кожної людини. Навіть люди часто стикаються з проблемою нерозуміння деякої експресії, наприклад, відкритий рот не завжди означає подив. Однак, основний недолік систем розпізнавання емоцій заключається в тому, що еталонний алгоритм аналізу емоцій ще досі не реалізований.

2.3.1 Метод ключових точок

Алгоритм розділений на два етапи: отримання координат ключових точок і класифікація емоцій на їх основі.

Вхідними даними для методу є набір зображень облич, в якому кожному зображенню відповідає файл розмітки, що містить координати ключових точок, обраних людиною. За цими даними будується дві статистичні моделі: модель форми та модель текстури [21].

Модель форми — параметрична лінійна модель, що описує можливі варіанти положення ключових точок. Формою при цьому називається вектор координат ключових точок:

$$s = (x_1, y_1, x_2, y_2, \dots, x_n, y_n,)^T.$$

Модель текстури — подібна модель, але описує вже можливі варіації інтенсивності пікселів. Відповідно текстурою називається вектор всіх пікселів всередині зовнішнього контуру форми:

$$s = (t, t_2, \dots, t_n,)^T.$$

Оскільки на різних зображеннях кількість пікселів всередині зовнішнього контуру форми може бути різним, перед створенням моделі текстури всі зображення облич приводяться до єдиної усередненої форми за допомогою кусочно-афінного перетворення: безліч точок форми триангулюється, а потім кожен симплекс звичайним афінним перетворенням транслюється в нові координати.

Визначити емоції можна на основі аналізу декількох ключових точок [22]. Наприклад, за допомогою комбінацій точок брів і рота (Рис. 15).

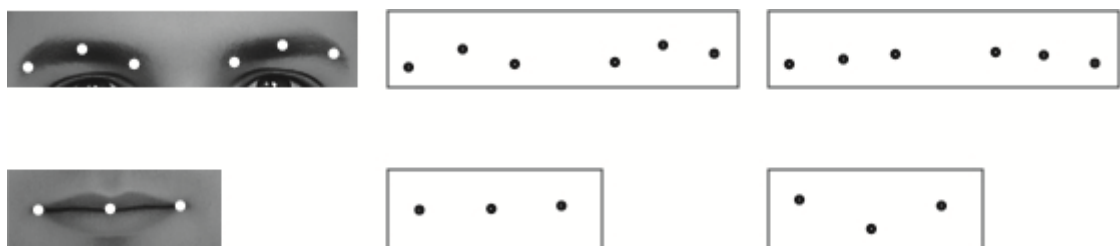


Рис. 15 Комбінація ключових точок брів і рота, відповідна їх зображенням

Виділення ключових точок здійснюється наступним чином, що проілюстровано на рисунку 16:

1. Перетворення кольорового зображення до напівтонового виду;
2. Перетворення з напівтонової форми до бінарного вигляду;
3. Застосування до бінарного зображення градієнтної маски;
4. Локалізація ключових точок.

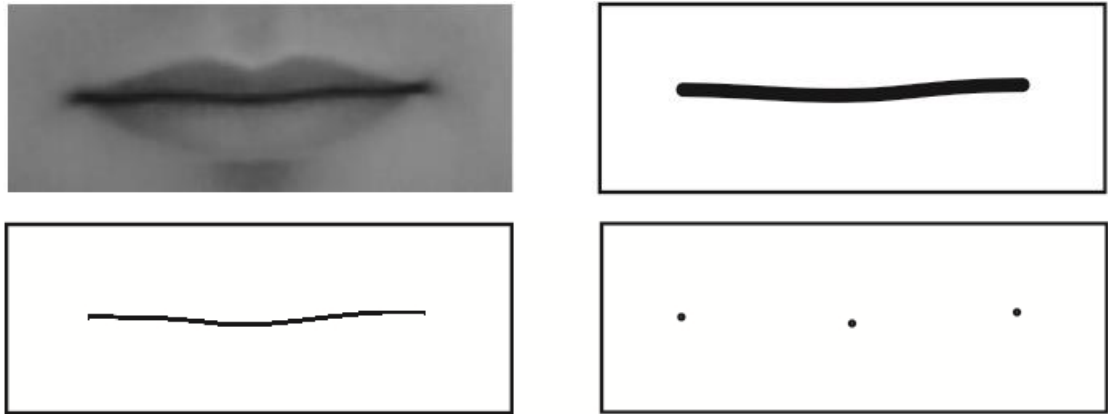


Рис. 16 Процес виділення ключових точок

Перехід від кольорового зображення до напівтонового виконується на етапі виділення обличчя [22]. Для отримання бінарного зображення використовується адаптивний поріг. Застосування градієнтної маски до бінарного зображення дозволяє отримати контурне уявлення аналізованого елемента. Локалізація ключових точок полягає в визначенні заданої кількості точок, що лежать на контурі елемента.

Після виділення ключових точок можна виконати класифікацію емоції (табл. 1).

Таблиця 1

Класифікація емоцій за допомогою ключових точок

Емоція	Брови	Рот
Подив	Піднімаються	Відкривається
Страх	Піднімаються і зморщуються	Відкривається і розтягується
Відраза	Знижуються	Піднімається і кінці зужуються
Гнів	Знижуються і зморщуються	Відкривається або кінці знижуються
Щастя	Піднімаються	Кінці піднімаються
Смуток	Кінці знижуються	Кінці знижуються

2.3.2 Згорткові нейронні мережі

Згорткові нейронні мережі (convolutional neural network, CNN) широко застосовується в багатьох підходах комп'ютерного зору, включаючи проблему розпізнавання емоцій. На початку XXI століття в кількох дослідженнях літератури розпізнавання емоцій обличчя, встановлено, що CNN є надійним підходом у випадку змін місця розташування та варіаціями масштабу і поведінки краще, ніж багат шаровий перцептрон (MLP) у випадку раніше небачених варіацій поз обличчя [23].

Згорткова нейронна мережа являє собою особливий клас багат шарового перцептрона, який володіє двовимірної структурою і добре підходить для обробки зображень з високим ступенем інваріантності до зсуву, поворотам, масштабування і іншим спотворень вхідних даних.

Структура нейронної мережі являє собою послідовність з двох типів шарів: згорткові і підвборчі. Кожен шар складається з набору площин (карт характеристик), які в свою чергу складаються з нейронів [23].

Кожен нейрон згорткового шару має зв'язок з невеликою групою нейронів попереднього шару (локальне рецептивне поле). Локальні рецептивні

поля нейронів згорткового шару частково накладаються один на одного за принципом черепиці. Значення нейронів з локального рецептивного поля множаться на матрицю синаптичних коефіцієнтів, а результат записується у відповідний нейрон згорткового шару.

Слідом за згортковим шаром розташовується підвиборчий шар, який забезпечує часткову інваріантність нейронної мережі до зміни масштабу вхідного зображення. Кількість площин підвиборчого шару зазвичай таке ж, як і в попередньому шарі. У підвиборчому шарі локальні рецептивні поля не перетинаються один з одним і мають фіксований розмір 2×2 нейрона [24]. Кожен нейрон даного шару обчислює середнє значення своїх чотирьох входів, примножує їх на вагові коефіцієнти синапсів і отриманий результат передає через функцію активації (Рис. 17). Таким чином, підвиборочний шар зменшує розмірність площин попереднього шару в два рази.

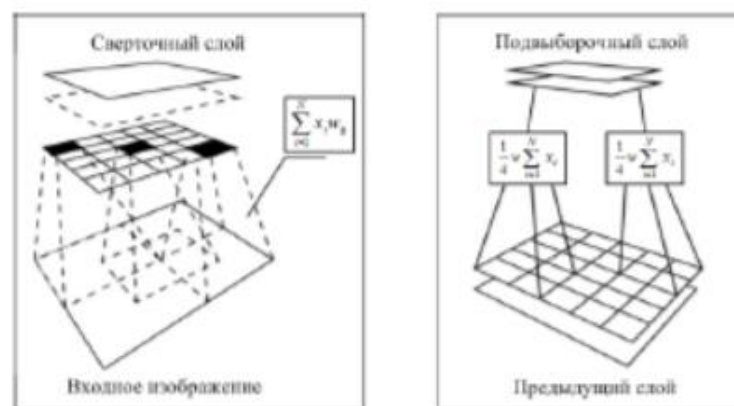


Рис. 17 Функціонування шарів згорткової нейронної мережі

Послідовно чергуючись один за одним, розміри площин зменшуються, але їх кількість збільшується. Чергування шарів дозволяє формувати різні карти характеристик, що наділяє згорткову нейронну мережу здатністю до ідентифікації більш складних ієрархічних ознак.

Поступово при проходженні декількох шарів карта ознак вироджується в вектор. Останні кілька шарів нейронної мережі являють собою класичний перцептрон, вони складаються зі звичайних нейронів і призначені для класифікації виділених ознак.

Важливою властивістю згорткової нейронної мережі є використання загальної матриці вагових коефіцієнтів для всіх нейронів в межах площині. Даний підхід дозволяє використовувати меншу кількість параметрів, що настраюються (вагових коефіцієнтів) при великій кількості зв'язків і підвищити швидкість процесу навчання [25]. Таким чином, площини нейронної мережі являють собою фільтри, кожен з яких здійснює пошук індивідуальних характерних ознак вхідного зображення. Це дозволяє згортковій нейронній мережі запам'ятовувати взаємозв'язок просторово залежних областей зображення. Характерні ознаки, які добуваються тій чи іншій площиною, визначаються в процесі навчання.

Якщо вхідне зображення спотворено або зміщене, то на виході площини буде аналогічно зміщений результат [24]. Завдяки цій властивості забезпечується стійкість нейронної мережі до спотворень вхідних даних.

Недолік використання згорткових нейронних мереж полягає в складності настройки оптимальних параметрів: кількість шарів, площин, нейронів, розмір рецептивного поля і т.д.

Переваги використання згорткових нейронних мереж для детектування об'єктів на зображеннях:

- Структура нейронної мережі добре підходить для обробки двовимірних даних.
- Нашаровуються один на одного рецептивні поля забезпечують взаємозв'язок просторово залежних областей зображення.
- Підвищена стійкість до афінних і проєкційним спотворень вхідних даних, шумів, зміни масштабу. Завдяки своїм перевагам на сьогоднішній день СНС активно використовуються для виявлення осіб та інших об'єктів на зображеннях і відео послідовностях.

2.3.3 Рекурентні нейронні мережі

Рекурентні нейронні мережі (Recurrent neural network, RNN) — вид нейронних мереж, де зв'язки між елементами утворюють спрямовану послі-

довність. Завдяки цьому з'являється можливість обробляти серії подій у часі або послідовні просторові ланцюжки. На відміну від багатошарових перцептронів, рекурентні мережі можуть використовувати свою внутрішню пам'ять для обробки послідовностей довільної довжини. Тому мережі RNN застосовні в таких завданнях, де щось цілісне розбите на частини, наприклад: розпізнавання рукописного тексту, розпізнавання емоцій або розпізнавання мови.

Існує багато різновидів, рішень і конструктивних елементів рекурентних нейронних мереж.

Труднощі рекурентної мережі полягають в тому, що якщо враховувати кожен крок часу, то стає необхідним для кожного кроку часу створювати свій шар нейронів, що викликає серйозні обчислювальні складності. Крім того, багатошарові реалізації виявляються обчислювально нестійкими, так як в них як правило зникають або зашкалюють ваги. Якщо обмежити розрахунок фіксованим тимчасовим вікном, то отримані моделі не будуть відображати довгострокових трендів [26]. Різні підходи намагаються вдосконалити модель історичної пам'яті і механізм запам'ятовування і забування.

Повністю рекурентна мережа, базова архітектура розроблена в 1980-х. Мережа будується з вузлів, кожен з яких з'єднаний з усіма іншими вузлами. У кожного нейрона поріг активації змінюється з часом і є дійсним числом. Кожне з'єднання має змінну речовинну вагу. Вузли поділяються на вхідні, вихідні і приховані.

Для навчання з учителем з дискретним часом, кожен (дискретний) крок часу на вхідні вузли подаються дані, а інші вузли завершують свою активацію, і вихідні сигнали готуються для передачі нейроном наступного рівня. Якщо наприклад мережу відповідає за розпізнавання мови, в результаті на вихідні вузли надходять вже мітки (розпізнані слова).

У навчанні з підкріпленням (reinforcement learning) немає вчителя, що забезпечує цільові сигнали для мережі, замість цього іноді використовується функція пристосованості (придатності) або функція оцінки (reward function), по якій проводиться оцінка якості роботи мережі, при цьому значення на ви-

ході впливає на поведінку мережі на вході [26]. Зокрема, якщо мережа реалізує гру, на виході вимірюється кількість пунктів виграшу або оцінки позиції.

Кожен ланцюжок обчислює помилку як сумарну девіацію по вихідним сигналам мережі. Якщо є набір зразків навчання, помилка обчислюється з урахуванням помилок кожного окремого зразка.

2.4 Згорткові нейронні мережі для розпізнавання об'єктів з зображень

2.4.1 Архітектура AlexNet

AlexNet — згорткова нейронна мережа, яка відіграла великий вплив на розвиток машинного навчання, особливо на алгоритми комп'ютерного зору. Мережа з великим відривом виграла конкурс з розпізнавання зображень ImageNet LSVRC-2012 у 2012 році (з кількістю помилок 15,3% проти 26,2% у другого місця) [27].

Архітектура AlexNet схожа до створеної Yann LeCun мережею LeNet. Однак у AlexNet більше фільтрів у шарі і вкладених згорткових шарів. Мережа включає в себе згортки, максимальне об'єднання, дропаутов, аугментацію даних, функції активацій ReLU і стохастичний градієнтний спуск.

AlexNet містить вісім шарів з ваговими коефіцієнтами. Перші п'ять з них згорткові, а решта три - повнозв'язні. Вихідні дані пропускаються через функцію втрат softmax, яка формує розподіл +1000 міток класів. Мережа максимізує багатолінійну логістичну регресію, що еквівалентно максимізації середнього по всім навчальним випадків логарифма ймовірності правильного маркування з розподілу очікування. Ядра другого, четвертого і п'ятого згорткових шарів пов'язані тільки з тими картами ядра в попередньому шарі, які знаходяться на одному і тому ж графічному процесорі. Ядра третього згорткового шару пов'язані з усіма картами ядер другого шару. Нейрони в повнозв'язних шарах пов'язані з усіма нейронами попереднього шару [27].

Таким чином, AlexNet містить 5 згортальних шарів і 3 повнозв'язних шари як проілюстровано на рисунку 18.

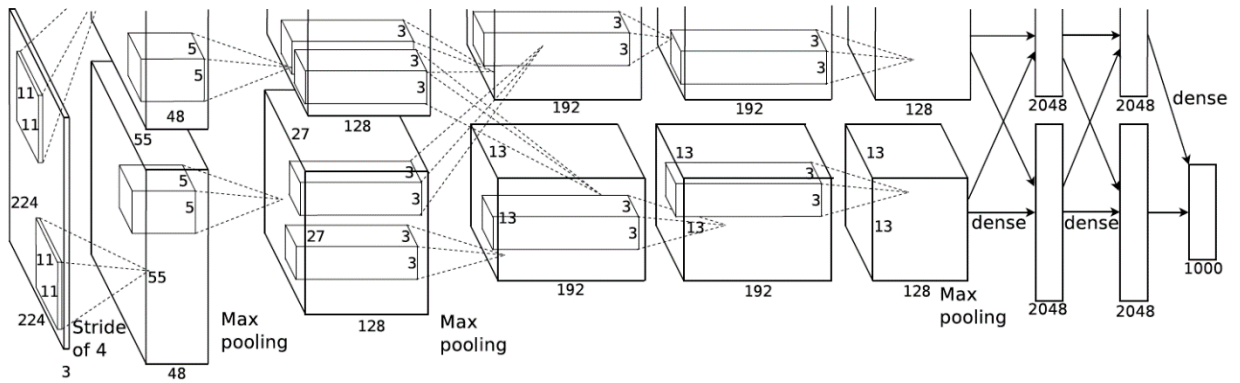


Рис. 18 Архітектура нейронної мережі AlexNet

Relu застосовується після кожного згорткового і повнозв'язну шару. Дропаут застосовується перед першим і другим повнозв'язними шарами. Мережа містить 62,3 мільйона параметрів і витрачає 1,1 мільярда обчислень при прямому проході. Згорткові шари, на які припадає 6% всіх параметрів, виробляють 95% обчислень.

AlexNet проходить 90 епох. Навчання займає 6 днів одночасно на двох графічних процесорах Nvidia Geforce GTX 580, що є причиною того, що мережа розділена на дві частини. Використовується стохаїстичний градієнтний спуск зі швидкістю навчання 0,01, імпульсом 0,9 і розпадом вагових коефіцієнтів 0,0005 [27]. Швидкість навчання ділиться на 10 після насичення точності і знижується в 3 рази протягом навчання.

Мережа досягає наступного рівня помилок першого і п'ятого рівнів: 37,5% і 17,0%, відповідно. Краща продуктивність, досягнута в ході конкурсу ILSVRC-2010, становила 47,1% і 28,2% при використанні підходу, в якому усереднюються передбачення, отримані шістьма моделями з розрідженим кодуванням, навчених на різних векторах властивостей. З тих пір досягнуті результати: 45,7% і 25,7% при використанні підходу, в якому усереднюються передбачення двох класифікаторів, яких навчають на векторах Фішера.

2.4.2 Архітектура VGG16

VGG16 — модель згорткової нейронної мережі, запропонована К. Simonyan і А. Zisserman з Оксфордського університету в статті "Very Deep Convolutional Networks for Large-Scale Image Recognition". Модель досягає точності 92.7% - топ-5, при тестуванні на ImageNet в задачі розпізнавання об'єктів на зображенні. Цей датасет складається з більш ніж 14 мільйонів зображень, що належать до 1000 класів.

VGG16 — одна з відомих моделей, відправлених на змагання ILSVRC-2014. Вона є покращеною версією AlexNet, в якій замінені великі фільтри (розміру 11 і 5 в першому і другому згортковому шарі, відповідно) на кілька фільтрів розміру 3x3, що йдуть один за іншим [28]. Мережа VGG16 навчалася протягом декількох тижнів при використанні відеокарт NVIDIA TITAN BLACK.

Архітектура VGG16 представлена на рисунку 19.

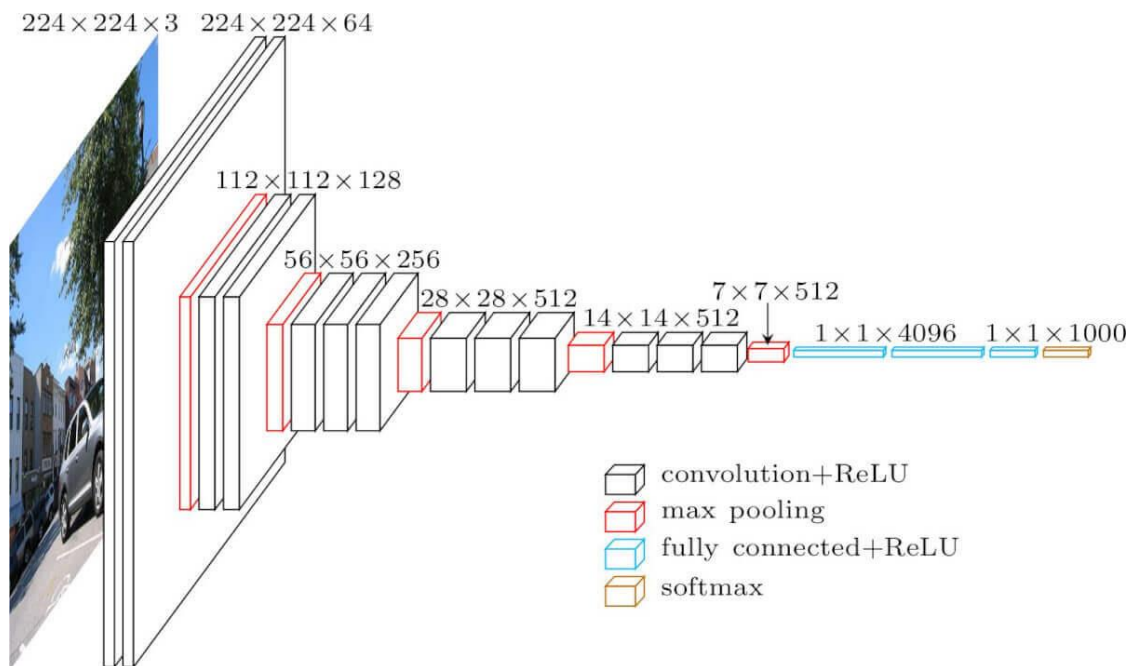


Рис. 19 Архітектура нейромережі vgg16

Розмір ядра мережі - 3×3 , а розмір пулу - 2×2 для всіх рівнів.

На рисунку 19 усі сині прямокутники представляють згорткові шари разом з функцією нелінійної активації, ReLU. Як видно з малюнка, є 13 синіх і 5 червоних прямокутників, тобто є 13 згорткових шарів і 5 шарів максима-

льного об'єднання. Поряд з ними є 3 зелених прямокутника, що представляють 3 повністю пов'язаних шару. Таким чином, загальна кількість шарів з налаштованими параметрами становить 16, з яких 13 призначені для згорткових шарів і 3 - для повнозв'язних шарів, тому ім'я дається як VGG-16. На виході маємо шар softmax, що має 1000 виходів для кожної категорії зображень в наборі даних imagenet.

Мережа VGG має два серйозні недоліки:

- Дуже повільна швидкість навчання.
- Сама архітектура мережі важить надто багато (з'являються проблеми з диском і пропускною спроможністю).

Через глибину і кількість повнозв'язних вузлів, VGG16 важить більше 533 МБ. Це робить процес розгортання VGG стомлюючою завданням. Незважаючи на недоліки, дана архітектура є відмінним будівельним блоком для навчання, так як її легко реалізувати.

2.4.3 Архітектура GoogLeNet

Нейронна мережа GoogLeNet (або Inception V1) була запропонована дослідженням Google (у співпраці з різними університетами) в 2014 році в дослідницькій статті під назвою «Going Deeper with Convolutions» [29]. Ця архітектура стала переможцем у випробуванні класифікації зображень ILSVRC 2014. Це забезпечило значне зниження кількості помилок в порівнянні з попередніми переможцями AlexNet (Переможець ILSVRC 2012) і ZF-Net (Переможець ILSVRC 2013) і значно менша кількість помилок, ніж VGG (друге місце в 2014 році). Ця архітектура використовує такі методи, як згортки 1×1 в середині архітектури і глобальний середній пул (global average pooling), що дозволяє створювати більш глибоку архітектуру.

У архітектурі Inception використовується згортка 1×1 . Ці згортки використовуються для зменшення кількості параметрів (ваги і зміщення) архітектури. Зменшуючи параметри, також збільшується глибина архітектури.

У попередній архітектурі, такий як AlexNet, повнозв'язні рівні використовувалися в кінці мережі. Ці повнозв'язні рівні містять більшість параметрів багатьох архітектур, що призводить до збільшення вартості обчислень. В архітектурі GoogLeNet в кінці мережі використовується метод, званий глобальним середнім пулом. Цей шар бере карту об'єктів 7×7 і усереднює її до 1×1 . Це також зменшує кількість навчальних параметрів до 0 і підвищує точність на 0,6%.

Inception модуль відрізняється від попередніх архітектур, таких як AlexNet, ZF-Net. У цій архітектурі існує фіксований розмір згортки для кожного рівня. У модулі Inception 1×1 , 3×3 , 5×5 згортка і максимальне об'єднання 3×3 , виконуються паралельно на вході і виході, складаються разом для генерування остаточного виведення [29]. Ідея, що лежить в основі згорткових фільтрів різних розмірів, дозволяє краще обробляти об'єкти в декількох масштабах.

У початковій архітектурі використовувалися деякі проміжні гілки класифікатора в середині архітектури, ці гілки використовуються тільки під час навчання. Ці гілки складаються з середнього рівня об'єднання 5×5 з кроком 3, згортки 1×1 з 128 фільтрами, двох повністю пов'язаних шарів з 1024 виходами і 1000 виходів і шару класифікації softmax. Втрати, що утворилися в цих шарах, додаються до загальних втрат з вагою 0,3. Ці шари допомагають боротися з проблемою зникнення градієнта, а також забезпечують регуляризацию.

Загальна архітектура складається з 22 шарів. Архітектура була розроблена з урахуванням обчислювальної ефективності. Ідея, що лежить в основі архітектури, може бути запущена на окремих пристроях навіть з невеликими обчислювальними ресурсами. Архітектура також містить два допоміжних рівня класифікатора.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Рис. 20 Модель архітектури нейронної мережі GoogLeNet

Ця архітектура дозволяє отримувати зображення розміром 224x224 з кольорними каналами RGB. Всі згортки всередині цієї архітектури використовують Rectified Linear Units (ReLU) в якості функцій активації.

2.4.4 Архітектура ResNet-50

ResNet, скорочення від Residual Networks — це класична нейронна мережа, яка використовується для вирішення багатьох завдань комп'ютерного зору. Ця модель стала переможцем змагання ImageNet у 2015 році. Основним проривом з ResNet було те, що вона дозволила успішно тренувати надзвичайно глибокі нейронні мережі із 150+ шарами [30]. До реалізації ResNet навчання глибоких нейронних мереж було важким процесом через проблему зникнення градієнтів.

Основна ідея ResNet полягає у реалізації терміну «з'єднання швидкого доступу» (shortcut connections), що пропускає один або кілька шарів і виконує зіставлення ідентифікаторів. Їх виходи додаються до виходів stacked layers.

Автори ResNet припустили, що трудність криється в оптимізації - більш глибокі моделі набагато гірше піддаються налаштуванню. Тоді вони вирішили не складати шари один на одного для вивчення відображення потрібної функції безпосередньо, а використовувати залишкові блоки, які намагаються «підігнати» це відображення. Так ResNet стала першою залишковою нейронною мережею (Residual neural network). Шари більше не містять ознак і використовуються для знаходження залишкової функції $H(x) = F(x) + x$ замість того, щоб шукати $H(x)$ безпосередньо (Рис. 21).

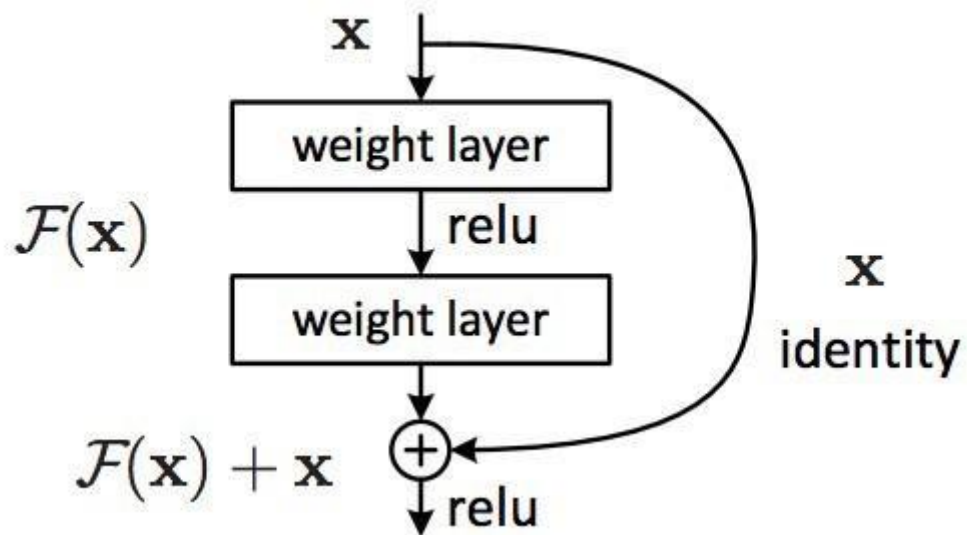


Рис. 21 Архітектура нейронної мережі ResNet

Нейронна мережа складається з великого стека однакових залишкових блоків, кожен з яких має два згорткових шари 3×3 . Періодично число фільтрів подвоюється, а їх розмірність зменшується з кроком 2 ($/ 2$ в кожному вимірі) [30]. На самому початку архітектури присутній додатковий згортковий шар. Також у ResNet немає повнозв'язних шарів в кінці, використовується тільки один шар з вихідними класами. Зі збільшенням числа шарів для зменшення розмірності зображення застосовуються такі самі додаткові шари, як і в GoogleNet. В якості фінального класифікатора в ResNet використовується pooling-шар з softmax.

Після кожного згорткового шара використовується пакетна нормалізація. Розмір міні-пакета 256. Швидкість навчання - 0.1, ділиться на 10 при загасання швидкості зміни помилки. Особливість архітектури - Dropout не використовується.

2.5 Датасети для розпізнавання емоцій на обличчі

Набір даних виразів обличчя — це колекція зображень або відеокліпів з мімікою різних емоцій. Добре анотований (позначений емоціями) медіа-вміст поведінки обличчя є важливим для навчання, тестування та перевірки алгоритмів для розробки систем розпізнавання облич. Анотація емоцій може бути зроблена на дискретних мітках емоцій або в безперервному масштабі. Більшість баз даних зазвичай базуються на теорії основних емоцій Пола Екмана, яка передбачає існування шести дискретних базових емоцій (гнів, страх, огида, здивування, радість, смуток). Однак деякі бази даних включають позначення емоцій у безперервному масштабі валентності збудження.

У базах даних висловлених емоцій учасникам пропонується відображати різні основні емоційні вирази, тоді як у базі спонтанних виразів вирази є природними. Спонтанні вирази надзвичайно відрізняються від поставлених за інтенсивністю, конфігурацією та тривалістю.

Датасет F2ED

F2ED — це датасет з розміченими зображеннями людських облич. Датасет призначений для вирішення завдання end-to-end розпізнавання емоцій за даними з камер спостереження. Дані збиралися дослідниками з Fudan University і Ping An OneConnect.

Датасет складається з 200 тисяч зображень 119 людей в 4-х різних позах і з 54 емоціями на обличчі. Дослідники також використовували техніки аугментації даних, щоб збільшити розмір даних. Для аугментації використовувалася генеративна нейронна мережа - facial pose generative adversarial

network (FaPE-GAN) [31]. FaPE-GAN генерувала додаткові зображення облич людей в різних позах. Ці додаткові зображення додавалися в навчальну вибірку.

Щоб зібрати зображення людей, дослідники запросили добровольців в кімнату, заповнену відеокамерами, на півгодини. Доброволець спілкувався з двома психологами-експертами, поки перебував у кімнаті. Після цього три психолога переглядали відеозапис і розмічали емоції на обличчі добровольця. Датасет включає в себе тільки ті відеозаписи, де всі три психолога не мали розбіжностей по розмітці. Кожен учасник знімався з чотирьох різних сторін: анфас, ліворуч, праворуч і зверху.

Психологи-експерти розмітили 54 різних вирази на обличчі. Ці вирази були пов'язані з емоціями. Ці вирази включають в себе нудьгу, страх, оптимістичність, агресивність, несхвалення і подібне.

Датасет EMOTIC

Набір даних EMOTIC, названий на честь EMOTions In Context, являє собою базу даних зображень із людьми в реальному середовищі, анотованих своїми очевидними емоціями. До зображень додано розширений список з 26 категорій емоцій, поєднаних з трьома загальними безперервними вимірами: Валентність, Збудженість та Домінантність.

На меті було створити базу даних природних зображень, фіксуючи об'єкти та їх контекст із їх природним необмеженим середовищем. Розробники датасету розпочали з того, що збирали зображення з усталених наборів даних, таких як MSCOCO та ADE20K [31]. Ці набори даних містять велику кількість зображень, які відповідали критеріям. Також були завантаженні зображення після пошуку в пошуковій системі Google. Розробники використовували різноманітну комбінацію слів, що представляють різноманітну суміш предметів, місць, ситуацій та контекстів. Це призвело до складної колекції зображень, які поєднуються з зображеннями людей, що перебувають у різних

ситуаціях, виконуючи різні завдання та демонструючи широкий спектр емоційних станів.

В даний час база даних EMOTIC складається з 18316 зображень з позначкою 23788 людей.

Датасет FER2013

Набір даних Facial Expression Recognition 2013 (FER2013) був представлений на конференції International Conference on Machine Learning 2013. Цей набір даних містить 35 887 зображень з роздільною здатністю 48×48 пікселів, більшість з яких зроблені в довільних умовах.

База даних була створена з використанням інструментів пошуку зображень Google. Кожне зображення класифіковано одним з семи видів емоцій: здивування (surprise), страх (fear), щастя (happy), гнів (angry), відраза (disgust), смуток (sad) і нейтральне стан або спокій (neutral). FER має велике число варіацій в зображеннях, включаючи часткове закриття особи (в основному, за допомогою руки), низько контрастні зображення і особи в окулярах.

Весь набір даних FER розділений на три частини: навчальний набір, валідаційний набір і тестовий набір. Перші два беруть участь при навчанні мережі: навчальний набір використовується для оптимізації ваг моделі, а валідаційний набір надає метрики після кожної епохи навчання, які допомагають оцінити якість навчання моделі. Тестовий набір необхідний для порівняння точності розпізнавання серед різних моделей.

2.6 Фреймворки глибинного навчання

Глибинне навчання базується на алгоритмах, які є дуже складні для розуміння та роботи з ними. Тому при роботі з алгоритмами глибинного навчання часто використовують фреймворки глибинного навчання.

Фреймворк — це набір інструментів і стандартних реалізацій для забезпечення більш швидкої розробки будь-якого програмного продукту. Його

часто плутають з бібліотекою, яка ж, в свою чергу, є набором стандартних реалізацій, функцій і структур даних, що полегшують рішення будь-якої задачі.

Існує безліч фреймворків глибокого навчання, спрямованих на різні цілі. Майже всі фреймворки глибокого навчання написані на Python. Python це переважаюча мова програмування для глибокого навчання. На рисунку 22 продемонстровано найпопулярніші фреймворки для глибокого навчання зі статистикою їх використання.

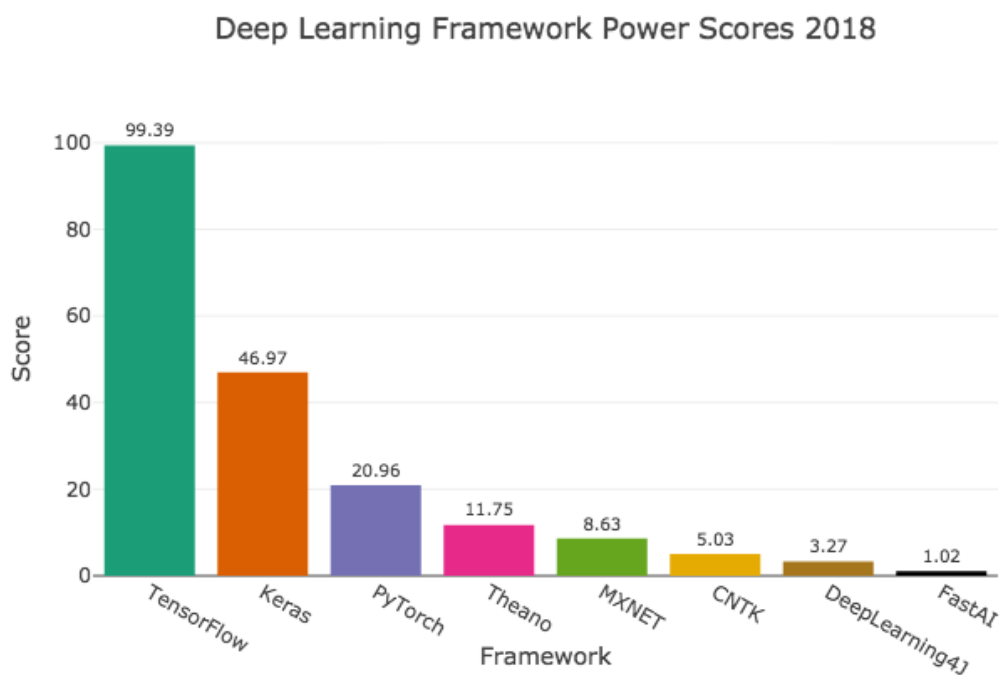


Рис. 22 Статистика використання фреймворків глибокого навчання

Фреймворк TensorFlow

TensorFlow — безперечно одна з найпопулярніших систем глибокого навчання. Розроблений командою Google Brain, TensorFlow підтримує такі мови, як Python, C++ та R, для створення моделей глибокого навчання разом із бібліотеками обгортки. Він доступний як на настільному ПК, так і на мобільному.

Найвідомішим варіантом використання TensorFlow повинен бути Google Translate у поєднанні з такими можливостями, як обробка природної

мови, класифікація тексту, узагальнення, розпізнавання мови/зображення/рукописного вводу, прогнозування та позначення тегами.

Набір інструментів візуалізації TensorFlow, TensorBoard, забезпечує ефективну візуалізацію даних моделювання та продуктивності мережі.

TensorFlow Serving, інший інструмент TensorFlow, використовується для швидкого розгортання нових алгоритмів / експериментів, зберігаючи ту саму архітектуру сервера та API [32]. Він також забезпечує інтеграцію з іншими моделями TensorFlow, що відрізняється від звичайних практик і може бути розширено для обслуговування інших моделей та типів даних.

TensorFlow — одна з найбільш улюблених платформ глибокого навчання, оскільки вона базується на Python, підтримується Google і постачається з першокласною документацією та покроковими інструкціями, які допомагають розробникам.

Переваги використання фреймворку TensorFlow:

- Велика кількість посібників та документації.
- Потужні інструменти для моніторингу процесу навчання моделей та візуалізації (Tensorboard).
- Велика спільнота розробників та технологічних компаній.
- Надання моделей обслуговування.
- Підтримка розподіленого навчання.

Недоліки використання фреймворку TensorFlow:

- Низька зручність для користувачів.
- Не найшвидший у бенчмаркінгу.

Фреймворк Keras

Написана на Python, бібліотека нейронних мереж Keras підтримує як згорткові, так і періодичні мережі, які можуть працювати на TensorFlow або Theano.

Оскільки інтерфейс TensorFlow трохи складний і може бути незрозумілим для нових користувачів, програма глибокого навчання Keras була побу-

дована, щоб забезпечити спрощений інтерфейс для швидкого створення прототипів шляхом побудови активних нейронних мереж, які можуть працювати з TensorFlow.

Keras є легким, простим у використанні та має мінімалістичний підхід. Ось ці причини, чому Keras є частиною основного API TensorFlow.

Основне використання Keras це класифікація, генерація тексту та узагальнення, позначення, переклад разом із розпізнаванням мови та інші.

Keras підтримує широкий спектр нейронних мережевих шарів: згортковий, періодичний та щільний. Бібліотека містить численні реалізації широко використовуваних будівельних блоків нейронних мереж: шари, цільові та передавальні функції, оптимізатори, а також різноманітні інструменти, що спрощують роботу із зображеннями та текстом [32].

Keras — одна з найкращих систем глибокого навчання для перекладу, розпізнавання зображень та мови.

Переваги використання фреймворку Keras:

- Швидке і просте створення прототипів.
- Наявність документації.
- Має модулі, що налаштовуються.
- Інтуїтивно зрозумілий інтерфейс.
- Вбудована підтримка викладання в декількох графічних процесорах.
- Вбудований всередині TensorFlow.

Недоліки використання фреймворку Keras:

- Непридатний для великих проектів.
- Непростий для налаштування.

Фреймворк PyTorch

Torch — це наукова обчислювальна база, яка пропонує широку підтримку алгоритмів машинного навчання. Це основа глибокого навчання на осно-

ві Lua, яка широко використовується серед таких гігантів галузі, як Facebook, Twitter та Google [33].

У ньому використовується CUDA разом із бібліотеками C/C++ для обробки. На відміну від Torch, PyTorch працює на Python, що означає, що кожен, хто має базове розуміння Python, може розпочати розробку своїх моделей глибокого навчання.

В останні роки PyTorch спостерігав високий рівень прийняття в рамках спільноти глибоких навчальних програм і вважається цілком конкурентом TensorFlow. PyTorch — це, по суті, порт поглибленого навчання Torch, який використовується для побудови нейронних мереж та виконання тензорних обчислень, які є високими з точки зору складності.

Переваги використання фреймворку PyTorch:

- Відмінно володіє швидкими прототипами.
- Потужна підтримка графічних процесорів, оскільки паралельні програми можуть бути реалізовані на декількох графічних процесорах.
- Забезпечує більш чистий інтерфейс, простіший у використанні.
- Сприяє обміну даними із зовнішніми бібліотеками.

Недоліки використання фреймворку PyTorch:

- Відсутність підтримки моделі.
- Відсутність інтерфейсів моніторингу та візуалізації.

Фреймворк Deeplearning4j

J у Deeplearning4j означає Java. Само собою зрозуміло, що це бібліотека глибокого навчання для віртуальної машини Java (JVM). Він розроблений на Java та підтримує інші мови JVM, такі як Scala, Clojure та Kotlin [33].

Паралельне навчання за допомогою ітеративного зменшення, адаптація архітектури мікросервісу в поєднанні з розподіленими центральними процесорами та графічними процесорами є одними з основних особливостей, коли мова йде про основу глибокого навчання Eclipse Deeplearning4j.

Широко прийнята як комерційна, орієнтована на галузь та розподілена платформа глибокого навчання, DeepLearning4j поставляється з глибокою мережевою підтримкою через RBM, DBN, згорткові нейромережі (CNN), рекурентні нейронні мережі (RNN), рекурсивну нейронну тензорну мережу (RNTN) [32].

Оскільки ця основа глибокого навчання реалізована на Java, вона набагато ефективніша в порівнянні з Python. Що стосується завдань розпізнавання зображень з використанням декількох графічних процесорів, DL4J швидкий, ніж Caffe. Цей фреймворк показує незрівнянний потенціал для розпізнавання зображень, виявлення шахрайства, видобування тексту, позначення частин мови та обробки природної мови.

Переваги використання фреймворку DeepLearning4j:

- Об'єднує всю екосистему Java для глибинного навчання.
- Може швидко обробляти великі обсяги даних.
- Включає як багатопотокові, так і однопотокові основи глибокого навчання.
- Можна вводити поверх Hadoop та Spark.

Недоліки використання фреймворку DeepLearning4j:

- Використання Java для машинного навчання, як правило, є дорогим і трудомістким способом.

2.7 Обґрунтування вибраних засобів для реалізації системи

Огляд дав змогу вибрати метод Віоли-Джонса в якості методу розпізнавання обличчя. В якості методу розпізнавання емоцій було вирішено побудувати згорткову нейронну мережу. Датасетом для навчання нейронної мережі був обраний датасет Facial Expression Recognition 2013 (FER2013).

В якості фреймворку для роботи з нейронними мережами було обрано фреймворк TensorFlow та Keras API. Вибір було зроблено виходячи з таких функціональних можливостей:

1. TensorFlow забезпечує доступний і читабельний синтаксис, який є важливим для спрощення використання цих ресурсів програмування.
2. TensorFlow — це бібліотека низького рівня, яка забезпечує більшу гнучкість.
3. TensorFlow забезпечує більше управління мережею. Таким чином, розробники та дослідники можуть зрозуміти, як операції реалізуються в мережі. Вони завжди можуть відстежувати нові зміни, внесені з часом.
4. Keras — це високорівневий API, який працює поверх TensorFlow, CNTK та Theano, тоді як TensorFlow - це структура, яка пропонує API високого та низького рівня.
5. Keras ідеально підходить для швидкого впровадження.

РОЗДІЛ 3 ПРОЕКТ ПРОГРАМНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ НА ОБЛИЧЧІ ЛЮДИНИ

3.1 Архітектура системи

Архітектуру розробленої системи можна поділити на три основні модуля: Веб-застосунок, Сервер та Модуль розпізнавання емоцій людини з відео-потоків.

Веб-застосунок розроблений за допомогою інструментів HTML, CSS та JavaScript. Сервер реалізований за допомогою багатофункціонального серверного веб-фреймворку Django, що написаний на Python. Модулем розпізнавання емоцій людини виступає розроблена згортована нейронна мережа.

Веб-застосунок отримує доступ до веб-камери користувача і пересилає 1 кадр в секунду з відео до серверу. Сервер отримує кадр і передає його модулю розпізнавання емоцій. Передача кадрів відбувається за допомогою веб-сокетів.

Клієнт-серверна архітектура

Архітектура «клієнт-сервер» визначає загальні принципи організації взаємодії в мережі, де є сервери, вузли-постачальники деяких специфічних функцій (сервісів) і клієнти, споживачі цих функцій.

В основі клієнт-серверної архітектури лежать два компоненти: клієнт і сервер.

Клієнт — комп'ютер на стороні користувача, який відправляє запит до сервера для надання інформації або виконання певних дій.

Сервер — більш потужний комп'ютер або обладнання, призначене для вирішення певних завдань з виконання програмних кодів, виконання сервісних функцій за запитом клієнтів, надання користувачам доступу до певних ресурсів, зберігання інформації і баз даних.

Модель такої системи полягає в тому, що клієнт відправляє запит на сервер, де він обробляється, і готовий результат відправляється клієнтові. Сервер може обслуговувати кілька клієнтів одночасно. Якщо одночасно приходить більше одного запиту, то вони встановлюються в чергу і виконуються сервером послідовно. Іноді запити можуть мати пріоритети. Запити з більш високими пріоритетами повинні виконуватися раніше.

Функції, які реалізуються на сервері:

- Зберігання, доступ, захист і резервне копіювання даних.
- Обробка клієнтського запиту.
- Відправлення результату (відповіді) клієнту.

Функції, які реалізуються на стороні клієнта:

- Надання користувальницького інтерфейсу.
- Формулювання запиту до сервера і його відправка.
- Отримання результатів запиту і відправка додаткових команд (запитів на додавання, оновлення або видалення даних).

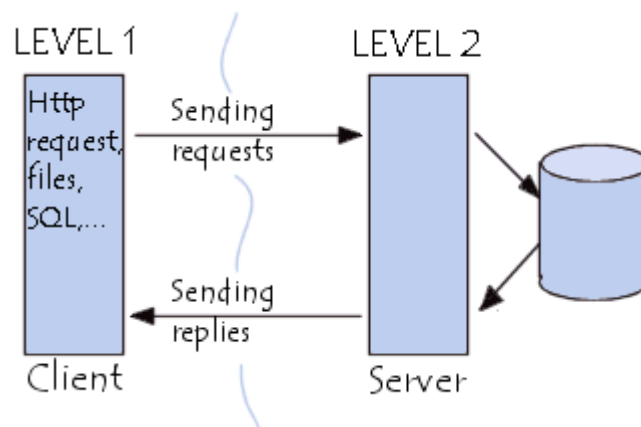


Рис. 23 Двухрівнева клієнт-серверна архітектура

Двухрівнева архітектура використовується в клієнт-серверних системах, де сервер відповідає на клієнтські запити безпосередньо і в повному обсязі, при цьому використовуючи тільки власні ресурси. Тобто сервер не викликає сторонні мережеві програми, але не звертається до сторонніх ресурсів для виконання будь-якої частини запиту.

Також існує трирівнева архітектура, що складається з наступних компонентів:

- Представлення даних — призначений для користувача інтерфейс.
- Прикладної компонент — сервер додатків.
- Управління ресурсами — сервер бази даних, який надає інформацію.

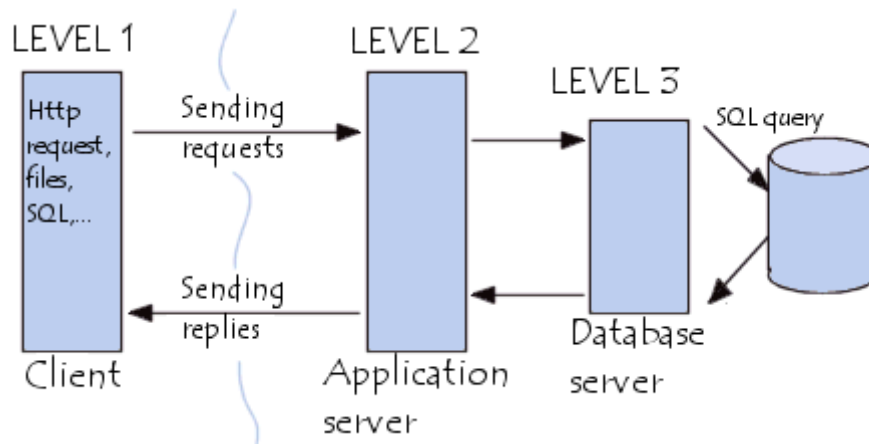


Рис. 24 Трирівнева архітектура

Принцип роботи полягає в тому, що кілька серверів обробляють запит клієнта. Розподіл операцій знижує навантаження на сервер.

3.2 Засоби реалізації

Під час реалізації програмної системи були використані наступні засоби:

1. Мова програмування Python.
2. Редактор коду Visual Studio Code.
3. HTML, CSS, JavaScript.
4. Фреймворк Django.
5. Протокол WebSockets.
6. Бібліотека Django Channels.
7. Бібліотека OpenCV.

8. Фреймворк Tensorflow та Keras API.

3.2.1 Мова програмування Python

Python є інтерпретованою, спочатку об'єктно-орієнтованою мовою програмування. Вона надзвичайно проста і містить невелику кількість ключових слів, в той же час дуже гнучка і виразна.

Безсумнівним достоїнством є те, що інтерпретатор Python реалізований практично на всіх платформах і операційних системах [34].

Наступна важлива риса — розширюваність мови, цьому надається велике значення. Це означає, що є можливість вдосконалення мови усіма бажаними. Інтерпретатор написаний на C і вихідний код доступний для будь-яких маніпуляцій.

Наступний плюс — наявність великого числа підключаємих до програми модулів, що забезпечують різні додаткові можливості. Такі модулі пишуться на C і на самій Python і можуть бути розроблені усіма досить кваліфікованими програмістами.

Мову Python можна зустріти у веб-додатках і на мобільних пристроях, в додатках і рішеннях, пов'язаних з машинним навчанням (нейромережі і штучний інтелект), а також в якості вбудованої системи.

Незважаючи на всі переваги, у мови є і недоліки.

- Програми на Python вважаються одними з найбільш повільних.
- Сильна залежність мови від системних бібліотек.
- Global Interpreter Lock (GIL) не дозволяє виконувати кілька потоків Python одночасно в реалізації CPython.

3.2.2 Редактор коду Visual Studio Code

Visual Studio Code — це сервіс, який позиціонується як «легкий» редактор коду для кросплатформеної розробки веб-і хмарних додатків.

Особливості Visual Studio Code

- VS Code дозволяє розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керуваному кодах для всіх платформ.
- У редакторі присутні вбудований відладчик, інструменти для роботи з Git і засоби рефакторингу, навігації по коду, автодоповнення типових конструкцій і контекстних підказок.
- Великим плюсом редактора є підтримка великої кількості мов, таких як C ++, C #, Python, PHP, JavaScript та інших. Підтримка Python 3.4 і вище, а також Python 2.7.

Працювати з Python в Visual Studio Code з використанням розширення Microsoft Python зручно і продуктивно. Розширення робить VS Code відмінним редактором Python і працює в будь-якій операційній системі з різними інтерпретаторами Python. Воно використовує всю міць VS Code для забезпечення автозаповнення і IntelliSense, літінга, налагодження та модульного тестування, а також можливості легко перемикатися між середовищами Python, включаючи віртуальні середовища і середовища conda.

3.2.3 HTML, CSS, JavaScript

HTML

HTML — основа кожної веб-сторінки, незалежно від складності сайту або кількості задіяних технологій.

HTML відображає мову розмітки гіпертексту. «Мова розмітки» означає, що HTML використовує теги для ідентифікації різних типів контенту і цілей, які кожен переслідує на веб-сторінці. Для розмітки використовуються HTML-теги, також відомі як «елементи».

Кожна веб-сторінка складається з декількох тегів HTML, що позначають певний тип контенту на сторінці. Кожен тип вмісту на сторінці вважається «обгорнутим», тобто оточений тегами.

Використовуючи HTML, можливо додавати заголовки, формувати абзаци, розриви рядків, створювати списки, виділяти текст, створювати спеціальні символи, вставляти зображення, створювати посилання, створювати таблиці, управляти деякими стилями і багато іншого.

CSS

CSS — це каскадні таблиці стилів. Ця мова розмітки визначає, як HTML-елементи веб-сайту повинні відобразитися на інтерфейсі сторінки.

У той час як HTML є основною структурою сайту, CSS — це те, що дає всьому сайту стиль. Кольори, шрифти і фонові зображення — все це заслуга CSS. CSS також дозволяє веб-сайтам адаптуватися до різних розмірів екрану і типів пристроїв.

JavaScript

JavaScript є більш складною мовою, ніж HTML або CSS. В даний час JavaScript підтримується всіма сучасними веб-браузерами і використовується практично на кожному сайті в Інтернеті для більш потужних і складних функцій.

JavaScript — це логічна мова програмування, який можна використовувати для зміни вмісту веб-сайту і змусити його поводитися по-різному у відповідь на дії користувача. Загальне використання JavaScript включає в себе вікна підтвердження, заклики до дії і додавання нових ідентифікаторів до існуючої інформації.

JavaScript є об'єктно-орієнтованою мовою, але що використовується в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними клас-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, - функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання - що додає мові додаткову гнучкість.

Незважаючи на схожий з Сі синтаксис, JavaScript в порівнянні з мовою Сі має корінні відмінності:

- об'єкти з можливістю інтроспекції;

- функції як об'єкти першого класу;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

3.2.4 Фреймворк Django

Django — фреймворк для веб-додатків на мові Python. Один з основних принципів фреймворку - DRY (do not repeat yourself). Веб-системи на Django будуються з одного або декількох додатків, які рекомендується робити відчужуваними і підключаючими. Це одне з помітних архітектурних відмінностей цього фреймворка від деяких інших (наприклад, Ruby on Rails). Також, на відміну від багатьох інших фреймворків, обробники URL в Django конфігуруються явно (за допомогою регулярних виразів), а не автоматично задаються зі структури контролерів.

Django проектувався для роботи під управлінням Apache (з модулем `mod_python`) і з використанням PostgreSQL в якості бази даних. В даний час, крім PostgreSQL, Django може працювати з іншими СУБД: MySQL (MariaDB), SQLite, Microsoft SQL Server, DB2, Firebird, SQL Anywhere і Oracle. Для роботи з базою даних Django використовує власний ORM, в якому модель даних описується класами Python, і по ній генерується схема бази даних.

Архітектура Django схожа на «Модель-Представлення-Контролер» (MVC). Контролер класичної моделі MVC приблизно відповідає рівню, який в Django називається Представлення (View), а презентаційна логіка Представлення реалізується в Django рівнем Шаблонів (Templates). Через це рівневу архітектуру Django часто називають «Модель-Шаблон-Представлення» (MTV).

Спочатку розробка Django велася для забезпечення більш зручної роботи з новинними ресурсами, що досить сильно відбилося на архітектурі: фреймворк надає ряд засобів, які допомагають у швидкій розробці веб-сайтів

інформаційного характеру. Наприклад, розробнику не потрібно створювати контролери та сторінки для адміністративної частини сайту, в Django є вбудований додаток для керування вмістом, яке можна включити в будь-який сайт, зроблений на Django, і який може управляти відразу декількома сайтами на одному сервері.

Веб-фреймворк Django використовується в таких великих і відомих сайтах, як Instagram, Disqus, Mozilla, The Washington Times, Pinterest, lamoda і ін.

Деякі можливості Django:

- ORM, API доступу до БД з підтримкою транзакцій.
- Вбудований інтерфейс адміністратора, з уже наявними перекладами багатьма мовами.
- Диспетчер URL на основі регулярних виразів.
- Розширювана система шаблонів з тегами і спадкуванням.
- Система кешування.
- Інтернаціоналізація.
- Авторизація та аутентифікація, підключення зовнішніх модулів аутентифікації: LDAP, OpenID та інші.
- Система фільтрів («middleware») для побудови додаткових обробників запитів, як наприклад включені в дистрибутив фільтри для кешування, стиснення, нормалізації URL і підтримки анонімних сесій.
- Бібліотека для роботи з формами (успадкування, побудова форм по існуючій моделі БД).
- Вбудована автоматична документація по тегам шаблонів і моделей даних, доступна через адміністративне додаток.

Деякі компоненти фреймворка між собою пов'язані слабо, тому їх можна досить просто замінювати на аналогічні. Але з деякими (наприклад, з ORM) це зробити не дуже просто. Крім можливостей, вбудованих в ядро фреймворка, існують пакети, що розширюють його можливості.

На базі Django розроблено досить багато готових рішень, які розповсюджуються під вільною ліцензією, серед яких системи для управління інтернет-магазинами, універсальні системи управління вмістом, а також більш цілеспрямовані проекти.

В реалізованій системі Django використовувався яка засіб для побудови серверу.

Django MVC — MVT Pattern

Фреймворк Django реалізує архітектурний патерн Model View Template або скорочено MVT, який за фактом є модифікацією популярного в веб-програмуванні паттерна MVC (Model View Controller) [35].

MVC (Model View Controller) — це шаблон проектування програмного забезпечення для розробки веб-застосунків. Шаблон контролера представлення моделі складається з наступних трьох частин:

- Модель — найнижчий рівень шаблону, який відповідає за підтримання даних.
- Представлення — відповідає за відображення всіх або частини даних для користувача.
- Контролер — програмний код, який управляє взаємодією між моделлю і представленням.

Модель-представлення-шаблон (MVT) трохи відрізняється від MVC. Насправді основна відмінність між цими двома шаблонами полягає в тому, що Django сам піклується про частини контролера (програмний код, який контролює взаємодію між моделлю і представленням), залишаючи розробника з шаблоном. Шаблон являє собою файл HTML, змішаний з мовою шаблонів Django (DTL).

На рисунку 25 проілюстровано, як кожен з компонентів шаблону MVT, що взаємодіє один з одним для обслуговування користувальницького запиту.

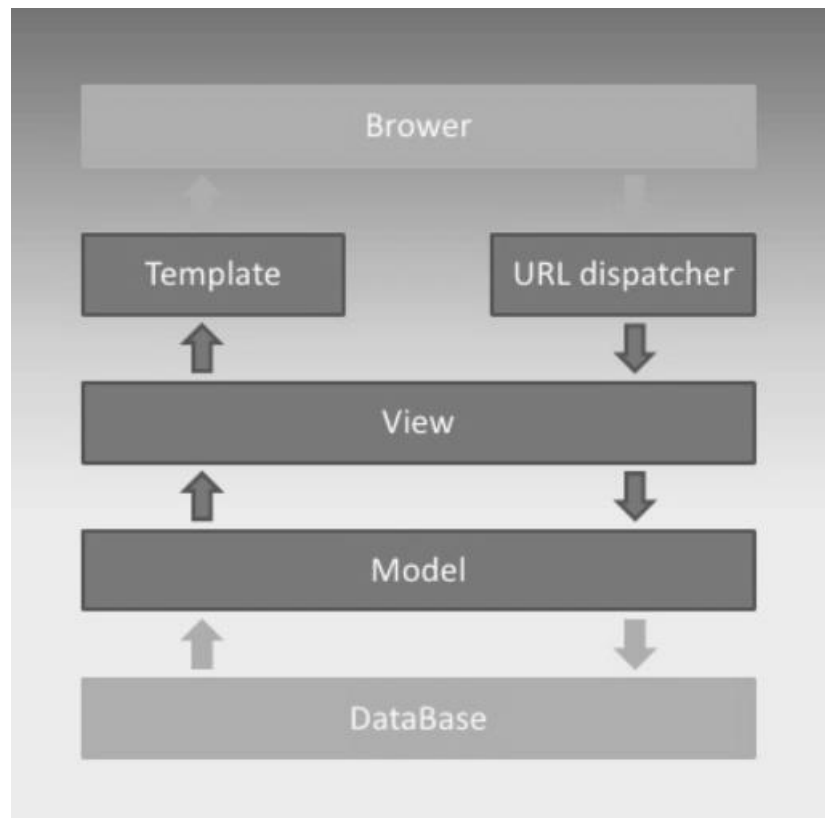


Рис. 25 Архітектура MVT в Django

Основні елементи патерну:

- URL dispatcher — при отримання запиту на підставі запитаної адреси URL визначає, який ресурс повинен обробляти даний запит.
- View — отримує запит, обробляє його і відправляє у відповідь користувачеві деяку відповідь. Якщо для обробки запиту необхідно звернення до моделі і бази даних, то View взаємодіє з ними.
- Model — описує дані, які використовуються в додатку. Окремі класи, що як правило, відповідають таблицями в базі даних.
- Template — представляє логіку представлення у вигляді згенерованої розмітки html.

Коли до додатка приходять запит, то URL dispatcher визначає, з яким ресурсом зіставляється даний запит і передає цей запит обраному ресурсу. Ресурс фактично являє функцію або View, який отримує запит і певним чином обробляє його. В процесі обробки View може звертатися до моделей і бази даних, отримувати з неї дані, або, навпаки, зберігати в неї дані [35]. Ре-

зультат обробки запиту відправляється назад, і цей результат користувач бачить в своєму браузері. Як правило, результат обробки запиту представляє згенерований html-код, для генерації якого застосовуються шаблони (Template).

3.2.5 Протокол WebSockets

Веб-сокети (WebSockets) — це передова технологія, яка дозволяє створювати інтерактивне з'єднання між клієнтом (браузером) та сервером для обміну повідомленнями в режимі реального часу. Основні особливості веб-сокетів наступні:

- Протокол веб-сокета стандартизований, що означає, що за допомогою цього протоколу можливий зв'язок між веб-серверами і клієнтами в режимі реального часу.
- Веб-сокети трансформуються в багатоплатформовий стандарт для обміну даними між клієнтом і сервером в режимі реального часу.
- Найбільша перевага WebSocket - це двосторонній зв'язок (повний дуплекс) по одному TCP-з'єднанню.

WebSocket — це незалежний протокол на основі TCP, але він призначений для підтримки будь-якого іншого протоколу, який традиційно працює тільки поверх чистого TCP-з'єднання. На рисунку 26 відображається процес встановлення зв'язку за допомогою протоколу Web Socket.

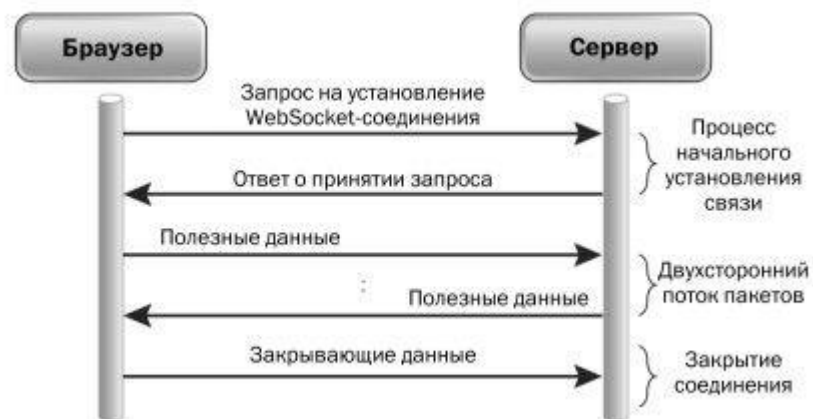


Рис. 26 Процес встановлення зв'язку по протоколу WebSocket

Це протокол транспортного рівня, поверх якого може працювати будь-який інший протокол. Web Socket API підтримує можливість визначення суб-протоколів: бібліотек протоколів, які можуть інтерпретувати певні протоколи. Приклади таких протоколів включають XMPP, STOMP і AMQP. Розробникам більше не потрібно думати з точки зору парадигми HTTP запит-відповідь. На стороні сервера промисловим стандартом є використання існуючих бібліотек протоколів, що працюють поверх TCP, і використання шлюзу Web Socket.

Остання специфікація протоколу Web Socket визначається як RFC 6455 — пропонується стандарт. RFC 6455 підтримується різними браузерами, такими як Internet Explorer, Mozilla Firefox, Google Chrome, Safari і Opera.

3.2.6 Бібліотека Django Channels

Channels — це проект, який використовує Django і розширює його можливості за межі HTTP - для обробки WebSockets, протоколів чату, IoT-протоколів і багато чого іншого. Він побудований на специфікації асинхронного сервера Python під назвою ASGI. Channels готує Django для створення асинхронного коду через синхронне ядро Django, що дозволяє проектам Django обробляти не тільки HTTP, але і протоколи, які також вимагають тривалих з'єднань [36].

Канали дають інструменти для написання основних споживачів (consumers) — окремих частин, які можуть обробляти обмін повідомленнями в чаті або повідомлення — і пов'язувати їх разом з маршрутизацією URL, виявленням протоколу і іншими зручними речами для створення повноцінного додатка.

Channels і ASGI поділяють вхідні з'єднання на два компоненти: scope (область) і серія events (події).

Scope — це набір відомостей про одне вхідне з'єднання - наприклад, шлях, з якого був зроблений веб-запит, або вихідний IP-адреса WebSocket'у - і яке зберігається протягом усього з'єднання.

Для HTTP область дії триває всього один запит. Для WebSocket вона триває протягом терміну роботи сокета (але змінюється, якщо сокет закривається і повторно підключається). Для інших протоколів це залежить від того, як написана специфікація протоколу ASGI [36]. Схема роботи технології Django Channels продемонстрована на рисунку 26.

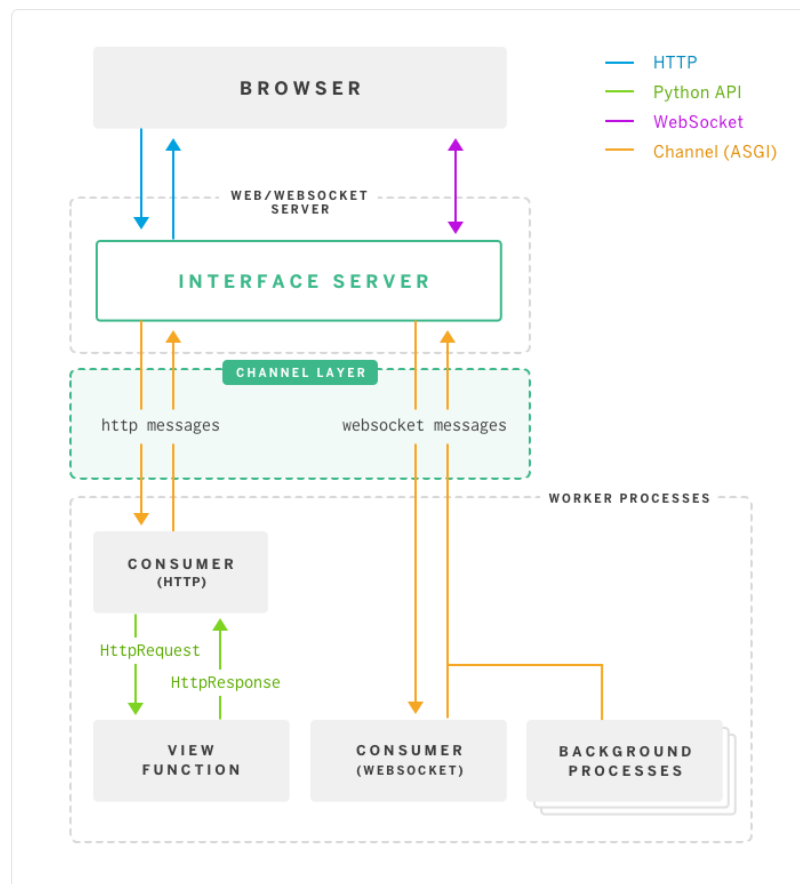


Рис. 26 Схема роботи Django з використанням каналів

Під час існування цієї області дії відбувається серія подій. Вони представляють призначені для користувача взаємодії - наприклад, створення HTTP-запиту або відправка фрейма WebSocket. Канали або програми ASGI будуть створені один раз для кожної області, а потім будуть передані потоки подій, що відбуваються в цій галузі, щоб вирішити, що робити з ними.

Мета Channels — дозволити створювати свої проекти Django для роботи з будь-яким протоколом або транспортом, з якими можливо зіткнутися в сучасній мережі, і в той же час дозволяючи працювати зі знайомими компонентами і стилем кодування.

3.2.7 Бібліотека OpenCV

OpenCV — бібліотека комп'ютерного зору і машинного навчання з відкритим вихідним кодом. У неї входять понад 2500 алгоритмів, в яких є як класичні, так і сучасні алгоритми для комп'ютерного зору і машинного навчання. Ця бібліотека має інтерфейси на різних мовах, серед яких є Python, Java, C++ і Matlab.

OpenCV — це набір типів даних, функцій і класів для обробки зображень алгоритмами комп'ютерного зору.

Основні модулі бібліотеки [37]:

core — ядро (Містить базові структури даних і алгоритми):

- базові операції над багатовимірними числовими масивами.
- матрична алгебра, математичні ф-ції, генератори випадкових чисел.

- Запис / відновлення структур даних в / з XML.

- базові функції 2D графіки.

CV — модуль обробки зображень і комп'ютерного зору:

- базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів і т. д.).

- аналіз зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми).

- аналіз руху, спостереження за об'єктами.

- виявлення об'єктів, зокрема осіб.

- калібрування камер, елементи відновлення просторової структури.

Класифікатор Haar Cascade

Класифікатор Haar Cascade (Haar Cascade classifier) — це ефективний підхід до виявлення об'єктів, який був запропонований Полом Віолою і Майклом Джонсом в їхній статті «Rapid Object Detection using a Boosted Cascade of Simple Features» в 2001 році [37].

Класифікатор заснований на методі Вейвлету Хаара для аналізу пікселів зображення на квадрати за функціями. При цьому використовуються концепції «цілісного зображення» для обчислення виявлених «особливостей».

Це, в основному, підхід, заснований на машинному навчанні, при якому каскадна функція навчається на великій кількості зображень, як позитивних, так і негативних. Потім на основі навчання класифікатор використовується для виявлення обличчя на інших зображеннях. Класифікатор представляє собою окремий файл .xml з безліччю наборів функцій, який відповідає дуже специфічному типу використання.

Лістинг 1 Імпортування необхідних бібліотек і підключення класифікатора для виділення обличчя (haarcascade_frontalface_default.xml)

```
import cv2
import numpy as np
face_classifier =
cv2.CascadeClassifier('/haarcascade_frontalface_default.xml
')
```

Для подальшого розпізнавання емоцій, отримане обличчя повинно бути чорно-білим. Коли OpenCV зчитує зображення RGB, вона зазвичай зберігає зображення в каналі BGR (синій, зелений, червоний). Для подальшої роботи потрібно перетворити цей канал BGR в сірий канал. Причина цього в тому, що сірий канал легко обробляти і цей процес вимагає менших обчислювальних витрат, оскільки містить тільки 1 канал чорно-білого.

Лістинг 2 Отримання чорно-білого зображення з кольорового

```
gray = cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)
```

```
faces = face_classifier.detectMultiScale(gray,
1.0485258, 6)
```

3.2.8 Фреймворк TensorFlow та Keras API

Еволюція TensorFlow в платформу глибокого навчання відбулася не відразу. Спочатку TensorFlow позиціонувалась як символічна математична бібліотека для програмування потоків даних для вирішування ряду завдань. Отже, рішення, яке спочатку пропонувало TensorFlow, не було повноцінною бібліотекою машинного навчання. Мета полягала в тому, щоб створити ефективну математичну бібліотеку для того, щоб алгоритми машинного навчання, побудовані на основі цієї ефективної структури, могли навчатися за короткий час з високою точністю.

Однак багаторазова побудова моделей з нуля за допомогою низькорівневих API була не дуже ідеальна. Отже, Франсуа Шоле, інженер Google, розробив Keras як окрему бібліотеку глибокого навчання високого рівня. Хоча Keras може працювати поверх різних бібліотек, таких як TensorFlow, Microsoft Cognitive Toolkit, Theano або PlaidML [33]. TensorFlow був і залишається найпоширенішою бібліотекою, з якої використовується Keras.

Побачивши проблеми в процесі побудови моделі, команда TensorFlow оголосила, що Keras стане центральним високорівневим API, що використовується для побудови і навчання моделей в TensorFlow 2.0. Альтернативний високорівневий API, API-інтерфейс Estimator, почав втрачати свою і без того зменшувану популярність після цього оголошення.

Є багато різних методів, за допомогою яких створюються моделі за допомогою TensorFlow. Основна причина цього — нездатність TensorFlow прийняти єдину модель API.

У версії 1.x для проектів виробничого рівня основним API для побудови моделей був API-інтерфейс Estimator. Але з недавніми змінами Keras API майже наздогнав API Estimator. Спочатку API-інтерфейс Estimator був більш масштабованим, допускав множинне розповсюдження і мав зручну кросплатформенну функціональність. Проте, більшість цих переваг API-інтерфейсу

Estimator в даний час усунуто, і тому незабаром API-інтерфейс Keras, ймовірно, стане єдиним стандартним API-інтерфейсом для побудови моделей TensorFlow.

Існує три методи побудови моделі Keras в TensorFlow:

- **Sequential API.** Послідовний API — кращий метод, коли для побудовання простої моделі з одним входом, виходом і гілкою шару. Це відмінний варіант для новачків, які хочуть швидко вчитися.
- **Functional API.** Функціональний API — найпопулярніший метод побудови моделей Keras. Він може робити все, що може робити Sequential API. Крім того, він дозволяє використовувати кілька входів, кілька виходів, розгалуження і спільне використання шарів.
- **Створення підкласів моделей.** Створення підкласів моделей призначене для розробників просунутого рівня, яким необхідний повний контроль над своєю моделлю, шаром і процесом навчання.

3.3 Модулі програмної системи

3.3.1 Клієнтська частина системи

Клієнтською частиною розробленої системи виступає веб-застосунок. Веб-застосунок створений з використання HTML, CSS та JavaScript.

Веб-застосунок отримує доступ до використання веб-камери користувача для того, щоб транслювати відео.

Лістинг 3 Отримання доступу на використання веб-камери

```
video = document.getElementById('video');
navigator.mediaDevices.getUserMedia({video: true,
audio: false})
.then(function(stream) {
    video.srcObject = stream;
    video.play();
    console.log('START CAMERA PLAY')
```



```

    })
    .catch(function(err) {
        console.log("An error occurred: " + err);
    });

```

Після отримання доступу відбувається передача кадрів з відео кожному секунду. Серверна частина отримує кадри для того, щоб аналізувати емоції користувача.

Лістинг 4 Отримання кадру з відео кожному секунду

```

function periodicGetFrame() {
    let timerId = setInterval(() => getFrame(),
1000);
}

function getFrame() {
    var context = canvas.getContext('2d');
    if (width && height) {
        canvas.width = width;
        canvas.height = height;
        context.drawImage(video, 0, 0, width, height);
        var data = canvas.toDataURL('image/png');

```

Передача кадрів відбувається за допомогою протоколу WebSocket. Спочатку ініціалізується з'єднання, далі відправляється отриманий кадр, який закодовано методом кодування Base64.

Лістинг 5 Встановлення WebSocket з'єднання та відправка кадру

```

function startup() {
    console.log('STARTUP PAGE')
    socket = new WebSocket (
        'ws://'
        + '127.0.0.1:8000'
        + '/ws/frames/'

```

```

    )
    socket.send(JSON.stringify(
        {
            'imageb64':data
        }
    ))

```

Також за допомогою веб-сокетів відбувається отримання даних аналізу щодо емоційного стану користувача, які відправлені сервером.

Лістинг 6 Отримання даних, відправлених сервером

```

socket.onmessage = function (e) {
    const data = JSON.parse(e.data)
    console.log(data)
    if(data.predicts != "EMPTY") {
        updateChart(data.predicts)
        updateChartStats(data.predicts)
        analyzeEmotion(data.predicts)
    }
}

```

Метод кодування Base64

Base64 - спосіб кодування довільних двійкових даних в ASCII текст. За своєю суттю кодування дуже просте. Кожні шість біт на вході кодується в один із символів 64-літерного алфавіту. "Стандартний" алфавіт, який для цього використовується - це A-Z, a-z, 0-9, +, / і = в якості заповнюваного символу в кінці. Таким чином, на кожні 3 байта даних доводиться 4 символу.

Так історично склалося, що багато форматів передачі і зберігання даних використовують текст замість бінарних кодів (html, url схеми, xml, email). Але що, якщо формат передачі даних текстовий, а передати необхідно

бінарні дані (окремо або разом з текстовими даними). Ось тут на допомогу і приходить base64.

Типове застосування в веб розробці

- data: URL схеми для зображень (css, html).
- Отримання base64 уявлення бінарних даних canvas бітмапами.
- Передача картинок і інших даних в XML (не використовуючи зовнішні файли).
- Зберігання зображень в базі.
- Включення зображень в email.

data: URL і base64 data: URL - це певна стандартом RFC 2397 схема, яка дозволяє включати невеликі елементи даних в рядок URL, як якщо б вони були посиланням на зовнішній ресурс.

3.3.2 Серверна частина системи

Серверна частина системи розроблена за допомогою фреймворку Django.

Робота серверу полягає в отриманні кадру з веб-застосунку, передача його модулю розпізнавання емоцій, отримання даних розпізнавання та відправлення їх веб-застосунку.

Серверна частина також працює з протоколом WebSocket, роботу з яким надає технологія Django Channels.

Лістинг 7 Встановлення WebSocket з'єднання, отримання кадру та аналіз емоційного стану користувача

```
websocket_urlpatterns = [
    re_path(r'ws/frames/',
consumers.FramesConsumer.as_asgi()),
]
class FramesConsumer(WebSocketConsumer):
```

```

def connect(self):
    self.accept()
    print ('Websocket connected')
def disconnect(self, close_code):
    pass
def receive(self, text_data):
    datas = json.loads(text_data)
    imageBase64 = datas["imageb64"]
    imageBase64 = imageBase64.partition(",")[2]
    print(len(str(imageBase64)))
    grayCv2Img = string64ToGrayScale(imageBase64)
    predicts =
videoCameraMain.get_data_by_frame(grayCv2Img)
    print(predicts)
    jsonPredicts = {'happy': 0, 'sad':0,
'surprise':0, 'angry':0, 'neutral':0}
    predictsPresented = False
    if(len(predicts) > 0):
        predictsPresented = True
        jsonPredicts['happy'] = str(predicts[1])
        jsonPredicts['sad'] = str(predicts[3])
        jsonPredicts['surprise'] = str(predicts[4])
        jsonPredicts['angry'] = str(predicts[0])
        jsonPredicts['neutral'] = str(predicts[2])

    print(jsonPredicts)
    jsonResponse = {'predicts' : ''}
    if(predictsPresented):
        jsonResponse['predicts'] = jsonPredicts
    else:
        jsonResponse['predicts'] = 'EMPTY'

    self.send(text_data=json.dumps(jsonResponse))

```

3.3.3 Архітектура нейронної мережі для розпізнавання емоцій людини

Побудована нейронна мережа являє собою згорткову нейронну мережу. На вхід подається зображення обличчя у відтинках сірого, розміром 48 * 48 пікселів. Завдання нейронної мережі полягає в класифікації кожного обличчя на основі емоцій, що відображаються у виразі обличчя, до однієї із п'яти категорій (радість, сум, злість, подив, нейтральний стан). Побудована нейронна мережа являється модифікацією нейронної мережі VGG-16.

Шари, додані до згорткової нейронної мережі:

- Згортковий шар (Convolution layer);
- Шар пулінга (Pooling layer);
- Пакетна нормалізація (Batch normalization);
- Шар активації (Activation Layer);
- Шар дропаут (Dropout Layer);
- Шар вирівнювання (Flatten Layer);
- Шар щільності (Dense Layer).

Нейронна мережа складається з шести шарів Conv2D, трьох шарів MaxPooling2D після кожної другої згортки та трьох повнозв'язних шарів.

В таблиці 2 показані шари побудованої згорткової мережі. Рівні активації і нормалізації пакетів не показані в таблиці, вони повинні бути після кожного рівня CONV і FC (CONV => Activation => BN і FC => Activation => BN).

Таблиця 2

Архітектура побудованої нейронної мережі

Layer Type	Output Size	Filter Size / Stride
Input Image	48 x 48 x 1	3 x 3, K = 32
CONV	48 x 48 x 32	3 x 3, K = 32
CONV	48 x 48 x 32	3 x 3, K = 32
POOL	24 x 24 x 32	2 x 2
Dropout	24 x 24 x 32	
CONV	24 x 24 x 64	3 x 3, K = 64
CONV	24 x 24 x 64	3 x 3, K = 64
POOL	12 x 12 x 64	2 x 2
Dropout	12 x 12 x 64	
CONV	12 x 12 x 128	3 x 3, K = 128
CONV	12 x 12 x 128	3 x 3, K = 128
POOL	6 x 6 x 128	2 x 2
Dropout	6 x 6 x 128	
FC	64	
Dropout	64	
FC	64	
FC	6	
softmax	5	

Згортання (Convolution) — це лінійна операція, яка передбачає множення набору ваг на вхід, подібно до традиційної нейронної мережі. Враховуючи, що методика була розроблена для двовимірного введення, множення

виконується між масивом вхідних даних та двовимірним масивом ваг, званим фільтром або ядром.

Пакетна нормалізація (Batch normalization) — метод, який дозволяє підвищити продуктивність і стабілізувати роботу нейронних мереж. Суть даного методу полягає в тому, що деяким шарам нейронної мережі на вхід подаються попередньо оброблені дані, що мають нульове математичне сподівання і одиничну дисперсію [35].

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.

Одним з найважливіших аспектів глибокої нейронної мережі є функція активації (Activation function), яка привносить в мережу нелінійність. Побудована нейронна мережа використовує функцію активації ELU, також відома як Експоненціальна лінійна одиниця. Ця функція активації, яка дещо схожа на ReLU з деякими відмінностями. Подібно до інших ненасичуючих функцій активації, ELU не страждає від проблеми зникнення градієнтів та вибухаючих градієнтів [34]. І подібно до Leaky-ReLU та PReLU, і на відміну від ReLU, ELU не страждає від проблем відмираючих нейронів. Використання ELU призводить до зменшення часу навчання та підвищення точності в нейронних мережах порівняно з ReLU та його варіантами.

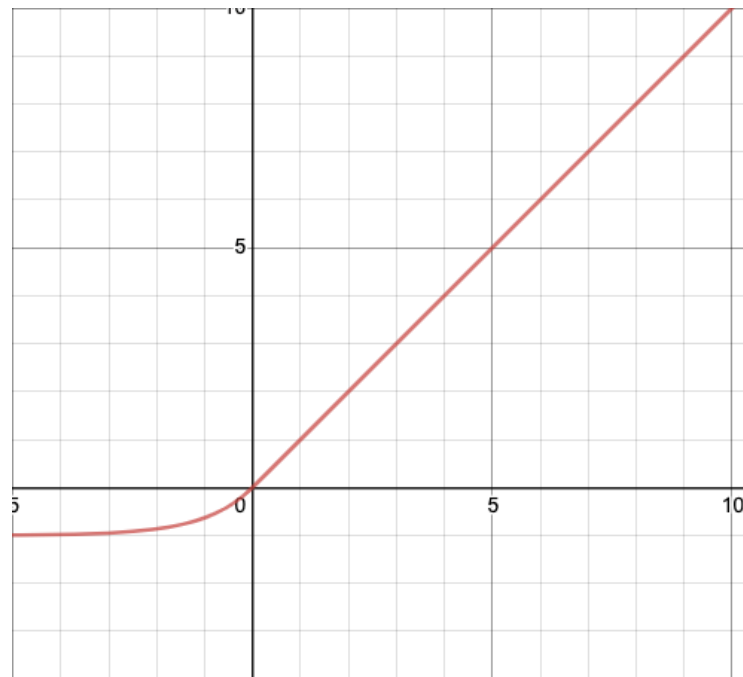


Рис. 27 Експоненціальна лінійна одиниця або функція активації *ELU*

Шар Dropout застосовується в нейронних мережах для вирішення проблеми перенавчання. Для цього він випадковим чином вибирає частки одиниць і при кожному оновленні призначає їм значення 0.

Шар вирівнювання (Flatten Layer) використовується для конвертації вхідних даних в меншу розмірність. Наприклад, вхідний шар розмірності (batch_size, 3,2) «вирівнюється» для виведення розмірності (batch_size, 6).

Останнім шаром нейронної мережі є функція активації Softmax. Softmax перетворює логіти (числові вихідні дані останнього лінійного шару нейронної мережі з мультикласовою класифікацією) в ймовірності, беручи експоненти кожного виведення, а потім нормалізує кожне число на суму цих показників, щоб весь вихідний вектор становив одиницю — всі ймовірності повинні в сумі дорівнювати одиниці. Втрати перехресної ентропії зазвичай є функцією втрат для такого завдання класифікації декількох класів. Softmax часто додається до останнього рівня мережі класифікації зображень.

Лістинг 8 Створення моделі нейронної мережі

```
model = Sequential()
# Block-1
```



```

    model.add(Conv2D(32, (3, 3), padding='same', kernel_initializer='he_normal', input_shape=(img_rows, img_cols, 1)))
    model.add(Activation('elu'))
    model.add(BatchNormalization())
    model.add(Conv2D(32, (3, 3), padding='same', kernel_initializer='he_normal', input_shape=(img_rows, img_cols, 1)))
    model.add(Activation('elu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.2))
    # Block-2
    model.add(Conv2D(64, (3, 3), padding='same', kernel_initializer='he_normal'))
    model.add(Activation('elu'))
    model.add(BatchNormalization())
    model.add(Conv2D(64, (3, 3), padding='same', kernel_initializer='he_normal'))
    model.add(Activation('elu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.2))
    # Block-3
    model.add(Conv2D(128, (3, 3), padding='same', kernel_initializer='he_normal'))
    model.add(Activation('elu'))
    model.add(BatchNormalization())
    model.add(Conv2D(128, (3, 3), padding='same', kernel_initializer='he_normal'))
    model.add(Activation('elu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.2))
    # Block-4
    model.add(Flatten())
    model.add(Dense(64, kernel_initializer='he_normal'))
    model.add(Activation('elu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    # Block-5
    model.add(Dense(64, kernel_initializer='he_normal'))
    model.add(Activation('elu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    # Block-6
    model.add(Dense(num_classes, kernel_initializer='he_normal'))
    model.add(Activation('softmax'))

```

3.4 Проект інтерфейсу

Розроблений веб-застосунок дозволяє користувачу аналізувати свій емоційний стан в режимі реального часу. Користувачу необхідно надати доступ до своєї веб-камери. Після надання доступу, користувач може бачити відео з веб-камери та діаграми з результатом аналізу емоцій. Сторінка веб-застосунку відображає дві діаграми: гістограму та кругову діаграму.

Гістограма транслює поточну емоцію користувача. Кругова діаграма відображає всі емоції, котрі були проявлені користувачем в період користування застосунком.

Емоції, що можуть бути проаналізовані розробленою системою: радість, подив, сум, злість та нейтральний стан. Для точності розпізнавання емоцій користувач має зафіксувати певну емоцію на протязі однієї секунди.

На рисунках 28 и 29 проілюстровано інтерфейс розробленого веб-застосунку та приклад розпізнавання двох емоцій: радості та подиву.

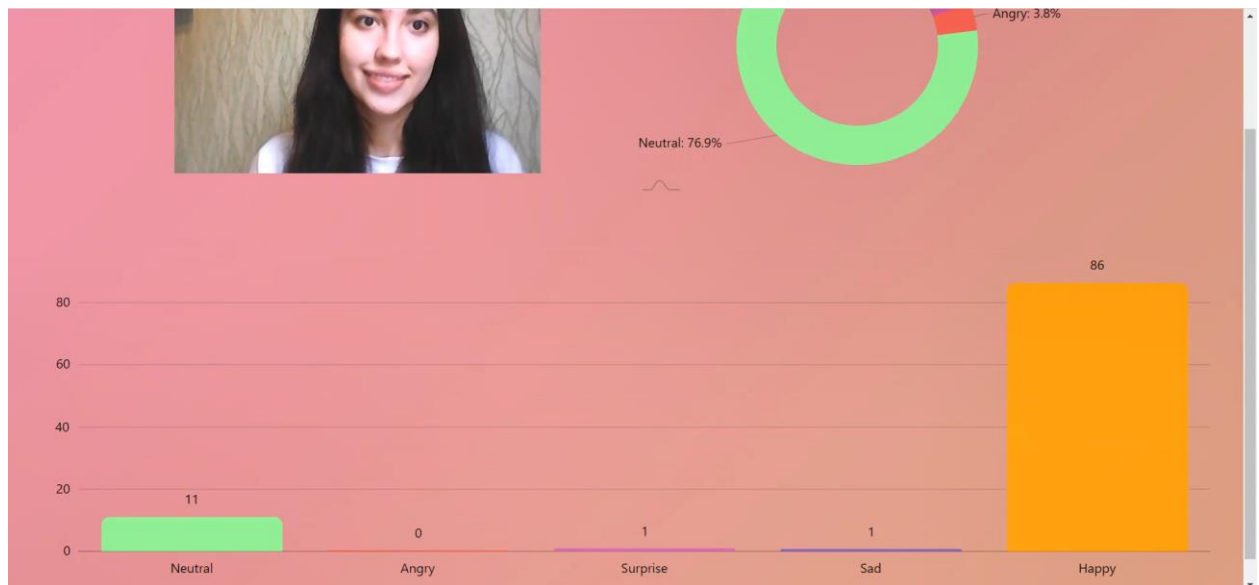


Рисунок 28 Інтерфейс розробленої системи. Розпізнавання емоцій радості

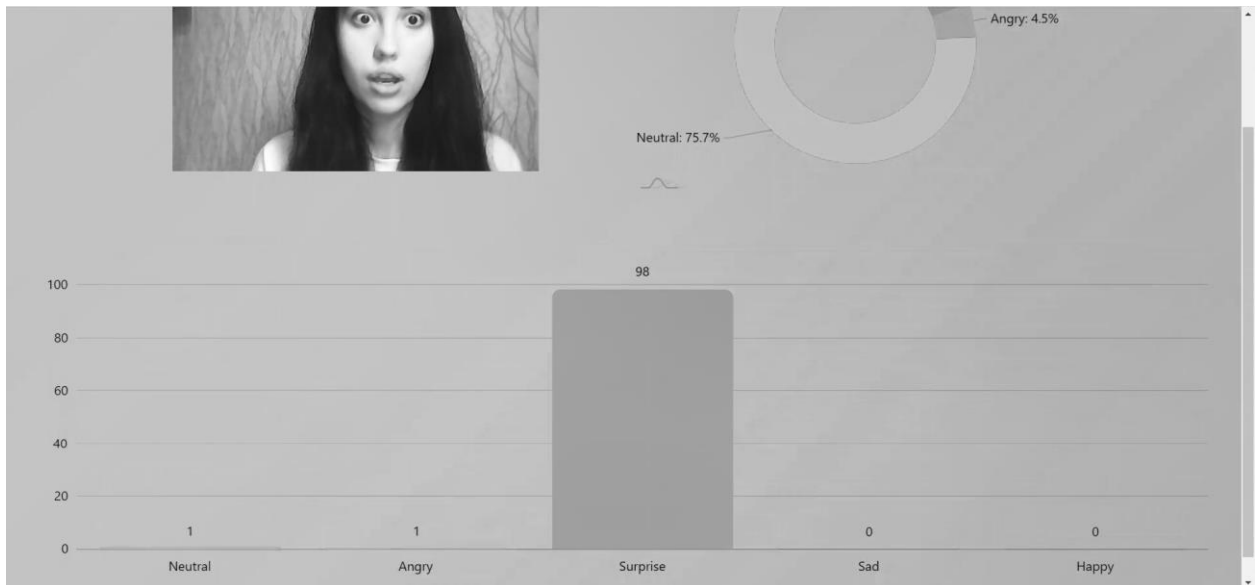


Рисунок 29 Інтерфейс розробленої системи. Розпізнавання емоції подиву

3.5 Вимоги до апаратного забезпечення

Основною задачею створеної системи є розпізнавання емоцій користувача з веб-камери в режимі реального часу.

Мінімальні вимоги до апаратного забезпечення:

- Одноядерний процесор з частотою 2.0 GHz.
- Оперативна пам'ять ємністю не менше 512 Мб.
- Монітор з розподільною здатністю 1024x768.
- Наявність Web – камери 0.5 Мп.

Рекомендовані вимоги до апаратного забезпечення:

- Двоядерний процесор з частотою 2.0 GHz.
- Монітор з розподільною здатністю 1024x768 і більше.
- Наявність Web – камери 1Мп і більше.

Розроблена система також складається з серверної частини. Тому були визначені такі вимоги до сервера.

Мінімальні вимоги до апаратного забезпечення серверу:

- Чотири-ядерний процесор з базовою тактовою частотою 3.5 GHz.
- Оперативна пам'ять ємністю не менше 8 Гб.

- Інтернет з'єднання 100 мбіт.

Рекомендовані вимоги до апаратного забезпечення серверу:

- Восьми-ядерний процесор з базовою тактовою частотою 3.6 GHz.
- Оперативна пам'ять ємністю не менше 16 Гб.
- Інтернет з'єднання 100 мбіт.

Мінімальні вимоги до програмного забезпечення клієнта

- ОС Windows , починаючи з Windows 7 до Windows 10

Вимоги до користувача:

- Від користувача вимагається базові навички роботи з інтернет браузерами.

3.6 Опис функціональних можливостей

Оскільки основною метою кваліфікаційної роботи є дослідження поставленого питання, існуючих алгоритмів та методів вирішення проблеми, а не розробка кінцевого програмного продукту, то функціональні можливості системи орієнтовані на зручність проведення дослідження, а не для використання у якості готової програмної системи для розпізнавання емоцій з обличчя людини. Було розроблено тестовий веб-застосунок, що дозволяє розпізнавати емоційний стан користувача з відео в режимі реального часу. Аналіз емоцій відбувається за допомогою розпізнавання кадру з відео, яке виконує розроблена згортова нейрона мережа. Результат аналізу надається користувачу у вигляді двох діаграм. Одна діаграма відображає поточну емоцію користувача, а інша діаграма відображає перелік всіх емоцій, що були помічені за період користування веб-застосунком.

РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖ- НОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ

4.1 Навчання та перевірка побудованої нейронної мережі

Датасет Facial Expression Recognition 2013 (FER2013) складається з двох наборів даних: навчальний (training) набір та валідаційний (validation) набір.

На рисунку 30 відображається графік точності та витрат для навчаючого набору та для валідаційного набору.

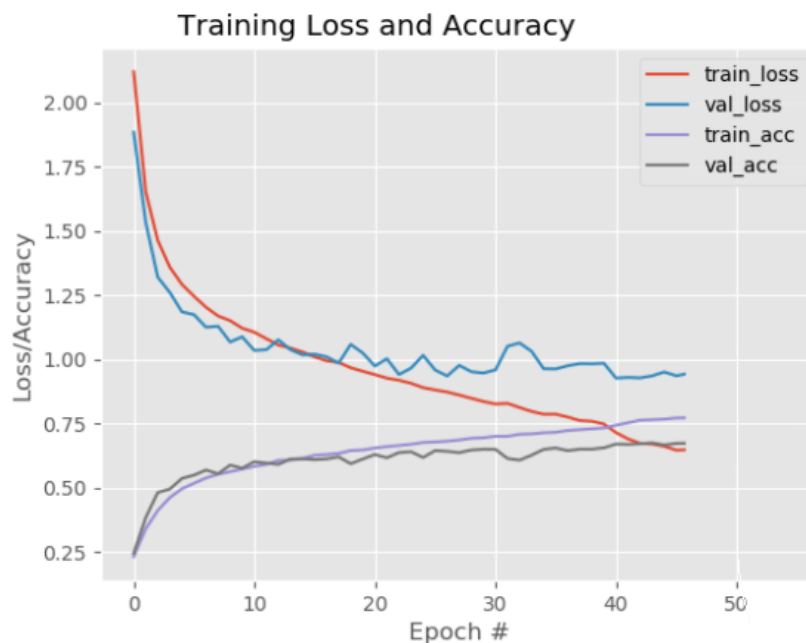


Рис 30 Графік точності та витрат для навчаючого набору та валідаційного наборів

Навчання було зупинено після 40 епох, оскільки на графіку видно, що відбулося перенавчання мережі, але валідаційна точність дорівнює 66.85%. Перенавчання — це явище, коли модель, що навчається, добре розпізнає приклади з навчальної множини, але при цьому не розпізнає або погано розпізнає будь-які інші приклади, які не брали участі в процесі навчання

4.2 Вимірювання точності розпізнавання емоцій людини в залежності від статі

Розроблена нейронна мережа дозволяє розпізнати 5 емоційних станів людини: радість, сум, подив, злість та нейтральний стан.

Для дослідження точності розпізнавання було проведено експеримент, в якому розпізнавались емоції людей жіночої і чоловічої статі. Це потрібно для того, щоб гарантувати правильність розпізнавання емоцій всіх людей, не залежно від їх статі.

Нейронна мережа розпізнавала різні зображення чоловіків і жінок, що проявляли різні емоції. Далі обчислювався відсоток невірних розпізнавань, який потім віднімався від загального відсотка числа вимірів. Також для проведення вимірювання було допущено, що за 1 секунду відбувається тільки одне розпізнавання.

Тобто оцінку ефективності програми можна визначити за допомогою формули:

$$p = \frac{I_{заг} - I_n}{I_{заг}}$$

Де $I_{заг}$ – загальна кількість вимірів, I_n – кількість невірних вимірів.

Результати дослідження представлені в таблицях 3 і 4 та на рисунках 31 і 32.

Таблиця 3

Результати розпізнавання емоцій жінок

	Радість	Злість	Сум	Подив	Нейтрально
Загальна кількість вимірів	15	15	15	15	15
Кількість невірних вимірів	1	8	4	0	0
Процент вірного розпізнавання	93%	60%	73%	100%	100%

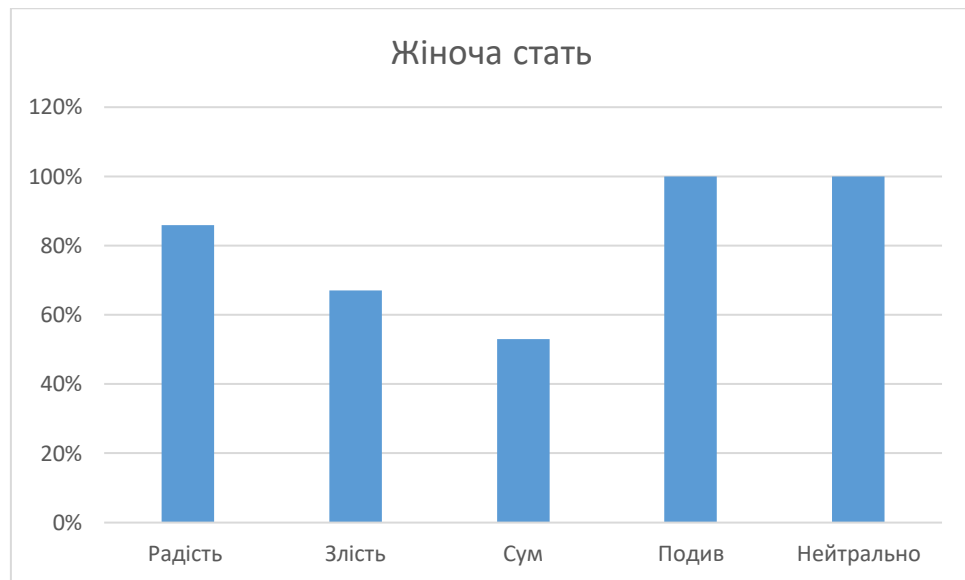


Рисунок 31 Діаграма з результатами розпізнавання емоцій жінок

Таблиця 4

Результати розпізнавання емоцій чоловіків

	Радість	Злість	Сум	Подив	Нейтрально
Загальна кількість вимірів	15	15	15	15	15
Кількість невірних вимірів	2	5	7	0	0
Процент вірного розпізнавання	86%	67%	53%	100%	100%

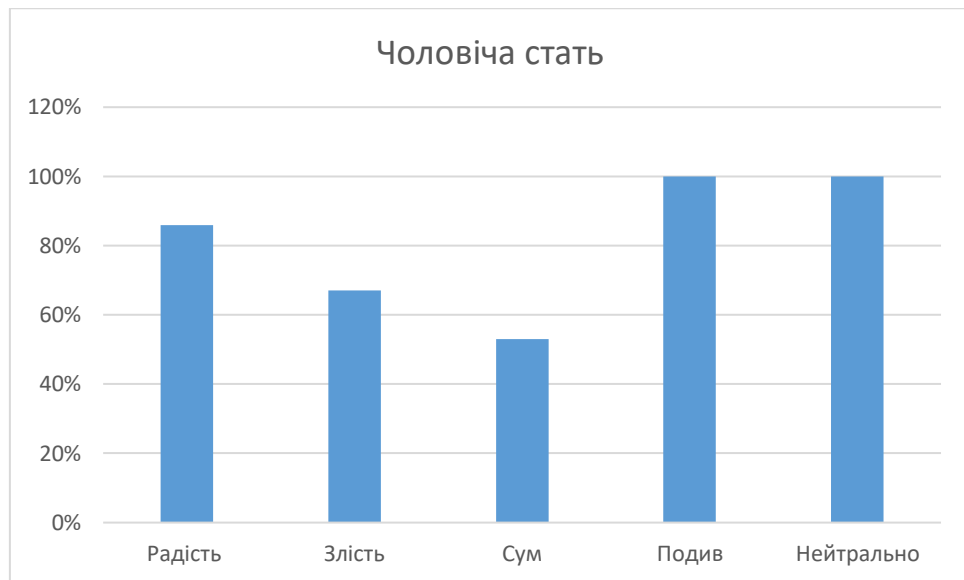


Рисунок 32 Діаграма з результатами розпізнавання емоцій чоловіків

З результатів дослідження можна зробити висновки, що система недосконало визначає дві емоції: сум та злість. Але у чоловіків визначення злості краще, ніж у жінок. А у випадку розпізнавання емоції суму у жінок результат розпізнавання краще, ніж у чоловіків. Емоції радості, подиву та нейтрального стану правильно визначаються незалежно від статі людини.

4.3 Вимірювання точності розпізнавання емоцій людини в залежності від віку

В рамках наступного дослідження було проведено дослідження точності розпізнавання емоцій людини в залежності від її віку. Це дослідження необхідне для того, щоб гарантувати, що система зможе розпізнати емоції як дитини, так і людини похилого віку.

Оцінка ефективності розпізнавання визначалась за той самою формулою, що і в попередньому дослідженні

$$p = \frac{I_{заг} - I_{н}}{I_{заг}}$$

Де $I_{заг}$ – загальна кількість вимірів, $I_{н}$ – кількість невірних вимірів.

Результати дослідження продемонстровані в таблицях 5 і 6 та на рисунках 33 і 34.

Таблиця 5

Результати розпізнавання емоцій дітей

	Радість	Злість	Сум	Подив	Нейтрально
Загальна кількість вимірів	15	15	15	15	15
Кількість невірних вимірів	1	3	5	0	2
Процент вірного розпізнавання	93%	80%	66%	100%	87%

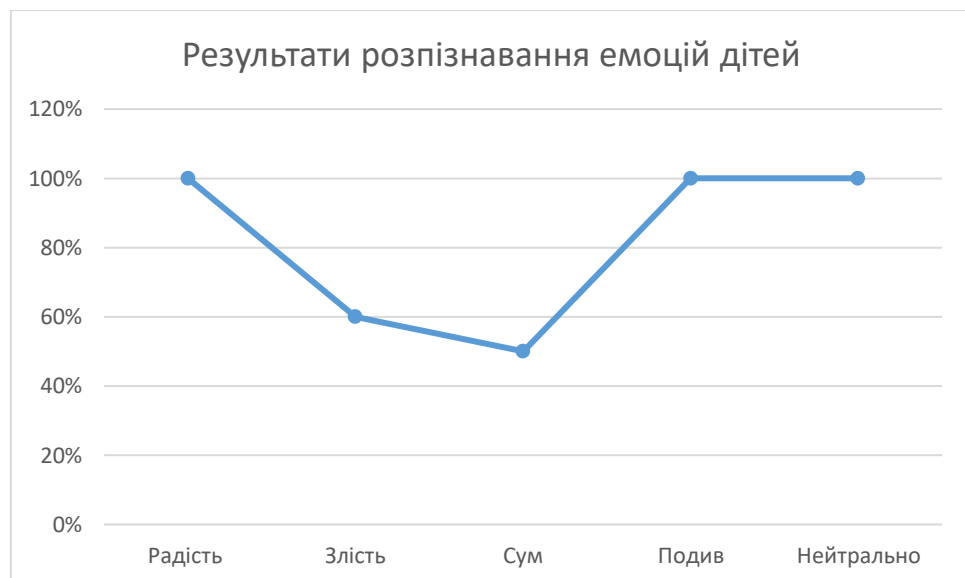


Рисунок 33 Діаграма з результатами розпізнавання емоцій дітей

Таблиця 6

Результат розпізнавання емоцій людей похилого віку

	Радість	Злість	Сум	Подив	Нейтрально
Загальна кількість вимірів	15	15	15	15	15
Кількість невірних вимірів	1	4	7	1	4
Процент вірного розпізнавання	93%	73%	53%	93%	73%



Рисунок 34 Діаграма з результатами розпізнавання емоцій людей похилого віку

За результатами проведеного дослідження можна побачити, що емоція сум не завжди може бути розпізнана у дітей і людей похилого віку. Також при проведенні цього дослідження помічено, що показник розпізнавання нейтрального стану гірше, ніж в попередньому тесті. Але показник розпізнавання емоцій злість краще у дітей. Це зумовлено тим, що діти більш чіткіше транслюють цю емоцію і використовують міміку. Емоції радості й подиву добре розпізнаються, як і в попередньому тесті.

4.4 Вимірювання точності розпізнавання емоцій людини на частково закритому обличчі

При дослідженні проблем, пов'язаних з розпізнаванням емоцій, було виділено проблему у розпізнавання у випадку, коли обличчя людини частково закрито. Це можуть бути окуляри, борода чи вуса, ластовиння.

Розроблена система була перевірена на точність розпізнавання емоцій людини, частково закрито. У вимірюванні перевірювались емоції людей, що носять окуляри, мають бороду та ластовиння. Результати дослідження проілюстровано в таблицях 7 і 8 та на рисунках 35 і 36.

Таблиця 7

Розпізнавання емоцій людей, що носять окуляри

	Радість	Злість	Сум	Подив	Нейтрально
Загальна кількість вимірів	10	10	10	10	10
Кількість невірних вимірів	0	4	5	0	0
Процент вірного розпізнавання	100%	60%	50%	100%	100%

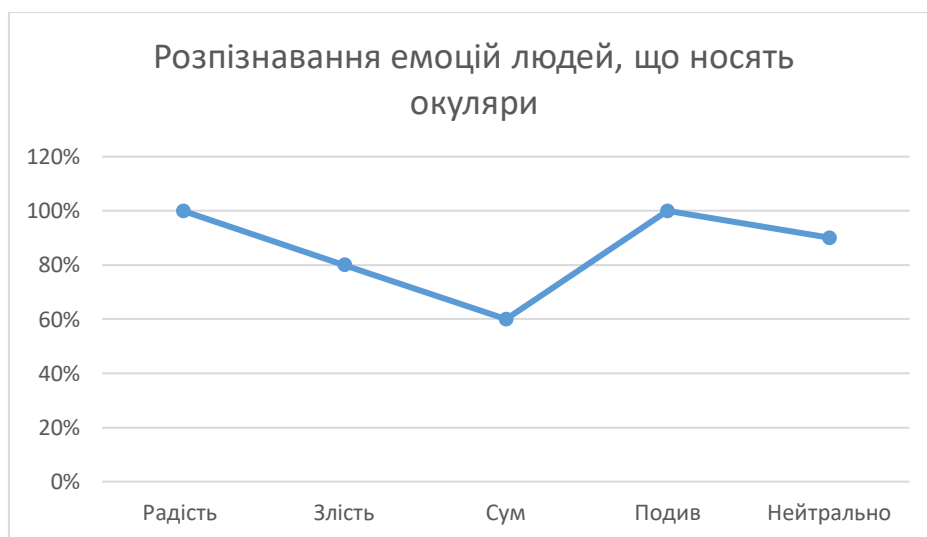


Рисунок 35 Діаграма з результатами розпізнавання емоцій людей, що носять окуляри

Таблиця 8

Розпізнавання емоцій людей, що мають бороду

	Радість	Злість	Сум	Подив	Нейтрально
Загальна кількість вимірів	10	10	10	10	10
Кількість невірних вимірів	0	2	5	0	1
Процент вірного розпізнавання	100%	80%	50%	100%	90%



Рисунок 36 Діаграма з результатами розпізнавання емоцій людей, що мають бороду

З отриманих результатів можна побачити, що наявність у людини окулярів, бороди чи ластовиння не впливає на коректність розпізнавання емоцій. Частково закриті обличчя можливо детектувати та можливо розпізнати емоції на ньому.

Отже, встановлено, що емоції радості, подиву та нейтральний стан завжди коректно розпізнаються. Емоція злість може бути розпізнана не завжди. Для коректного розпізнавання емоція повинна бути яскраво виражена: зморщені брови, ніс, відкритий рот. Найскладніше розпізнати емоцію сум, система може плутати її з нейтральним станом. Але, в цілому, точність розпізнавання емоцій на обличчі людини складає 82%.

ВИСНОВКИ

1. Встановлено актуальність використання систем розпізнавання емоцій людини з обличчя. Подібні системи застосовуються в багатьох сферах людської діяльності.
2. Досліджено існуючі методи для розпізнавання обличчя, їх основні переваги та недоліки.
3. Досліджено сучасні підходи до вирішення проблеми розпізнавання емоцій та можливості їх використання.
4. Побудовано згорткову нейронну мережу, що є модифікацією нейронної мережі VggNet. Розроблена нейронна мережа дозволяє розпізнавати 5 емоцій людини: злість, радість, сум, подив та нейтральний стан.
5. Реалізовано програмне забезпечення у вигляді веб-застосунку для використання нейронної мережі.
6. Створене програмне забезпечення може використовуватись в різних сферах:
 - здоров'я – для моніторингу психічного здоров'я людей;
 - маркетингу та реклами – для оцінки емоцій людей, що дивляться рекламу або розважальне шоу;
 - криміналістиці – для розпізнавання емоцій підозрюваного під час допиту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Потапов А. С. Системы комп'ютерного зору: сучасні завдання і методи. *Control Engineering*. 2014. № 1. С. 24-25.
2. Освоюємо комп'ютерний зір - 8 основних кроків. URL: <https://vc.ru/ml/76964-osvaivaem-kompyuternoe-zrenie-8-osnovnyh-shagov> (дата звернення: 12.11.2020).
3. Victor Wiley, Thomas Lucas. Computer Vision and Image Processing: A Paper Review. *International Journal Of Artificial Intelegence Research*. 2018. Vol. 2. P. 28-36.
4. Computer Vision: What it is and why it matters. URL: https://www.sas.com/en_us/insights/analytics/computer-vision.html (дата звернення: 12.11.2020).
5. Пару слів про розпізнавання образів. URL: <https://habrahabr.ru/post/208090/> (дата звернення: 13.11.2020).
6. Аналіз існуючих підходів до розпізнавання облич. URL: <https://habr.com/ru/company/synesis/blog/238129/> (дата звернення: 14.11.2020).
7. Лугуев Т. С. Методы компьютерного анализа выражения человеческого лица. *Известия ЮФУ*. 2013. №5. С. 253.
8. Пол Экман. Психология лжи. Санкт-Петербург: Питер, 2016. 384 с.
9. Paweł Tarnowski, Marcin Kołodziej, Andrzej Majkowski, Remigiusz J. Rak. Emotion recognition using facial expressions. *Procedia Computer Science*. 2017. Vol. 108. P. 1175-1184.
10. Paul Viola, Michael J. Jones. Robust Real-time Object Detection. *IJCV*. 2001. №3. P. 1.
11. Юрко И.В., Алдобаева В.Н. Исследование основных проблем, связанных с распознаванием и идентификацией лиц по видеофиксации и совершенствование работы алгоритмов распознавания лиц по видеофиксации в режиме реального времени. *Проблемы современной науки и образования*. 2018. С. 28.

12. Borylo P., Face Occurrence Verification Using Haar Cascades - Comparison of Two Approaches. *Communications in Computer and Information Science*. 2011. P. 149.
13. L. Wiskott, J.-M. Fellous, N. Kruger, C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Xplore*. 2002. Vol. 1. P. 129-132.
14. Jean-Marc Fellous. Face Recognition by Elastic Bunch Graph Matching. CRC Press. 1999. P. 355- 396.
15. Kar A. High Performance Human Face Recognition using Gabor Based Pseudo Hidden Markov Model. *International Journal of Applied Evolutionary Computation (IJAEC)*. 2013. P. 11-22.
16. Srinivasan M., Vijayakumar S., Pseudo 2D Hidden Markov Model Based Face Recognition System Using Singular Values Decomposition Coefficients. *Computer Vision, & Pattern Recognition*. 2013. P. 252-258.
17. A step by step explanation of principal component analysis. URL: <https://builtin.com/data-science/step-step-explanation-principal-component-analysis> (дата звернення: 19.11.2020).
18. J. Shlens. A Tutorial on Principal Component Analysis. *Google Research*. 2014. P. 17.
19. T. Cootes, G. Edwards, C. Taylor. Active appearance models. *European Conference on Computer Vision*. 1998. Vol. 2. P. 484–498.
20. X. Hou, S. Li, H. Zhang, Q. Cheng. Direct appearance models. *IEEE Conference on Computer Vision and Pattern Recognition*. 2001. Vol. 1. P. 828–833.
21. Безыкорнов Д. С., Тюрин А. И., Степаненко М. А., Соколова Э. С. Анализ эффективности алгоритмов поиска ключевых точек на лице в условиях недостаточной графической информации. *Труды НГТУ им. П.Е. Алексеева*. 2015. С. 102-105.
22. Исаев А. Л., Газаров Д. А., Евсеев С. Д. Распознавание лиц по изображениям. *Символ науки*. 2017. С. 70.

23. Krizhevsky, A., Sutskever, I., Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems Foundation*. 2012. P. 1097-1105.
24. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*. 2014. P. 112-117.
25. R. Feraud, O. Bemier, J.-E. Viallet. A fast and accurate face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002. Vol. 23. P. 42-53.
26. What are Recurrent Neural Networks? URL: <https://www.ibm.com/cloud/learn/recurrent-neural-networks> (дата звернення: 21.11.2020).
27. Krizhevsky Alex, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in neural information processing systems*. 2012. P. 221-252.
28. The Architecture and Implementation of VGG-16. URL: <https://medium.com/towards-artificial-intelligence/the-architecture-and-implementation-of-vgg-16-b050e5a5920b> (дата звернення: 21.11.2020).
29. Christian Szegedy , Wei Liu , Yangqing Jia , Pierre Sermanet , Scott Reed. Going Deeper with Convolutions. *Google Research*. 2015.
30. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. *IEEE Xplore*. 2016.
31. Top 8 Datasets Available For Emotion Detection URL: <https://analyticsindiamag.com/top-8-datasets-available-for-emotion-detection/> (дата звернення: 23.11.2020).
32. Deep Learning Frameworks URL: <https://developer.nvidia.com/deep-learning-frameworks> (дата звернення: 23.11.2020).
33. Ali Shatnawi, Ghadeer Al-Bdour, Raffi Al-Qurran, Mahmoud Al-Ayyoub. *A Comparative Study of Open Source Deep Learning Frameworks*. Inter-

national Conference on Information and Communication Systems. Jordan. 2018. P. 293-295.

34. Paul Barry. Head First Python. Sebastopol, California: O'Reilly Media, 2016. 625 p.

35. Django — фреймворк для веб-розробки на Python. URL: <https://web-creator.ru/articles/django> (дата звернення: 25.11.2020).

36. Django Channels Documentation. URL: <https://channels.readthedocs.io/en/stable/> (дата звернення: 25.11.2020).

37. Howse J. OpenCV: Computer Vision Projects with Python. *IEEE Xplore*. 2016. P. 541.

38. Бублик Анна, студентка магістратури ФЕЕІТ ІІ ЗНУ. Наук. кер.: к. техн. н., доц. Лимаренко Ю. О. «НЕЙРОННА МЕРЕЖА ДЛЯ АНАЛІЗУ ТА РОЗПІЗНАВАННЯ ЕМОЦІЙ НА ОБЛИЧЧІ ЛЮДИНИ». Збірник наукових праць студентів, аспірантів і молодих вчених «Молода наука-2020» : у 5 т. Запорізький національний університет. Запоріжжя: ЗНУ, 2020. Т.5. С. 77-78.

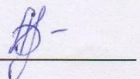
39. Бублик А. О., магістрантка, Лимаренко Ю. А., доцент, канд. техн. наук – науковий керівник. Актуальність використання систем розпізнавання емоцій людини. МАТЕРІАЛИ XXV науково-технічної конференції студентів, магістрантів, аспірантів, молодих вчених та викладачів. Запоріжжя : ЗНУ, 2020. С. 158.

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Бублик Анна Олександрівна, студентка 2 курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти sp115-04@stu.zsea.edu.ua, — підтверджую, що написана мною кваліфікаційна робота на тему **«Нейромережна система для розпізнавання емоцій на обличчі людини»** відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-систем, а також на архівування моєї роботи в базі даних цієї системи.

Дата 30.11.2020 Підпис  Бублик Анна Олександрівна (студентка)

Дата 30.11.2020 Підпис  Лимаренко Юлія Олексіївна
(науковий керівник)