

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки

(повна назва факультету)

Кафедра комп'ютерних та інформаційних технологій і систем

(повна назва кафедри)

**Пояснювальна записка
до дипломного проекту (роботи)**

магістра

(освітньо-кваліфікаційний рівень)

на тему

Розробка програмного забезпечення для розпізнавання емоцій людини на
зображенні

Виконав: студент 6 курсу, групи 602-ТН
спеціальності

122 Комп'ютерні науки

(шифр і назва напрямку)

Худя І.В.

(прізвище та ініціали)

Керівник

Янко А.С.

(прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ «ПОЛТАВСЬКА ПОЛІТЕХНІКА
ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
спеціальність 122 «Комп'ютерні науки»**

на тему

**«Розробка програмного забезпечення для розпізнавання емоцій людини
на зображенні»**

Студента групи 602-ТН Худі Ігора Васильовича

Керівник роботи
кандидат технічних наук,
доцент Янко А.С.

Завідувач кафедри
кандидат технічних наук,
доцент Двірна О.А.

РЕФЕРАТ

Кваліфікаційна робота магістра: 102 с., 51 рисунок, 1 додаток, 36 джерел.

Об'єкт дослідження: програмне забезпечення, яке використовує сучасні методи та алгоритми для аналізу зображень і розпізнавання емоцій на основі рис обличчя.

Мета роботи: розробка програмного забезпечення, яке дозволить точно ідентифікувати емоції людини на основі аналізу зображень.

Методи: аналіз існуючих рішень, структурне програмування, нейронні мережі, машинне навчання, обробка зображень та моніторинг системних ресурсів.

Ключові слова: PYTHON, DEEPFACE, IMAGE, TKINTER, ОБРОБКА ЗОБРАЖЕНЬ, ЕМОЦІЇ, ГРАФІЧНИЙ ІНТЕРФЕЙС.

ANNOTATION

Qualification work of master's degree: 102 p., 51 figures, 1 application, 36 sources.

Object of study: software that uses modern methods and algorithms to analyze images and recognize emotions based on facial features.

The goal of the work: to develop software that will accurately identify human emotions based on image analysis.

Methods: analysis of existing solutions, structured programming, neural networks, machine learning, image processing, and system resource monitoring.

Keywords: PYTHON, DEEPFACE, IMAGE, TKINTER, IMAGE PROCESSING, EMOTIONS, GRAPHICAL INTERFACE.

ЗМІСТ

| | |
|-------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ..... | 5 |
| ВСТУП..... | 6 |
| РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ЕМОЦІЙ | 8 |
| 1.1 Моделі емоцій | 8 |
| 1.2 Традиційні методи розпізнавання емоцій..... | 19 |
| 1.3 Огляд наявних наборів даних для розпізнавання емоцій | 27 |
| 1.4 Основи комп'ютерного зору | 32 |
| РОЗДІЛ 2 ЗАСТОСУВАННЯ ГЛИБОКОГО НАВЧАННЯ У РОЗПІЗНАВАННІ ЕМОЦІЙ..... | 39 |
| 2.1 Алгоритми глибокого навчання | 39 |
| 2.2 Передобробка зображень..... | 43 |
| 2.3 Виявлення обличчя | 46 |
| 2.4 Класифікація емоцій нейронними мережами..... | 50 |
| 2.5 Алгоритми класифікації | 54 |
| РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ | 59 |
| 3.1 Огляд інструментів та технологій..... | 59 |
| 3.2 Функціональні вимоги до ПЗ | 66 |
| 3.3 Нефункціональні вимоги до ПЗ | 68 |
| 3.4 Реалізація програмного забезпечення..... | 70 |
| 3.5 Тестування та оцінка ефективності..... | 78 |
| ВИСНОВКИ | 92 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 93 |
| ДОДАТОК А | 97 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- AI (Artificial Intelligence) – штучний інтелект;
- CNN (Convolutional Neural Network) – згорткова нейронна мережа;
- ML (Machine Learning) – машинне навчання;
- DL (Deep Learning) – глибинне навчання;
- OCR (Optical Character Recognition) – оптичне розпізнавання символів;
- CV (Computer Vision) – комп'ютерний зір;
- API (Application Programming Interface) – інтерфейс програмування;
- GPU (Graphics Processing Unit) – графічний процесор;
- CPU (Central Processing Unit) – центральний процесор;
- RAM (Random Access Memory) – оперативна пам'ять;
- FPS (Frames Per Second) – кадри за секунду;
- UI (User Interface) – користувацький інтерфейс;
- UX (User Experience) – користувацький досвід;
- FER (Facial Emotion Recognition) – розпізнавання емоцій за обличчям;
- ROI (Region of Interest) – область інтересу;
- DLIB – бібліотека для комп'ютерного зору та обробки обличчя;
- KNN (K-Nearest Neighbors) – алгоритм найближчих сусідів;
- LFW (Labeled Faces in the Wild) – база даних для розпізнавання обличчя.

ВСТУП

Сьогодні в суспільстві, де взаємодія між людьми все частіше здійснюється через цифрові платформи та дистанційні комунікації, розпізнавання емоцій набуває особливої актуальності. Технології, що дозволяють ідентифікувати емоції людини за допомогою зображень, можуть сприяти розвитку різних сфер: від індустрії розваг до медицини та психології. Завдяки таким технологіям з'являється можливість автоматизувати процеси моніторингу емоційних станів, що може бути корисним у сфері освіти, маркетингу, комунікацій та навіть безпеки.

Актуальність теми: у той час, як технології штучного інтелекту та комп'ютерного зору продовжують розвиватися, все більше виникає потреба в системах, здатних розпізнавати емоції на зображеннях. Це відкриває можливості для нових інновацій, зокрема в галузі психології, де автоматизовані системи можуть бути використані для аналізу емоційних реакцій під час різних ситуацій. Розробка програмного забезпечення для розпізнавання емоцій також допомагає підвищити якість взаємодії між людиною та машиною, покращуючи взаєморозуміння.

Проблематика дослідження полягає в тому, що попри досягнення у сфері розпізнавання емоцій, існують значні проблеми, пов'язані з точністю ідентифікації емоційних станів. Складність полягає у варіативності виразу обличчя, впливі освітлення, ракурсу, а також культурних та індивідуальних відмінностей у вираженні емоцій. Більше того, сучасні моделі інколи стикаються з труднощами в розпізнаванні складних емоцій або змішаних емоційних станів. Це створює потребу у вдосконаленні алгоритмів та розробці нових підходів до аналізу емоцій на зображеннях.

Однією з головних труднощів є залежність точності розпізнавання емоцій від якості зображення та різних зовнішніх чинників, таких як освітлення або кут нахилу голови. Також важливо враховувати проблему відсутності універсальної моделі, яка б точно розпізнавала емоції незалежно

від етнічної чи культурної приналежності. Більше того, для розпізнавання складних або змішаних емоційних станів існують обмеження у наявних алгоритмах, що ставить під питання їхню універсальність.

Метою цієї роботи є розробка програмного забезпечення, яке дозволить точно ідентифікувати емоції людини на основі аналізу зображень. Це програмне забезпечення повинно мати достатню точність у різних умовах освітлення та ракурсах, а також здатність працювати з різними типами емоцій та форматами зображень.

Об'єктом дослідження є процеси розпізнавання емоцій на основі зображень обличчя людини за допомогою алгоритмів комп'ютерного зору та машинного навчання.

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ЕМОЦІЙ

1.1 Моделі емоцій

Науковці Фер і Рассел у 1984 підкреслили труднощі у виробленні чіткого визначення емоцій, коли написали, що «всі знають, що таке емоція, доки їх не попросять дати визначення. Тоді, здається, ніхто не знає».

Визначення емоцій варіюються не лише залежно від дисциплін чи підходів, але й від історії та культури. Науковці наголошують на необхідності розглянути, чи існує історія емоцій, тобто розуміння того, як емоції та концепція емоцій могли змінюватися протягом історичного часу. Як описує Констан, англійський термін «емоція» з'явився відносно недавно і став використовуватися частіше, ніж, наприклад, «пристрасть», «прихильність» і «почуття», лише в останні 200 років.

Задовго до цього періоду можна зустріти й інші терміни, близькі за значенням до «емоції», наприклад, давньогрецький термін «пафос». Насправді, визначення Аристотелем пафосу як «того, через що люди змінюються і відрізняються у своїх судженнях, і що супроводжує біль і задоволення» можна вважати одним з перших важливих експліцитних визначень емоцій. Це визначення було важливим не лише тому, що воно вказувало на зв'язок між емоціями та судженнями, але й тому, що воно вже містило вимір, який майже всі сучасні моделі вважають необхідним: валентність (тут – «біль і задоволення»).

Історію емоцій можна простежити з часів визначення Аристотеля, розглядаючи, як еволюціонували визначення і чи змінювалися емоції протягом історичного часу; наприклад, чи є сьгоднішні емоції «сорому» або «гніву» такими самими, як ті, що описували стародавні греки, месопотамці або представники інших цивілізацій. Звичайно, відмінності, згадані тут у часовому вимірі, можна дослідити і в просторовому. Хоча відмінності, виявлені

протягом століть, не можуть бути безпосередньо досліджені афективною нейронаукою, культурні відмінності, які спостерігаються сьогодні, є класичною темою психології емоцій і почали досліджуватися з точки зору афективної нейронауки – як, наприклад, публікація спеціального випуску, присвяченого культурній нейронауці, в журналі «Соціальна когнітивна та афективна нейронаука».

Розгляд історичних та культурних впливів на емоції виходить далеко за межі цієї глави, але, як буде показано далі, питання про те, чи є емоції універсальним феноменом, чи вони змінюються залежно від часу та простору, є фундаментальним для багатьох теорій в афективних науках.

Іншим прикладом категорії, яка визначається типом емоції, є поширена валентна відмінність між «позитивними» та «негативними» емоціями.

Наприклад, вагомим внеском Томкінса у 1963 в афективні науки була книга, розділена на два томи: перший том присвячений позитивним афектам, а другий – негативним. Хоча тип валентності, який використовується для розрізнення так званих позитивних і негативних емоцій, часто не є чітким, часто розглядається саме компонент відчуття: Емоція є позитивною, коли «вона відчувається приємною», або негативною, коли «вона відчувається неприємною».

Це розрізнення на основі валентності стало ключовим для розвитку циркумлексних/двомірних теорій емоцій, які обговорюються далі (див., зокрема, розділ «Чи є емоція почуттям?»). Звичайно, вимір валентності не обмежується компонентом відчуття; події, що викликають емоції, іноді класифікуються як позитивні або негативні з точки зору їхньої внутрішньої приємності або сприятливості для досягнення мети.

Не завжди існує відповідність між оціночною валентністю події та валентністю почуття. Наприклад, хоча емоція «інтерес» вважається позитивною з точки зору відчуття, вона також може бути викликана негативними стимулами (наприклад, огидні стимули можуть викликати інтерес) [1].

Хоча почуття часто розглядаються як позитивні або негативні, деякі вчені стверджують, що оцінка подій може бути амбівалентною. Це означає, що людина може відчувати як добре, так і погане ставлення до події, а не тільки добре чи погано. Залежно від того, який аспект події оцінюється, та сама подія може бути оцінена як позитивна чи негативна, тобто, якщо два аспекти оцінюються одночасно двома відокремленими оціночними каналами, можуть виникати як позитивні, так і негативні почуття.

Наприклад, сексуальні стосунки з кимось іншим, ніж значуща інша людина, можуть оцінюватися як позитивні в тому сенсі, що вони приносять задоволення, але як негативні, тому що вони втручаються в моральні проблеми людини. Амбівалентне ставлення розглядалося як доказ відокремленості позитивних і негативних субстратів, а також того, що змішані емоції можуть викликати спільні емоції. Наприклад, було висловлено припущення, що під час перегляду трагікомічних фільмів люди можуть відчувати одночасно і радість, і смуток. В афективній нейронауці уявлення про те, що системи мозку можуть бути диференційовано залучені в обробку позитивних і негативних стимулів, ґрунтується на різних дослідницьких традиціях. Наприклад, розуміння мозкових механізмів, задіяних у системі болю/відрази, на відміну від системи заохочення/винагороди, було основною метою афективної нейронауки.

Ліberman та Айзенбергер у 2009 припустили, що «мережа болю» складається з дорсально-передньої поясної кори, інсули, соматосенсорної кори, таламуса та периакведуктальної звивини (див. розділ 9), тоді як «мережа винагороди» складається з вентральної тегментальної області, вентрального стріатума, вентромедіальної префронтальної кори та мигдалини.

Інша споріднена традиція ілюструється роботою Берріджа та його колег, які розрізняють мозкові механізми, що обслуговують процеси «подобатися» та «хотіти». Цей напрямок досліджень привів до припущення, що в мозку існує кілька «гедоністичних точок», включаючи ділянки ядра і вентральної долоні, які беруть участь у процесі симпатії, тоді як дофамінова система, яка часто

вважається посередником задоволення, насправді опосередковує особливу форму прагнення до винагороди, яка називається «виразність стимулу».

Ця відмінність між симпатією і бажанням також може, в принципі, призвести до амбівалентних процесів, враховуючи, що, наприклад, людям може подобатися те, чого вони не хочуть, або хотіти те, що їм не подобається, що узгоджується з уявленням про те, що люди можуть одночасно хотіти чогось більше і менше. Така відмінність між симпатією і бажанням свідчить на користь важливості розрізнення різних типів валентностей в афективній нейронауці, як це має місце в інших дисциплінах. Цей підхід може мати значний вплив на літературу про вподобання, цінності та прийняття рішень, як це обговорюється, наприклад, у нейроекономічній літературі, оскільки він може бути доповненням до підходу, який передбачає переведення всіх валентних процесів у спільну цінність [1].

Інша дослідницька традиція, яка вплинула на пояснення афективної неврології щодо опозиції між позитивними та негативними емоціями, ґрунтується на дослідженнях функціональної півкульної асиметрії. Так звана валентна гіпотеза півкульної асиметрії стверджує, що існує центр позитивних почуттів у лівій півкулі та центр негативних почуттів у правій півкулі (напр.,

Ця гіпотеза є дуже дискусійною, і в літературі було запропоновано кілька альтернативних варіантів. Наприклад, так звана правопівкульна гіпотеза припускає, що всі механізми, пов'язані з емоціями, більш латералізовані в правій півкулі.

Альтернативою «валентній гіпотезі», описаній вище, є теорія «асиметрії мозку до підходу», яку в основному досліджували Девідсон та його колеги.

Девідсон та Ірвін у 1999 році припустили існування системи підходу, яка сприяє апетитній поведінці та генерує деякі пов'язані з підходом позитивні емоції, такі як гордість або радість. Такі емоції генеруються в контексті руху до мети. Така система була б латералізована до лівої півкулі. Ці автори також постулювали другу систему, яка, однак, була б латералізована до правої півкулі. Ця система полегшує абстиненцію і генерує деякі пов'язані з

абстиненцією негативні емоції, такі як страх або відраза. Опозиція між поведінкою наближення та уникнення в емоційному реагуванні дуже часто пропонується і вважається, що вона має сильну філогенетичну основу.

Більшість теорій емоцій визнають тенденції до наближення до задоволення та уникнення болю, і ця концепція бере свій початок у філософських працях. Наприклад, науковець Гоббс запропонував розмежування між цими двома типами поведінки, коли писав, що «це прагнення, коли воно спрямоване на те, що його викликає, називається апетитним. А коли зусилля спрямоване від чогось, воно зазвичай називається Відразою. Слова «Апетит» і «Відраза» означають емоції, одна з яких – наближення, інша – віддалення». Особливо цікавим аспектом гіпотези асиметрії півкуль є те, що вона не перетинається з гіпотезою валентності, оскільки стверджує, що негативна, але пов'язана з наближенням емоція, така як гнів, буде латералізована в лівій півкулі.

Поділ на «підхід та уникнення», або «апетит та уникнення», є важливим, оскільки дозволяє відрізнити дисоціацію на основі валентності від дисоціації на основі тенденції до дії [1].

Моделі емоцій є важливими інструментами для розуміння складної природи людських почуттів, їх впливу на поведінку та психічний стан. Однією з найвідоміших концепцій у цій сфері є розподіл емоцій на позитивні та негативні. Позитивні емоції відіграють ключову роль у підтриманні психологічного благополуччя, формуванні соціальних зв'язків і розвитку особистісних якостей.

Позитивні емоції (див. Рис. 1.1) – це такі емоційні стани, які пов'язані з приємними переживаннями, відчуттям радості, задоволення і натхнення. Вони сприяють збільшенню рівня щастя, мотивують до активної діяльності, покращують продуктивність та творчий потенціал. До основних позитивних емоцій належать радість, задоволення, натхнення, любов, вдячність, захоплення, гордість, оптимізм і впевненість у собі.



Рисунок 1.1 – Приклади вираження позитивних емоцій

Існує кілька моделей, що допомагають класифікувати та аналізувати емоції. Дві найвідоміші моделі емоцій – це дискретна та вимірна.

1. Дискретні моделі емоцій: базуються на теорії, що емоції можна поділити на певні основні категорії. Одним із засновників цієї ідеї був американський психолог Пол Екман, який запропонував модель "основних емоцій". За цією моделлю, існує набір універсальних емоцій, що властиві всім культурам і націям. До них належать шість базових емоцій: радість, смуток, гнів, страх, здивування та відраза. Пізніше Екман додав ще кілька емоцій, серед яких з'явилися вдячність і задоволення, що належать до позитивних.

2. Вимірні моделі емоцій: ці моделі описують емоції як точки на багатовимірному просторі. Однією з найвідоміших вимірних моделей є Циркумплексна модель емоцій Джеймса Рассела. Вона пропонує дві основні осі: одна вісь вимірює "валентність" (позитивний або негативний характер емоції), а інша – "збудження" (інтенсивність емоції). Позитивні емоції в цій моделі займають верхню частину діаграми, оскільки зазвичай асоціюються з високим рівнем задоволення та активності.

Теорія розширення і побудови позитивних емоцій: позитивні емоції відіграють критичну роль у людському житті не тільки завдяки їх

безпосередньому впливу на настрій, але й через їхній внесок у довготривале психологічне зростання. Професор психології Барбара Фредріксон розробила теорію розширення і побудови ("Broaden-and-Build Theory"), яка стверджує, що позитивні емоції сприяють розширенню когнітивних і поведінкових ресурсів людини. Коли людина переживає позитивні емоції, її здатність думати ширше, відкривати нові можливості та будувати соціальні зв'язки зростає [2].

Ця теорія пояснює, як позитивні емоції можуть призводити до довготривалих змін у житті. Наприклад, емоції, як-от вдячність чи гордість, зміцнюють соціальні відносини, а натхнення чи радість стимулюють творчу активність і відкриття нових знань.

Кожна позитивна емоція має унікальний ефект на поведінку та сприйняття. Ось деякі з ключових позитивних емоцій:

1. Радість: це базова позитивна емоція, яка виникає в результаті досягнення бажаних цілей чи подій. Радість стимулює людину до повторення дій, які викликали цей стан, і сприяє загальному відчуттю благополуччя;

2. Натхнення: це відчуття, яке мотивує людину до творчості, інновацій та самовдосконалення. Натхнення часто виникає як результат контакту з чимось винятковим або піднесеним;

3. Вдячність: це почуття, яке виникає, коли людина усвідомлює позитивні дії або допомогу, отриману від інших. Вдячність сприяє зміцненню соціальних зв'язків і відчуття задоволеності життям;

4. Гордість: це позитивна емоція, яка виникає в результаті досягнення важливих особистих цілей або успіху. Гордість може мотивувати до подальших досягнень і підвищувати самооцінку;

5. Любов: одна з найпотужніших позитивних емоцій, яка сприяє розвитку глибоких зв'язків між людьми. Любов включає спектр емоцій від романтичної прив'язаності до альтруїстичної турботи про інших;

6. Захоплення: це відчуття подиву і радості від чогось вражаючого або прекрасного, яке сприяє зростанню інтересу та допитливості до світу;

7. Оптимізм: віра в позитивний результат подій, навіть за умов невизначеності. Оптимізм є ключовою емоцією для подолання труднощів і стресу [3].

Негативні емоції (див. Рис. 1.2) є невід'ємною частиною людського досвіду, вони виникають у відповідь на стресові ситуації, небезпеки або проблеми. Хоча часто вони асоціюються з дискомфортом і неприємними відчуттями, негативні емоції також мають важливу функцію в житті людини, допомагаючи пристосовуватися до складних ситуацій, ухилятися від загроз і зберігати виживання. Негативні емоції включають страх, смуток, гнів, відразу, сором, провину, тривогу та заздрість.



Рисунок 1.2 – Приклади вираження негативних емоцій

Негативні емоції – це такі емоційні стани, що спричиняють неприємні відчуття і можуть призводити до зниження загального рівня задоволення життям або до виникнення стресу. Вони виникають як реакція на ситуації, що сприймаються як небезпечні, несправедливі, фруструючі або шкідливі. Однак, не всі негативні емоції є руйнівними – деякі з них виконують захисні та адаптивні функції, мотивуючи до дій, які можуть уникнути проблем або мінімізувати шкоду.

Як і позитивні емоції, негативні також досліджуються через різні моделі емоцій. Дві головні моделі, що використовуються для аналізу емоцій, це дискретні моделі та вимірні моделі.

1. Дискретні моделі емоцій: у рамках цієї моделі емоції класифікуються як окремі та відносно незалежні один від одного стани. Пол Екман, засновник моделі основних емоцій, виділяв негативні емоції, такі як гнів, страх, смуток і відраза, серед базових емоцій, універсальних для всіх культур. Ці емоції виникають у відповідь на загрози чи втрати, і їхня функція полягає в захисті людини від шкідливих або небезпечних ситуацій.

2. Вимірні моделі емоцій: вимірні моделі, такі як Циркумплексна модель емоцій Джеймса Рассела, розташовують емоції на двох осях: "валентність" (позитивність або негативність) і "збудження" (інтенсивність емоції). Негативні емоції мають негативну валентність і можуть розміщуватися в різних точках на шкалі збудження. Наприклад, страх та гнів мають високий рівень збудження, тоді як смуток і сором зазвичай відрізняються нижчим рівнем енергії.

Незважаючи на те, що негативні емоції часто сприймаються як небажані, вони мають важливу роль у виживанні та адаптації. Ось кілька прикладів того, як негативні емоції можуть бути корисними:

1. Страх: ця емоція викликає швидку мобілізацію організму, стимулюючи реакцію на загрозу (наприклад, "бий або біжи"). Страх допомагає уникати небезпек і діяти швидко в ситуаціях, коли життя або безпека знаходяться під загрозою.

2. Гнів: гнів є реакцією на несправедливість або порушення прав. Він може мотивувати людину до активної боротьби за свої права або до вирішення конфліктів. Однак, надмірний або неконтрольований гнів може бути руйнівним, якщо його не спрямовувати в конструктивне русло.

3. Смуток: смуток, який часто виникає після втрати або невдачі, допомагає людині прийняти ситуацію і адаптуватися до неї. Смуток може

слугувати сигналом для оточуючих про потребу в підтримці, сприяючи соціальній взаємодії та емоційному зціленню.

4. Відраза: це негативна емоція, яка виникає як відповідь на щось огидне або небезпечне для здоров'я. Відраза має важливу біологічну функцію – захищає людину від потенційно шкідливих або токсичних речовин, таких як зіпсована їжа або отруйні хімічні речовини.

5. Тривога: тривога виникає як реакція на потенційні загрози чи невизначеність у майбутньому. Ця емоція може стимулювати до підготовки та планування, щоб уникнути небажаних наслідків. Водночас хронічна тривога може призводити до стресу та виснаження, тому важливо навчитися управляти цією емоцією.

6. Сором і провина: ці емоції мають соціальний характер і виникають у відповідь на порушення моральних або соціальних норм. Сором може спонукати людину до самовдосконалення та уникання повторних помилок. Провина, у свою чергу, може стимулювати до відшкодування або виправлення шкоди, завданої іншим [4].

Нейтральні емоції (див. Рис. 1.3) часто залишаються менш обговорюваними порівняно з позитивними або негативними, оскільки вони не викликають яскраво виражених почуттів задоволення або дискомфорту. Проте вони також відіграють важливу роль у житті людини, дозволяючи їй зберігати стабільність, об'єктивність та баланс між різними емоційними станами. Нейтральні емоції – це ті емоційні стани, які не викликають чітко вираженої реакції радості чи смутку, задоволення чи роздратування. Вони можуть служити перехідним етапом між іншими емоціями або фоновими емоціями під час повсякденних справ.



Рисунок 1.3 – Приклади вираження нейтральних емоцій

Нейтральні емоції – це ті емоційні стани, які не мають яскраво вираженої позитивної або негативної валентності. Іншими словами, людина не відчуває ані задоволення, ані страждання, коли перебуває у цьому стані. Нейтральні емоції можуть включати такі почуття, як спокій, рівновага, байдужість або зосередженість.

Наприклад, під час виконання звичайної рутинної роботи, яка не приносить ані радості, ані розчарування, людина може перебувати в нейтральному емоційному стані. Такі емоції часто сприяють концентрації на завданнях, об'єктивності в прийнятті рішень і можуть допомагати уникнути зайвих емоційних витрат у ситуаціях, які не потребують емоційного залучення.

Приклади нейтральних емоцій:

1. **Спокій:** це один із найбільш поширених прикладів нейтральної емоції. Людина відчуває спокій, коли немає явних причин для радості чи тривоги, і вона може зосередитися на поточних завданнях або насолоджуватися моментом без сильно виражених емоцій.

2. **Байдужість:** байдужість виникає, коли людина не відчуває емоційного зв'язку або інтересу до певної ситуації, події чи іншої особи. Ця емоція дозволяє уникнути емоційного перевантаження та зберегти емоційну дистанцію в ситуаціях, що не викликають особливого залучення.

3. **Зосередженість:** коли людина повністю занурена в якесь завдання або роботу, вона може перебувати в нейтральному емоційному стані,

зосередившись на результаті і не відчуваючи при цьому сильних емоцій. Це дозволяє ефективно виконувати складні завдання без емоційних перешкод.

4. Рівновага: рівноважний емоційний стан означає відсутність коливань між сильними позитивними або негативними емоціями. Це відчуття стабільності і врівноваженості, коли людина не схиляється до жодної з крайнощів і відчувається комфортно без зайвих емоційних проявів [5].

Нейтральні емоції виконують кілька важливих функцій:

1. Збереження енергії: перебування в нейтральному емоційному стані допомагає уникнути емоційного виснаження. У ситуаціях, які не вимагають сильних емоційних реакцій, нейтральні емоції дозволяють зберегти енергію та зосередитися на важливих завданнях.

2. Об'єктивність: нейтральні емоції дозволяють приймати рішення без зайвого емоційного впливу. Наприклад, під час вирішення складних проблем або під час обговорення важливих питань людина, яка перебуває в нейтральному стані, здатна мислити більш раціонально.

3. Перехід між емоційними станами: нейтральні емоції можуть виконувати функцію "емоційного моста" між позитивними і негативними емоціями. Наприклад, після сильного гніву людина може відчути нейтральний стан перед тим, як перейти до спокою або радості.

4. Стабілізація емоційного фону: нейтральні емоції підтримують загальний баланс емоцій, допомагаючи уникнути частих коливань між позитивними і негативними емоціями. Це дозволяє зберігати стійкість у повсякденному житті [6].

1.2 Традиційні методи розпізнавання емоцій

Датчики, що використовуються для розпізнавання емоцій, в основному включають візуальні датчики, аудіодатчики, радарні датчики та інші датчики фізіологічних сигналів, які можуть збирати сигнали різних вимірів і здійснювати емоційний аналіз за допомогою певних алгоритмів. Різні датчики

мають різне застосування в розпізнаванні емоцій. Переваги та недоліки різних сенсорів для розпізнавання емоцій показані в Таблиці 1.1.

Таблиця 1.1 – Переваги та недоліки сенсорів для розпізнавання емоцій

| Датчик | Переваги | Недоліки |
|---------------------------|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Візуальний датчик | Простий збір даних; висока масштабованість | Залежність від освітлення; легке спричинення витоку конфіденційної інформації |
| Аудіо датчик | Низька вартість; широкий спектр застосувань | Недостатня стійкість для складного аналізу емоцій |
| Радарний датчик | Дистанційний моніторинг фізіологічних сигналів | Радіальний рух може спричинити перешкоди |
| Інші фізіологічні датчики | Можливість моніторингу фізіологічних сигналів, що відображають реальні емоції | Інвазивні, вимагають носіння близько до поверхні шкіри |
| Мультисенсорна комбінація | Багатший збір інформації; висока надійність | Необхідна синхронізація багатоканальної інформації; великі розрахункові витрати |

Розпізнавання емоцій на основі візуальних сенсорів є одним з найпоширеніших методів розпізнавання емоцій. Його перевагами є низька вартість і простота збору даних. В даний час візуальні сенсори в основному використовуються для розпізнавання виразу обличчя (FER) для виявлення емоцій або технології дистанційної фотоплетизмографії (rPPG) для виявлення частоти серцевих скорочень. Точність цих методів сильно падає зі зменшенням інтенсивності світла.

Процес розпізнавання виразу обличчя показаний на Рис. 1.4. Вираз обличчя може інтуїтивно відображати емоції людини. Машинам важко вловити деталі виразу обличчя так, як людині. Вираз обличчя легко приховати, що призводить до помилок розпізнавання емоцій. Наприклад, під час деяких соціальних заходів ми зазвичай ввічливо посміхаємося, навіть якщо не перебуваємо в радісному настрої. Різні люди мають різний колір шкіри, зовнішність і риси облич, що створює труднощі для точності класифікації. Риси обличчя при одній і тій же емоції можуть бути різними, а невеликі зміни

в різних емоціях однієї і тієї ж людини не дуже очевидні. Таким чином, існує проблема класифікації, пов'язана з великою внутрішньокласовою та малою міжкласовою відстанню для виявлення емоцій за допомогою розпізнавання виразу обличчя камерою. Також важко ефективно розпізнавати емоції, коли обличчя закрито (у масці) або під різними кутами зйомки.

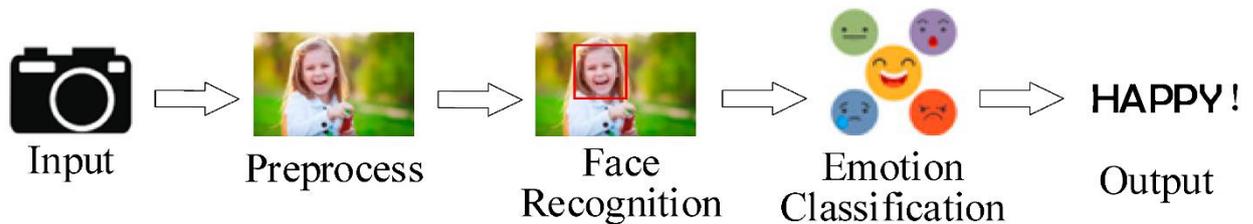


Рисунок 1.4 – Процес розпізнавання виразу обличчя

Фотоплетизмографія (ФПГ) – це оптична технологія для неінвазивного виявлення різних життєво важливих ознак, яка була вперше запропонована в 1930-х роках. ФПГ широко використовується для виявлення фізіологічних сигналів в персональних портативних пристроях (розумні браслети, розумні годинники тощо). Успішне застосування PPG призвело до швидкого розвитку дистанційної фотоплетизмографії (rPPG). Багатохвильова RGB-камера використовується в технології rPPG для виявлення незначних змін кольору шкіри на обличчі людини, викликаних змінами об'єму крові під час серцебиття, як показано на Рис. 1.5. rPPG-технологія може бути використана для отримання ступеня звуження периферичних судин і аналізу емоцій учасника дослідження. Зовнішнє звуження судин вважається захисною фізіологічною реакцією.

Коли людина перебуває в стані болю, голоду, страху або гніву, звуження зовнішніх кровоносних судин посилюється. І навпаки, у спокійному або розслабленому стані ця реакція зменшується [7].

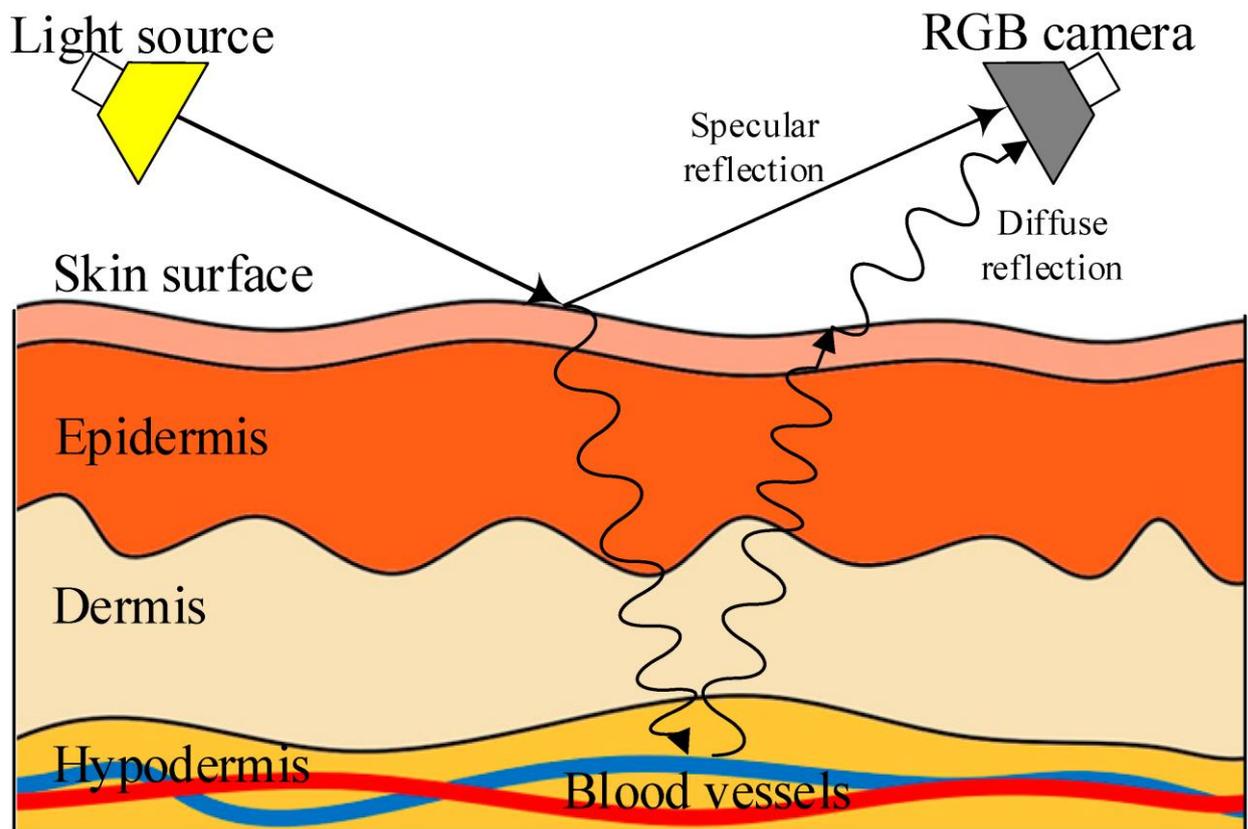


Рисунок 1.5 – Схема технології rPPG

З удосконаленням апаратного та алгоритмічного рівня технологія rPPG може також реалізувати віддалений безконтактний моніторинг та оцінку частоти серцевих скорочень, частоти дихання, артеріального тиску або інших сигналів. Розпізнавання емоцій здійснюється після аналізу великої кількості даних моніторингу. За цими сигналами можна класифікувати емоції на кілька типів та інтенсивностей. Існують певні помилки при розпізнаванні декількох типів емоцій. Для підвищення точності розпізнавання емоцій необхідно комбінувати іншу фізіологічну інформацію.

Мова – одна з найважливіших складових людської культури. За допомогою мови люди можуть виражати себе або спілкуватися з іншими. Розпізнавання мови сприяло розвитку розпізнавання мовних емоцій (SER). Людська мова містить багату інформацію, яка може бути використана для розпізнавання емоцій. Розуміння емоцій в інформації є важливим для штучного інтелекту, щоб вести ефективний діалог. SER можна використовувати для діалогу в колл-центрах, системах автоматичного

реагування, діагностики аутизму тощо. SER спільно доповнюється екстракцією акустичних ознак та мовними ознаками. Процес SER показано на Рис. 1.6.

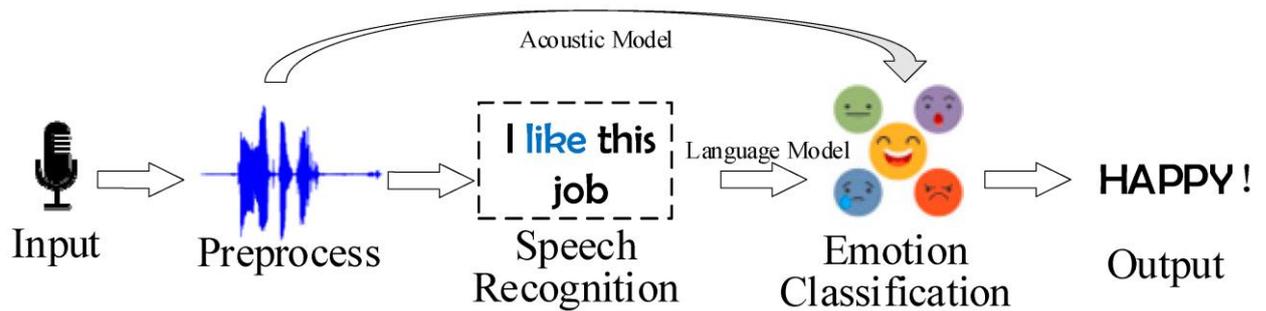


Рисунок 1.6 – Схема процесу розпізнавання мовних емоцій

На етапі попередньої обробки вхідний сигнал розбивається на сегменти після зменшення шуму, а потім виконується виділення та класифікація ознак. Мовна модель може ідентифікувати емоційні вирази зі специфічним семантичним внеском. Акустична модель може розрізняти різні емоції, що містяться в одному реченні, аналізуючи особливості просодії або спектра. Поєднання цих двох моделей може підвищити точність SER.

Розуміння емоцій у мовленні – складний процес. Різні стилі мовлення різних людей призводять до акустичної варіабельності, що безпосередньо впливає на маркування та вилучення мовних ознак. Одне й те саме речення може містити різні емоції, а деякі специфічні емоційні відмінності часто залежать від місцевої культури або середовища проживання мовця, що також створює виклики для SER.

Різні емоції викликають низку фізіологічних реакцій, таких як зміна частоти дихання, серцебиття, мозкових хвиль, кров'яного тиску тощо. Наприклад, збудження, викликане щастям, гнівом або тривогою, може призвести до збільшення частоти серцевих скорочень. Позитивні емоції можуть збільшувати частоту дихання, а депресивні емоції, навпаки, можуть пригнічувати дихання. Частота дихання також впливає на варіабельність серцевого ритму (BCR), яка зменшується на видиху і збільшується на вдиху. В даний час радіолокаційні технології широко використовуються для

дистанційного виявлення життєво важливих ознак і бездротового зондування. Радіолокаційні датчики можуть використовувати ехо-сигнал цілі для аналізу мікрорухів грудної клітки, викликаних диханням і серцебиттям. Вони можуть здійснювати дистанційне отримання цих фізіологічних сигналів. Загальний процес розпізнавання емоцій на основі радарного датчика показаний на Рис. 1.7 [7].

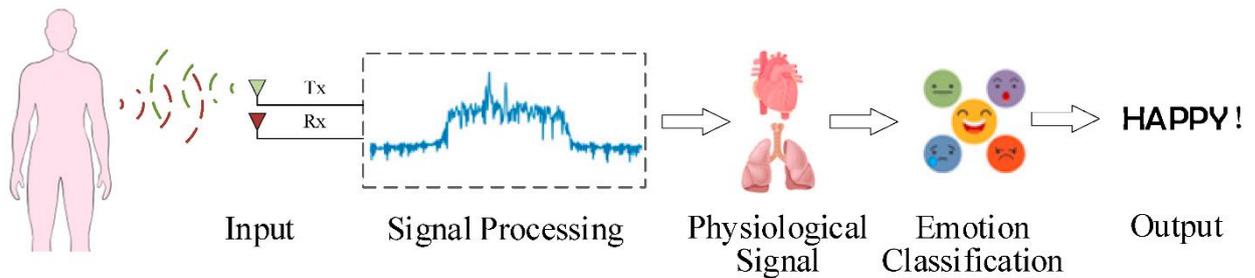


Рисунок 1.7 – Схема процесу розпізнавання ехо-сигналів

Порівняно з візуальними датчиками, розпізнавання емоцій на основі радіолокаційних датчиків не обмежується інтенсивністю світла. Однак у реальних умовах на ехо-сигнали радарів впливають шуми, особливо для радіального доплерівського руху поблизу або далеко від радара, що впливає на точність аналізу емоцій.

З давніх часів було доведено, що емоції мають біологічну природу. Вважається, що надмірні емоції мають певний вплив на функціонування життєво важливих органів. Аристотель вважав, що вплив емоцій на фізіологію проявляється у зміні фізіологічних станів, таких як прискорене серцебиття, підвищення температури тіла або втрата апетиту. Вільям Джеймс вперше запропонував теорію фізіології емоцій. Він вважав, що зовнішні подразники викликають активність у вегетативній нервовій системі та створюють фізіологічну реакцію в мозку. Наприклад, коли ми відчуваємо радість, ми сміємося; коли ми відчуваємо страх, наше волосся стає дибки; коли нам сумно, ми плачемо [7].

Людська емоція – це спонтанний психічний стан, який відображається у фізіологічних змінах людського тіла і суттєво впливає на нашу свідомість.

Багато інших фізіологічних сигналів в організмі людини, таких як електроенцефалограма (ЕЕГ), електрокардіограма (ЕКГ), електроміограма (ЕМГ), гальванічна реакція шкіри (ГШР), об'ємний пульс крові (ОПК) та електроокулографія (ЕОГ), як показано на Рис. 1.8.

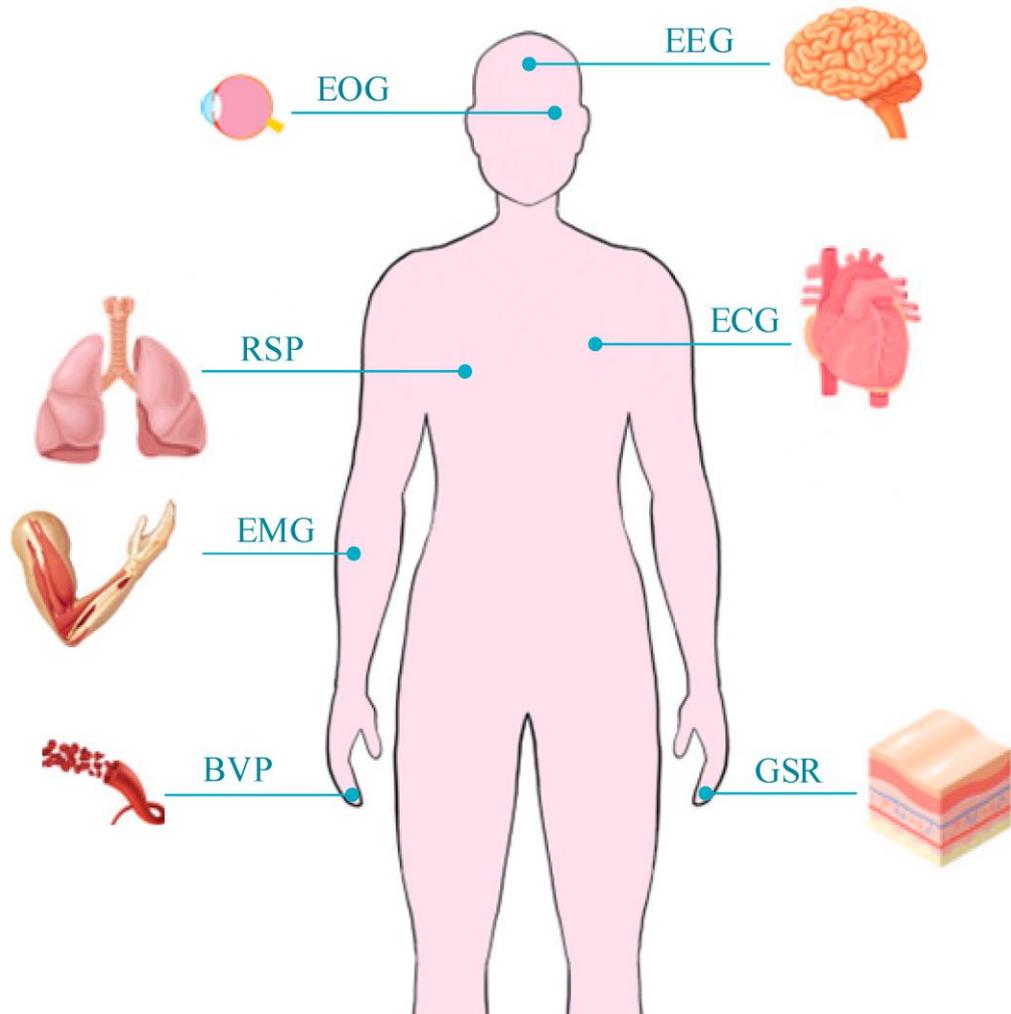


Рисунок 1.8 – Схема фізіологічних сигналів в організмі людини

ЕЕГ вимірює електричну активність мозку шляхом встановлення електродів на поверхні шкіри голови. Багато досліджень показали, що префронтальна кора, скронева частка і передня поясна звивина мозку пов'язані з контролем емоцій. Рівень їх активності викликає такі емоції, як тривога, дратівливість, депресія, занепокоєння та образа відповідно.

ЕКГ – це метод електричного моніторингу на поверхні шкіри, який виявляє серцебиття, контрольоване електричними сигналами організму. Частота серцевих скорочень і варіабельність серцевого ритму, отримані

шляхом подальшого аналізу, широко використовуються в афективних обчисленнях. Частота серцевих скорочень і варіабельність серцевого ритму контролюються симпатичною і парасимпатичною нервовими системами. Симпатична нервова система може прискорювати серцевий ритм, що відображається на більшому психологічному стресі та активації. Парасимпатична нервова система відповідає за зниження частоти серцевих скорочень до нормального рівня, переводячи організм у більш розслаблений стан.

ЕМГ вимірює ступінь активації м'язів шляхом збору різниці напруг, що генеруються під час м'язового скорочення. Сучасні технології вимірювання ЕМГ-сигналів можна розділити на два типи. Перший полягає у вивченні міміки за допомогою вимірювання лицьових м'язів. Другий – розміщення електродів на тілі для розпізнавання емоційних рухів.

Ще одним сигналом, який зазвичай використовується для розпізнавання емоцій, є електроенцефалограма. Шкіра людини зазвичай є ізолятором. Коли потові залози виділяють піт, електропровідність шкіри змінюється, тому GSR може відображати стан потовиділення людини. Зазвичай GSR вимірюють на долонях або підшвах ніг, де потові залози, як вважається, найкраще відображають зміни в емоціях. Коли людина перебуває в тривожному або напруженому настрої, потові залози зазвичай виділяють більше поту, що викликає більшу зміну струму.

Пов'язані з цим фізіологічні сигнали також включають BVP, EOG тощо. Всі ці сигнали змінюються при емоційних змінах, і вони не підлягають свідомому контролю з боку людини. Тому ці сигнали можуть бути виміряні різними фізіологічними сенсорами для досягнення мети розпізнавання емоцій. За допомогою цих фізіологічних сенсорів можна точно і швидко отримати реальні фізіологічні сигнали людини. Однак фізіологічні датчики, крім візуальних, аудіо- та радіолокаційних, зазвичай потребують дотику до шкіри або носіння відповідного обладнання для отримання фізіологічних сигналів, що вплине на щоденний комфорт людей (більшість людей не сприймають

такий метод моніторингу). Контактні датчики обмежені за вагою та розміром. Ці контактні пристрої також можуть викликати у людей напругу і занепокоєння, що вплине на точність розпізнавання емоцій [7].

1.3 Огляд наявних наборів даних для розпізнавання емоцій

Набір даних FER2013 (Facial Expression Recognition 2013) (див. Рис. 1.9) був створений для конкурсу на Kaggle у 2013 році і став одним із найвідоміших у сфері розпізнавання емоцій. Він містить понад 35,000 чорно-білих зображень розміром 48x48 пікселів, що зображають обличчя людей із різними емоційними виразами. FER2013 призначений для навчання моделей розпізнавання емоцій за допомогою аналізу виразів обличчя. Набір даних класифікує емоції на такі категорії: щастя, сум, подив, страх, злість, відроза та нейтральний вираз. Він користується популярністю завдяки своїй доступності, однак низька роздільна здатність зображень і незбалансованість даних через переважання емоції "щастя" можуть створювати труднощі в точності моделей [8].



Рисунок 1.9 – Демонстрація частини вмісту набору даних FER2013

AffectNet (див. Рис. 1.10) – це один із найбільших наборів даних для розпізнавання емоцій, що містить понад мільйон зображень обличчя. Зображення зібрані за допомогою пошукових систем на основі емоційних запитів. AffectNet є універсальним інструментом для аналізу емоцій, оскільки охоплює широкий спектр реальних умов і містить зображення з різних джерел. Емоції в цьому наборі представлені такими категоріями: щастя, сум, подив, страх, злість, відраза, нейтральний вираз і презирство. Хоча цей набір даних має величезний обсяг і різноманітність, автоматичне анотування часто призводить до появи шуму, що може вплинути на якість навчання моделей [9].

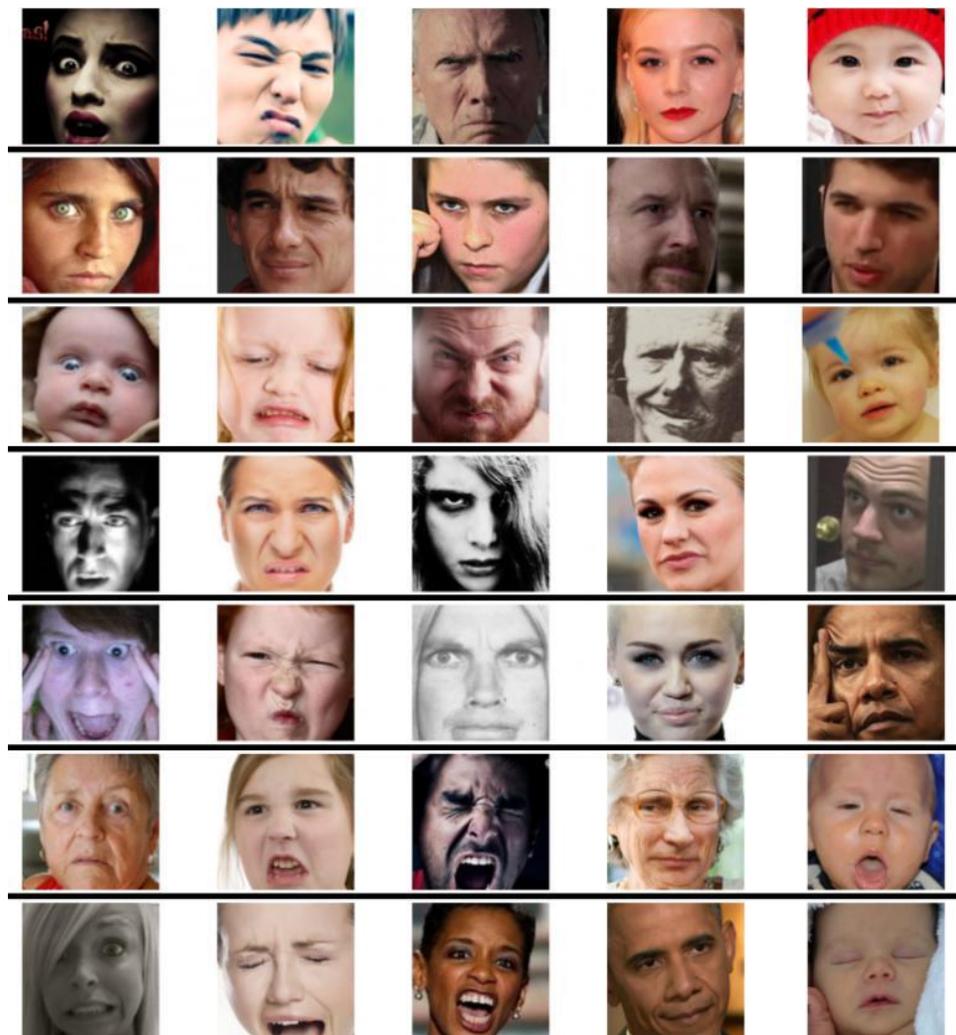


Рисунок 1.10 – Демонстрація частини вмісту набору даних AffectNet

СК+ (The Extended Cohn-Kanade Dataset) (див. Рис 1.11) є розширеною версією класичного набору даних Cohn-Kanade, який широко використовується для аналізу виразів обличчя. Він містить 593

відеопослідовності з емоційними реакціями 123 учасників. СК+ відомий своєю здатністю демонструвати динамічні емоції, що робить його корисним для дослідження змін виразу обличчя в часі. Емоції в цьому наборі даних представлені такими категоріями: щастя, сум, подив, страх, злість, відраза та презирство. Висока якість зображень і можливість аналізу динаміки емоцій є його основними перевагами, хоча інсценовані емоції можуть відрізнятися від природних, і розмір набору є досить обмеженим [10].



Рисунок 1.11 – Демонстрація частини вмісту набору даних СК+

Набір даних RAF-DB (Real-world Affective Faces Database) (див. Рис 1.12) створений для аналізу емоцій у реальних умовах. Він містить понад 29,000 зображень із різними виразами обличчя, зібраних з Інтернету. Цей набір даних дозволяє проводити дослідження на основі реальних емоційних виразів, що з'являються у повсякденному житті. Категорії емоцій включають щастя, сум, подив, страх, злість, відразу та нейтральний вираз. RAF-DB є корисним для вивчення емоцій у непередбачуваних умовах, однак автоматизований збір даних може призвести до появи неточних анотацій [10].



Рисунок 1.12 – Демонстрація частини вмісту набору даних RAF-DB

JAFFE (Japanese Female Facial Expression) (див. Рис 1.13) є класичним набором даних, який містить 213 зображень обличчя японських жінок із різними емоційними виразами. Він був створений спеціально для тестування моделей розпізнавання емоцій і широко використовується в академічних дослідженнях. Категорії емоцій у JAFFE включають щастя, сум, подив, страх, злість, відраза та нейтральний вираз. Основною перевагою JAFFE є висока якість зображень, що робить його ідеальним для детального аналізу виразів обличчя. Однак набір є обмеженим як за кількістю зображень, так і за репрезентативністю, оскільки він охоплює лише японських жінок [10].

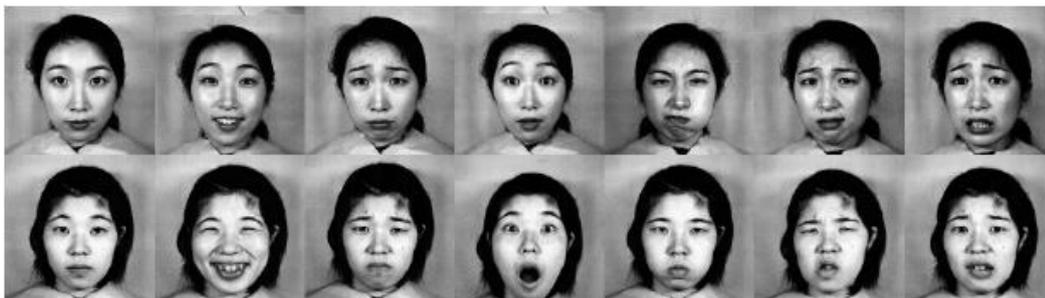


Рисунок 1.13 – Демонстрація частини вмісту набору даних JAFFE

Набір даних EmoReact (див. Рис 1.14) створений для вивчення емоційних реакцій дітей на різний відеоконтент і містить понад 110,000 кадрів із відео. Його основною метою є аналіз емоцій, які виникають під час перегляду мультимедіа. Зображення в цьому наборі представляють реакції на такі емоції, як радість, сум, захоплення, злість, нейтральний стан, здивування, відраза та

страх. Цей набір даних дозволяє вивчати реальні емоції, що виникають у дітей під час перегляду відео [10].



Рисунок 1.14 – Демонстрація частини вмісту набору даних EmoReact

Ці набори даних пропонують широкий вибір для різних завдань у сфері розпізнавання емоцій і підходять для досліджень як статичних, так і динамічних виразів облич. Порівняння їх показано у вигляді Таблиці 1.2 [11].

Таблиця 1.2 – Порівняння наборів даних для розпізнавання облич

| Набір даних | Кількість зображень | Емоції | Переваги | Недоліки |
|-------------|---------------------|-------------------------------------------------------------------|--------------------------------------------------------------------|-----------------------------------------------------------------------|
| FER2013 | 35,887 | Щастя, Сум, Подив, Страх, Злість, Відраза, Нейтральна | Простий у використанні; доступний на Kaggle | Низька роздільна здатність; незбалансованість даних |
| AffectNet | Понад 1 мільйон | Щастя, Сум, Подив, Страх, Злість, Відраза, Нейтральна, Презирство | Великий обсяг даних; широкий діапазон емоцій, включаючи презирство | Наявність шуму через автоматизований збір даних |
| CK+ | 593 послідовності | Щастя, Сум, Подив, Страх, Злість, Відраза, Презирство | Висока якість зображень; можливість аналізу динамічних емоцій | Невеликий розмір; інсценовані емоції можуть відрізнитися від реальних |
| RAF-DB | Понад 29,000 | Щастя, Сум, Подив, Страх, Злість, Відраза, Нейтральна | Реальні умови; більше варіацій у виразах | Можливі помилки в анотаціях |
| JAFFE | 213 | Щастя, Сум, Подив, Страх, Злість, Відраза, Нейтральна | Висока якість зображень; стандарт для порівняння | Обмежена кількість зображень; лише японські жінки |

1.4 Основи комп'ютерного зору

За своєю суттю, комп'ютерний зір – це здатність комп'ютерів розуміти та аналізувати візуальний контент так само, як це робить людина. Це включає такі завдання, як розпізнавання об'єктів і облич, читання тексту і розуміння контексту зображення або відео.

Комп'ютерний зір тісно пов'язаний зі штучним інтелектом (ШІ) і часто використовує методи ШІ, такі як машинне навчання, для аналізу та розуміння візуальних даних. Алгоритми машинного навчання використовуються для «навчання» комп'ютера розпізнавати шаблони та особливості візуальних даних, такі як краї, форми та кольори.

Після навчання комп'ютер може використовувати ці знання для ідентифікації та класифікації об'єктів на нових зображеннях і відео. Точність цих класифікацій можна покращити з часом завдяки подальшому навчанню та доступу до більшої кількості даних.

На додаток до машинного навчання, комп'ютерний зір може також використовувати такі методи, як глибоке навчання, яке передбачає навчання штучних нейронних мереж на великих обсягах даних для розпізнавання шаблонів і особливостей у спосіб, подібний до того, як працює людський мозок.

Історія комп'ютерного зору налічує понад 60 років, а перші спроби зрозуміти, як людський мозок обробляє візуальну інформацію, призвели до розробки технології сканування зображень у 1959 році. У 1960-х роках штучний інтелект став академічною галуззю досліджень, а комп'ютери почали перетворювати двовимірні зображення у тривимірні форми.

У 1970-х роках була розроблена технологія оптичного розпізнавання символів, що дозволила комп'ютерам розпізнавати текст, надрукований будь-яким шрифтом або гарнітурою. За цим послідувала розробка інтелектуального розпізнавання символів, яке могло розшифровувати рукописний текст за допомогою нейронних мереж. Реальне застосування цих технологій включає

обробку документів і рахунків, розпізнавання автомобільних номерів, мобільні платежі та машинний переклад.

У 1980-х роках нейробіолог Девід Марр встановив, що зір працює ієрархічно, і представив алгоритми для машин для виявлення країв, кутів, кривих та інших базових форм. У той же час комп'ютерний вчений Куніхіко Фукусіма розробив мережу клітин під назвою «Неокогнітрон», яка могла розпізнавати патерни, в тому числі згорткові шари в нейронній мережі (див. Рис. 1.15) [12].

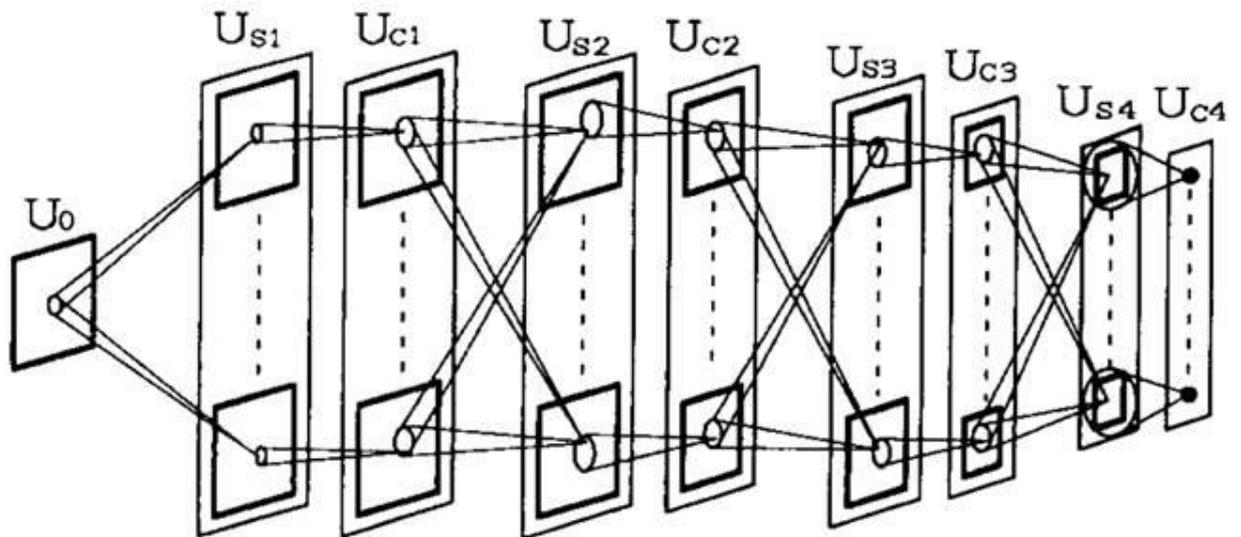


Рисунок 1.15 – Схема мережі клітин «Неокогнітрон»

У 1990-х і 2000-х роках з'явилися програми для розпізнавання облич у реальному часі, а також відбулася стандартизація тегування та анотування наборів візуальних даних. У 2010 році став доступним набір даних ImageNet, що містить мільйони тегованих зображень у тисячах класів об'єктів і є основою для згорткових нейронних мереж (CNN) та моделей глибокого навчання, які використовуються сьогодні.

У 2012 році модель AlexNet зробила прорив у розпізнаванні зображень, знизивши рівень помилок до кількох відсотків. Ці розробки проклали шлях до широкого використання комп'ютерного зору в різних сферах застосування.

Система комп'ютерного зору складається з двох основних компонентів: сенсорного пристрою, такого як камера, і інтерпретуючого пристрою, такого

як комп'ютер. Сенсорний пристрій фіксує візуальні дані з навколишнього середовища, а інтерпретуючий пристрій обробляє ці дані для вилучення сенсу.

Алгоритми комп'ютерного зору ґрунтуються на гіпотезі, що «наш мозок покладається на шаблони для декодування окремих об'єктів». Подібно до того, як наш мозок обробляє візуальні дані, шукаючи закономірності у формах, кольорах і текстурах об'єктів, алгоритми комп'ютерного зору обробляють зображення, шукаючи закономірності в пікселях, що складають зображення. Ці закономірності можна використовувати для ідентифікації та класифікації різних об'єктів на зображенні.

Щоб проаналізувати зображення, алгоритм комп'ютерного зору спочатку перетворює його на набір числових даних, які можуть бути оброблені комп'ютером. Зазвичай це робиться шляхом поділу зображення на сітку маленьких одиниць, які називаються пікселями, і представлення кожного пікселя набором числових значень, які описують його колір і яскравість. Ці значення можуть бути використані для створення цифрового представлення зображення, яке може бути проаналізоване комп'ютером.

Після того, як зображення перетворено на числові дані, алгоритм комп'ютерного зору може почати його аналізувати. Зазвичай це передбачає використання методів машинного навчання та штучного інтелекту для розпізнавання закономірностей у даних і прийняття рішень на основі цих закономірностей. Наприклад, алгоритм може аналізувати значення пікселів на зображенні, щоб ідентифікувати краї об'єктів або розпізнавати специфічні візерунки чи текстури, характерні для певних типів об'єктів.

Загалом, мета комп'ютерного зору – дозволити комп'ютерам аналізувати і розуміти візуальні дані так само, як це роблять людський мозок і очі, і використовувати це розуміння для прийняття інтелектуальних рішень на основі цих даних.

Комп'ютерний зір забезпечив численні технологічні переваги в різних галузях і сферах застосування. Одним із прикладів є використання ІВМ комп'ютерного зору для створення програми «Мої моменти» для турніру з

гольфу Masters 2018 року (див. Рис. 1.16). Ця програма використовувала комп'ютерний зір для аналізу відеозаписів турніру в реальному часі та визначення ключових моментів, таких як вдалі удари або визначні події. Потім ці моменти були відібрані і доставлені вболівальникам у вигляді персоналізованих відеороликів, що дозволило їм легко відстежувати турнір і залишатися залученими до події [13].

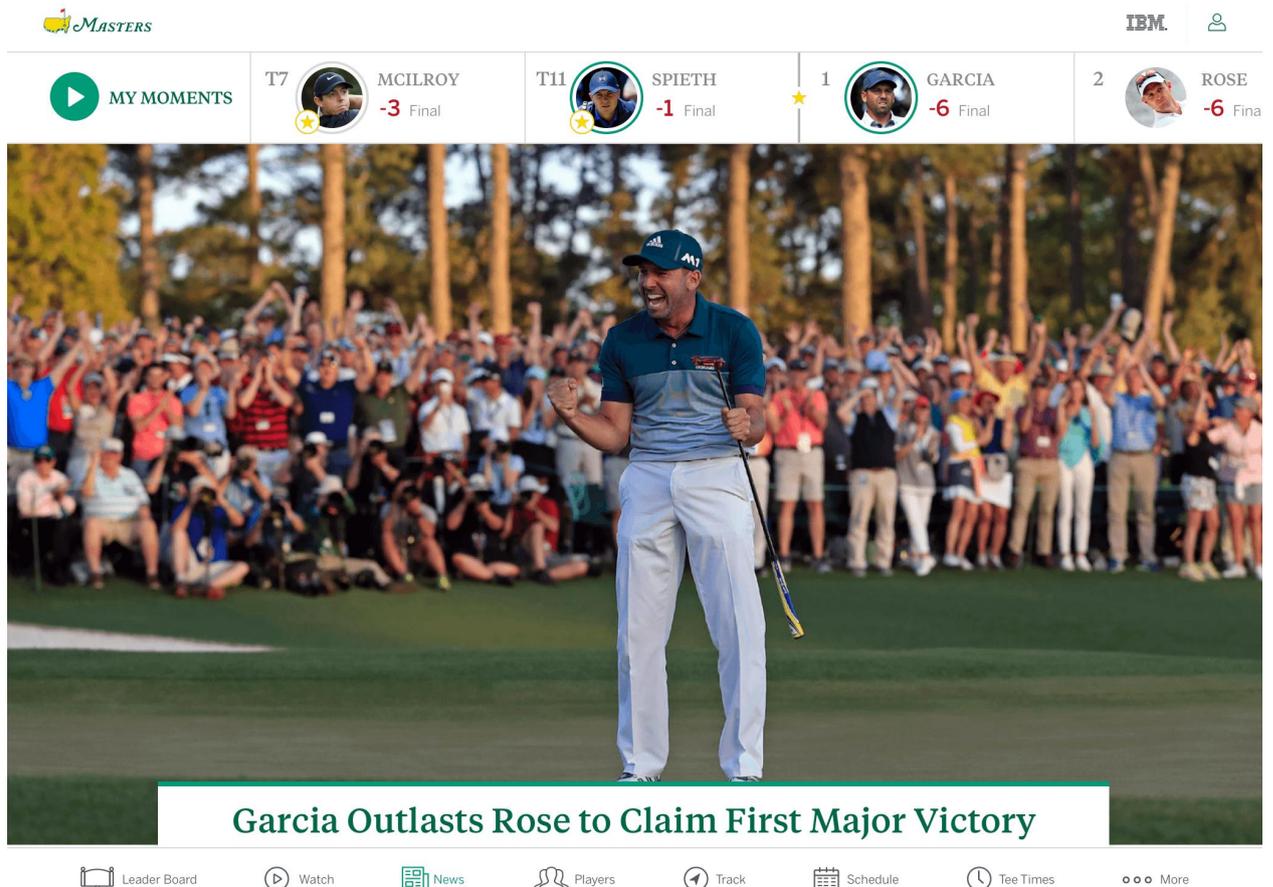


Рисунок 1.16 – Програма «Мої моменти» для турніру з гольфу Masters 2018 року

Тематичні парки Діснея також використовують комп'ютерний зір і технологію прогнозування ШІ для покращення своєї роботи. Технологія працює з високотехнологічними датчиками, допомагаючи підтримувати безперебійну роботу атракціонів з мінімальними перебоями. Наприклад, якщо в атракціоні виникають технічні проблеми, система може передбачити проблему і автоматично відправити технічний персонал для її вирішення,

допомагаючи підтримувати безперебійну роботу атракціону і запобігаючи збоям для гостей.

Google Translate – ще один приклад використання комп'ютерного зору в технологіях. Ця програма використовує камеру смартфона і алгоритми комп'ютерного зору для аналізу і перекладу тексту на зображеннях, таких як вивіски або документи іноземними мовами. Це дозволяє користувачам легко перекладати текст на ходу, полегшуючи спілкування і навігацію в незнайомому середовищі.

Нарешті, IBM і Verizon спільно працюють над тим, щоб допомогти автомобільним компаніям виявляти дефекти автомобілів ще до того, як вони виїжджають з заводу. Використовуючи комп'ютерний зір та інші передові технології, вони розробляють системи, які можуть аналізувати якість компонентів транспортних засобів і виявляти дефекти в режимі реального часу, що дозволяє компаніям виявляти і виправляти проблеми до того, як вони стануть більш серйозними. Це може допомогти підвищити якість і безпеку транспортних засобів, а також знизити виробничі витрати за рахунок виявлення проблем на ранніх стадіях виробничого процесу.

Комп'ютерний зір має широкий спектр можливостей і застосувань у різних галузях. Ось кілька прикладів можливостей комп'ютерного зору, а також короткі пояснення до кожного з них:

- Оптичне розпізнавання символів (OCR): здатність розпізнавати і витягувати текст із зображень або відсканованих документів;
- Інспекція машин: використання комп'ютерного зору для перевірки та оцінки якості або стану різних компонентів або продуктів;
- Роздрібна торгівля: використання комп'ютерного зору в автоматизованих касових системах та інших додатках для роздрібною торгівлі, таких як управління запасами і відстеження клієнтів;
- Побудова 3D-моделей: використання комп'ютерного зору для аналізу декількох зображень об'єкта або середовища і побудови його 3D-моделі;

- Медична візуалізація: використання комп'ютерного зору для аналізу медичних зображень, таких як рентгенівські знімки або комп'ютерна томографія, щоб допомогти в діагностиці та лікуванні пацієнтів;
- Автомобільна безпека: використання комп'ютерного зору в системах допомоги водієві та автономних транспортних засобах для виявлення і реагування на перешкоди та інші небезпеки на дорозі;
- Match move: використання комп'ютерного зору для вирівнювання і злиття CGI-елементів з кадрами в реальному часі у фільмах та інших візуальних ефектах;
- Захоплення руху: використання комп'ютерного зору для захоплення та аналізу руху акторів або інших об'єктів, як правило, для використання в анімації або додатках віртуальної реальності;
- Спостереження: використання комп'ютерного зору для аналізу відеоматеріалів з метою безпеки та моніторингу;
- Розпізнавання відбитків пальців і біометрія: використання комп'ютерного зору для аналізу і розпізнавання унікальних фізичних характеристик, таких як відбитки пальців, для перевірки особи та інших застосувань [14].

Проблеми комп'ютерного зору також існують до сьогодні. Комп'ютерний зір – це складна галузь, яка пов'язана з багатьма викликами і труднощами. Деякі з цих проблем включають:

Обмеження даних: комп'ютерний зір вимагає великих обсягів даних для навчання і тестування алгоритмів. Це може бути проблематично в ситуаціях, коли дані обмежені або конфіденційні, і можуть не підходити для обробки в хмарі. Крім того, масштабування обробки даних може бути дорогим і може бути обмежене апаратними та іншими ресурсами.

Швидкість навчання: ще однією проблемою комп'ютерного зору є час і ресурси, необхідні для навчання алгоритмів. Хоча кількість помилок з часом зменшилася, вони все ще трапляються, і потрібен час, щоб навчити комп'ютер розпізнавати і класифікувати об'єкти і патерни на зображеннях. Цей процес

зазвичай включає в себе надання наборів мічених зображень і порівняння їх з прогнозованою вихідною міткою або вимірами розпізнавання, а потім модифікацію алгоритму для виправлення будь-яких помилок.

Вимоги до апаратного забезпечення: алгоритми комп'ютерного зору є складними в обчислювальному плані, що вимагає швидкої обробки та оптимізованої архітектури пам'яті для швидкого доступу до неї. Правильно сконфігуровані апаратні системи та програмні алгоритми також необхідні для забезпечення безперебійної та ефективної роботи програм обробки зображень.

Складність, притаманна візуальному світу: у реальному світі об'єкти можна бачити під різними кутами і в незліченних умовах освітлення, а в справжній системі технічного зору існує нескінченна кількість можливих сцен. Ця притаманна складність ускладнює створення універсальної «зорової машини», яка може впоратися з усіма можливими візуальними сценаріями.

Загалом, ці виклики підкреслюють той факт, що комп'ютерний зір є складною і комплексною галуззю, і що попереду ще багато роботи для того, щоб створити машини, які зможуть бачити і розуміти світ так само, як це робить людина [15].

РОЗДІЛ 2 ЗАСТОСУВАННЯ ГЛИБОКОГО НАВЧАННЯ У РОЗПІЗНАВАННІ ЕМОЦІЙ

2.1 Алгоритми глибокого навчання

Глибоке навчання – це частина машинного навчання і одна з передових технологій штучного інтелекту. Його завданням є імітація процесу навчання людини – навчання на прикладі.

Для науковців, які працюють з даними, глибинне навчання є незамінним, оскільки воно полегшує та прискорює збір, аналіз та обробку великих обсягів даних. Хоча ця передова технологія звучить так, ніби вона належить майбутньому, вона присутня у повсякденному використанні; наприклад, вона дозволяє самокерованим автомобілям розпізнавати дорожні знаки і коригувати свій стиль водіння або відрізнити пішоходів від дерев на узбіччі дороги.

Щоб зрозуміти, як глибинне навчання здобуває нові знання, візьмемо простий приклад: уявімо малюка, який вивчає світ і незнайомі об'єкти навколо себе. Малюк показує на речі і називає їх, а батьки підтверджують або заперечують його спроби. Скажімо, дитина називає кружку – батьки відповідають: «так, це кружка» або «ні, це не кружка».

Кожного разу, коли малюк намагається назвати предмет, незалежно від того, правильно чи ні, він все більше усвідомлює його ознаки, на основі яких він може відрізнити його від інших речей. Таким чином дитина вибудовує складний абстрактний образ поняття, вибудовуючи рівні ієрархії з кожним шаром набутих знань.

Терміни «машинне навчання», «глибинне навчання» та «нейронні мережі» іноді використовують як взаємозамінні, і хоча вони пов'язані між собою, вони не є тотожними.

Розглянемо основні відмінності між цими трьома технологіями:

Машинне навчання – це підмножина штучного інтелекту, яка фокусується на використанні даних і алгоритмів для імітації способів, якими людина вчиться створювати машини, що «навчаються» і роблять прогнози з мінімальним втручанням людини. Це дає змогу комп'ютерам працювати автономно, самостійно навчатися та розвиватися.

Глибоке навчання – це частина машинного навчання. Воно встановлює складні кореляції між даними і вчиться на прикладах і попередніх помилках. Глибоке навчання вимагає більших обсягів даних порівняно з машинним навчанням.

Хоча навчання триває довше, його точність вища, ніж у машинного навчання. Глибинне навчання також може навчатися самостійно, вимагаючи менше людської взаємодії, ніж машинне навчання, але набори даних не обов'язково маркувати – алгоритми можуть навчатися на «сирих» неструктурованих даних.

Нейронні мережі – це метод штучного інтелекту, який навчає комп'ютери обробляти дані так, як це робить людський мозок. Вони складаються з шарів нейронів або вузлів, які пов'язані між собою і можуть отримувати дані до певного значення, після чого передають їх на наступний вузол для обробки.

Термін «глибокий» в глибинному навчанні відноситься до кількості вузлів у нейронних мережах. Якщо їх більше трьох, крім входу і виходу, ми говоримо про алгоритм глибокого навчання – глибоку нейронну мережу. Якщо нейронна мережа складається лише з трьох вузлів, це базова нейронна мережа.

Глибинне навчання використовує алгоритми, які намагаються зробити висновки, подібні до тих, які робить людина, аналізуючи дані з логічною структурою. Для цього глибинне навчання використовує нейронні мережі, багат шарові структури алгоритмів, які можуть виявляти закономірності та класифікувати інформацію, так само як це робить людина.

Коли наш мозок отримує нову інформацію, він намагається порівняти її з тим, що ми знаємо. Той самий принцип застосовується в глибокому навчанні з використанням нейронних мереж.

Все, що потрібно програмам глибокого навчання, – це доступ до великих обсягів неструктурованих і немаркованих даних, і вони самостійно навчатимуться і створюватимуть складні, точні моделі прогнозування.

Наприклад, модель глибокого навчання отримує набір зображень, що містять кружки. Її алгоритми навчатимуться на основі пікселів, що містяться на зображеннях, до яких вона отримала доступ, класифікуватимуть групи пікселів і створюватимуть набір ознак, тобто предиктивну модель, яка розпізнає всі зображення, включно з горнятками. З кожною ітерацією програма набуває більше знань і точності.

Якщо в машинному навчанні експерту довелося б витратити значний час на розробку системи машинного навчання, здатної розпізнавати зображення чашок, то в глибокому навчанні достатньо надати велику кількість зображень із зображенням чашок, і система навчиться самостійно. Важливо зазначити, що комп'ютерні програми, які використовують глибоке навчання, є високошвидкісними: вони створюють набори ознак і сортують мільйони даних всього за кілька хвилин [16].

Завдяки вдосконаленим алгоритмам глибоке навчання може аналізувати та сортувати будь-які набори даних. Ось деякі з найкращих алгоритмів глибокого навчання:

1. Класичні нейронні мережі

Класичні нейронні мережі, які також називають повністю зв'язаними нейронними мережами, використовуються для обробки зображень і виявлення об'єктів. Класичні нейронні мережі складаються з багат шарових перцептронів, нейрони яких з'єднані в безперервну мережу. Вони включають адаптацію моделі до первинних двійкових даних за допомогою трьох функцій, а саме:

- Лінійна функція: має одну лінію, яка множить вхідні дані на постійний множник;
- Нелінійна функція: Далі поділяється на три підмножини:
 1. Сигмоїдна крива: S-подібна крива з діапазоном від 0 до 1.
 2. Гіперболічний тангенс: S-подібна крива з діапазоном від -1 до 1.
 3. Випрямлена лінійна одиниця (ReLU): Одноточкова функція, яка повертає 0, коли вхідне значення менше заданого, і повертає лінійне кратне, якщо вхідне значення більше заданого.

2. Згорткові нейронні мережі

Згорткова нейронна мережа або CNN – це класична модель штучної нейронної мережі з високим потенціалом для вирішення складних завдань та аналізу зображень і не-зображень. Вона заснована на тому ж принципі, що і нейрони в корі головного мозку тварин.

CNN поділяються на чотири шари:

1. Один вхідний шар містить двовимірне розташування нейронів для аналізу вмісту зображення;
2. Вихідний шар нейронів одновимірний, а зображення обробляються на вхідних вузлах за допомогою згорткових шарів;
3. Третій шар складається з прихованих вузлів і називається шаром вибірки, який визначає кількість нейронів, що беруть участь в ідентифікації вмісту даних;
4. Таким чином, ШНМ можуть мати один або більше прихованих шарів між входом і виходом, що з'єднують шари дискретизації з вихідними шарами.

3. Рекурентні нейронні мережі

ШНМ використовуються для прогнозування послідовностей, і в цьому процесі вони використовують знання з попереднього стану як вхідні дані для майбутніх прогнозів. ШН бувають двох видів:

- LSTM: Прогнозує дані в часових послідовностях, використовуючи набуту пам'ять. Має три входи: вхід, вихід і забуття.
- Шлейфові ШНМ: Ця форма також прогнозує дані часових послідовностей через пам'ять, але має два входи – оновлення та скидання.

4.Обмежені машини Больцмана

Цей алгоритм не має наперед визначеного напрямку, тому його вузли з'єднані по колу.Завдяки своїй унікальній формі він використовується для отримання параметрів моделі.

Модель машини Больцмана є стохастичною, оскільки має випадковий розподіл ймовірностей, який можна проаналізувати статистично, але не можна точно передбачити.

5.Трансферне навчання

Трансферне навчання – це процес вдосконалення моделі, яка вже була вивчена для виконання нових і більш точних завдань. Воно може бути успішним лише в тому випадку, якщо характеристики моделі з початкового дослідження є загальними. Одним з найпопулярніших підходів до глибокого навчання є навчання з перенесенням, яке вимагає меншого обсягу даних, ніж інші, а отже, потребує менше часу на обробку даних [17].

2.2 Передобробка зображень

Передобробка зображень є важливим етапом у розробці моделей розпізнавання емоцій, оскільки якість вхідних даних суттєво впливає на точність результатів. Перед тим як навчати модель, зображення необхідно очистити та підготувати за допомогою ряду методів. Основні з них включають очищення, фільтрацію, вирівнювання та нормалізацію зображень.

Очищення зображень полягає у видаленні зайвих елементів, шумів і артефактів, які можуть спотворити результати моделі. Шум у зображеннях може виникати через недосконалість апаратного забезпечення або через умови зйомки. Зазвичай, для очищення зображень застосовують різні фільтри, такі

як середній фільтр або гаусівський фільтр. Це допомагає зменшити рівень шуму і зробити зображення більш однорідними.

Фільтрація використовується для усунення шуму та покращення якості зображення без втрати важливих деталей. Існують різні типи фільтрів:

- Гаусівський фільтр: розмиває зображення для зменшення шуму, зберігаючи основні елементи;
- Медіанний фільтр: застосовується для усунення спотворень, таких як спалахи або артефакти, викликані сенсорами;
- Більший фільтр (high-pass): використовується для підсилення контурів і країв об'єктів на зображенні, що допомагає моделям краще ідентифікувати межі обличчя.

Вирівнювання зображень, або гама-корекція, допомагає підвищити контраст і рівномірність освітлення на зображенні. Це особливо важливо, якщо зображення зроблені за різних умов освітлення або містять тіні. Часто використовують методи вирівнювання гистограми, які дозволяють розширити діапазон інтенсивності пікселів на зображенні, роблячи важливі деталі більш видимими. Вирівнювання забезпечує, щоб зображення мали однаковий рівень освітлення та контрасту, що робить їх більш придатними для машинного навчання.

Нормалізація зображень – це процес приведення піксельних значень до стандартного діапазону, зазвичай від 0 до 1 або від -1 до 1. Це зменшує вплив різниці у яскравості та кольорах на результати моделі. Нормалізація дозволяє моделі навчатися від реальних емоційних виразів, не реагуючи на такі фактори, як надто світлі чи темні ділянки зображення. Вона також допомагає покращити швидкість і стабільність навчання, оскільки значення пікселів стають узгодженими по всьому набору даних [18].

Аугментація даних – це метод, який використовується для штучного збільшення розміру вашого набору даних шляхом створення нових зображень з існуючих. Це допомагає зменшити надмірну підгонку і покращує

узагальнення вашої моделі. Деякі поширені методи доповнення даних зображень включають

Перегортання та обертання: просте перевертання (горизонтальне чи вертикальне) або обертання (на 90, 180, 270 градусів) зображень може створити нові точки даних. Наприклад, якщо у вас є 1 000 зображень котів, їх перевертання та обертання дасть вам 4 000 зображень (1 000 оригінальних + 1 000 перевернутих по горизонталі + 1 000 перевернутих по вертикалі + 1 000 обернутих на 90 градусів).

Обрізання зображень до різних розмірів і співвідношень створює нові зображення з того самого оригіналу. Таким чином, ваша модель отримує різні обрамлення і композиції одного і того ж вмісту. Ви можете створювати випадкові обрізки різного розміру або задавати більш конкретні співвідношення обрізки, наприклад, квадрати.

Маніпуляції з кольором: регулювання яскравості, контрасту, відтінку і насиченості – прості способи створення нових доповнених зображень. Наприклад, ви можете довільно змінити яскравість і контрастність зображень на 30%, щоб створити нові точки даних. Будьте обережні, щоб не спотворити зображення занадто сильно, інакше ви ризикуєте заплутати свою модель.

Накладання зображень: накладання прозорих зображень, текстур або шуму на існуючі зображення – ще одна проста техніка доповнення. Додавання таких елементів, як водяні знаки, логотипи, бруд/подряпини або гаусівський шум, може створити реалістичні варіації ваших вихідних даних. Почніть з ледь помітних накладок і подивіться, як відреагує ваша модель.

Поєднання технік: для максимального збільшення обсягу даних ви можете комбінувати кілька технік доповнення на одних і тих самих зображеннях. Наприклад, ви можете перевертати, обертати, обрізати і змінювати колір зображень, створюючи багато нових точок даних з одного оригінального зображення. Але будьте обережні, щоб не переборщити з доповненням, інакше ви ризикуєте спотворити зображення до невпізнання!

Використовуючи доповнення даних, ви можете легко збільшити розмір вашого набору зображень у 4, 10 і більше разів, і все це без збору нових зображень. Це допомагає боротися з надмірною підгонкою і підвищує точність моделі, при цьому час і вартість навчання залишаються незмінними [18].

Таким чином, передобробка зображень відіграє критичну роль у підготовці даних для моделей розпізнавання емоцій. Очищення, фільтрація, вирівнювання та нормалізація покращують якість зображень і допомагають усунути шуми та артефакти, роблячи дані придатними для аналізу та підвищуючи точність моделі. Від правильного вибору методів передобробки залежить не тільки якість навчання моделі, але й її здатність точно розпізнавати емоції на обличчях .

2.3 Виявлення обличчя

Виявлення обличчя є першочерговим кроком у багатьох завданнях комп'ютерного зору, таких як розпізнавання емоцій, ідентифікація осіб чи аналіз виразів обличчя. Для цього використовуються різні алгоритми та підходи, які дозволяють ідентифікувати обличчя на зображенні з високою точністю. Серед найбільш поширених методів – це метод Хаара (Haar cascades), глибокі нейронні мережі та бібліотека dlib. Кожен із них має свої переваги та недоліки, що визначають їх застосування у різних завданнях.

Метод Хаара (Haar cascades) (див. Рис. 2.1): метод Хаара є класичним підходом до виявлення обличчя, розробленим на основі каскадних класифікаторів Хаара. Цей алгоритм аналізує зображення на основі набору простих ознак, які порівнюються з попередньо навченою моделлю. Основна ідея полягає у використанні прямокутних фільтрів, які обчислюють різницю між яскравими та темними областями на зображенні, що дозволяє визначити контури обличчя.

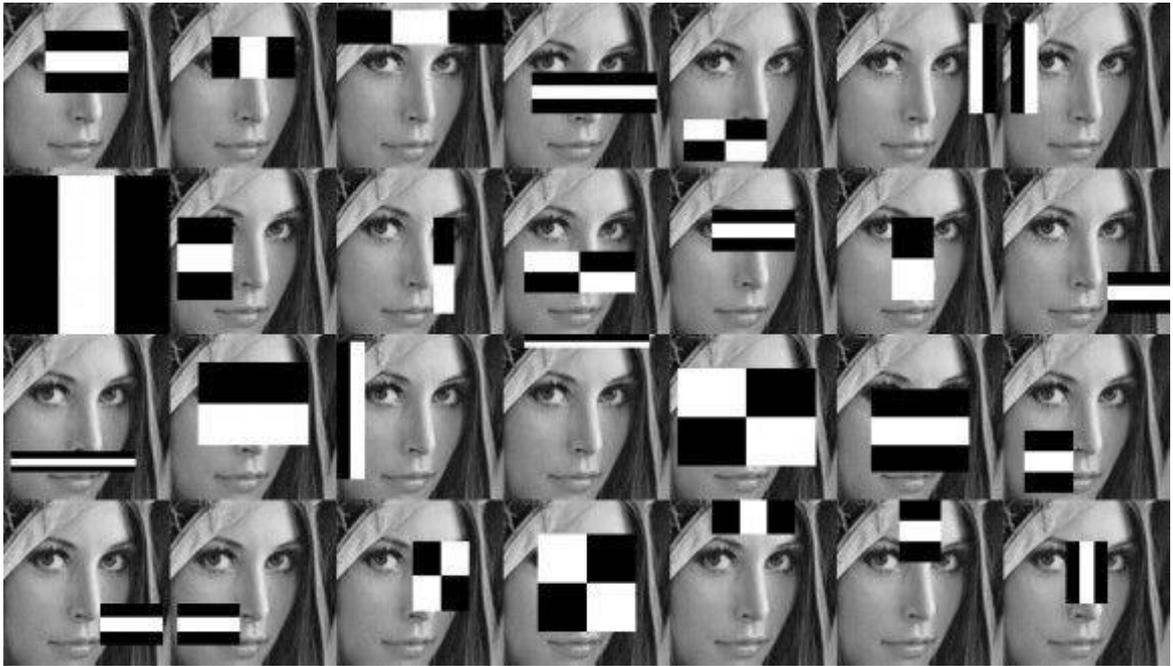


Рисунок 2.1 – Демонстрація Haar cascades

Перевагами цього методу є його швидкодія та відносна простота. Він особливо корисний для додатків у реальному часі, де необхідна швидка обробка зображень. Однак метод Хаара є чутливим до змін освітлення і положення обличчя, що може призвести до помилкових результатів або пропуску облич [19].

Глибокі нейронні мережі (Deep Neural Networks, DNN) (див. Рис. 2.2): глибокі нейронні мережі є більш сучасним і потужним інструментом для виявлення облич. Вони використовують багат шарову архітектуру, яка дозволяє моделі навчатися складних ознак зображень. На відміну від методів, що базуються на жорстко закодованих ознаках, нейронні мережі можуть автоматично визначати найбільш релевантні ознаки обличчя, навіть якщо воно розташоване під кутом або частково закрите.

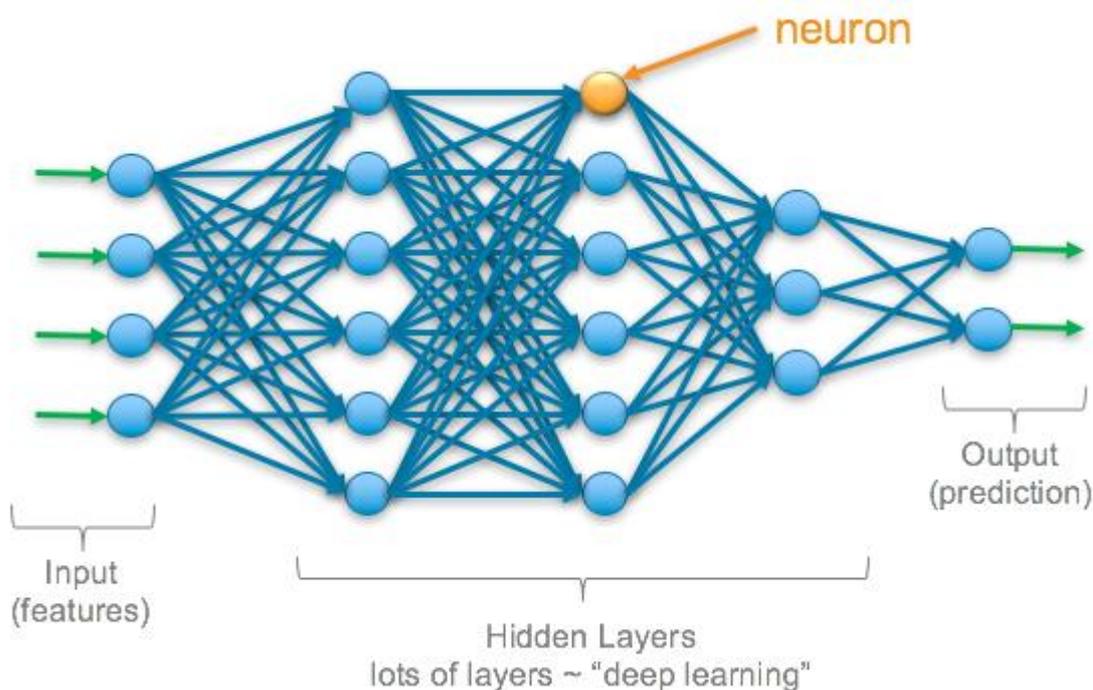


Рисунок 2.2 – Демонстрація загальної схеми Deep Neural Networks

Застосування глибоких нейронних мереж для виявлення обличчя відзначається високою точністю та стійкістю до змін умов, таких як освітлення, масштаб і кут нахилу обличчя. Проте вони потребують значних обчислювальних ресурсів, що може бути проблематичним у додатках з обмеженими можливостями. Крім того, тренування таких моделей вимагає великого обсягу даних і часу [20].

Одним із популярних підходів є використання моделей на основі архітектури CNN (Convolutional Neural Networks), таких як MTCNN (Multi-task Cascaded Convolutional Networks) або Faster R-CNN, які забезпечують чудові результати в детекції облич на зображеннях з високою роздільною здатністю [21].

Бібліотека dlib (див. Рис. 2.3): є інструментом для обробки зображень і виявлення облич на основі алгоритмів машинного навчання. Вона містить переднавчені моделі для виявлення облич, які використовують сучасні методи, зокрема Histogram of Oriented Gradients (HOG) та глибокі нейронні мережі. dlib дозволяє ідентифікувати обличчя з високою точністю і є ефективною для різних завдань, включаючи розпізнавання осіб та аналіз виразів обличчя.

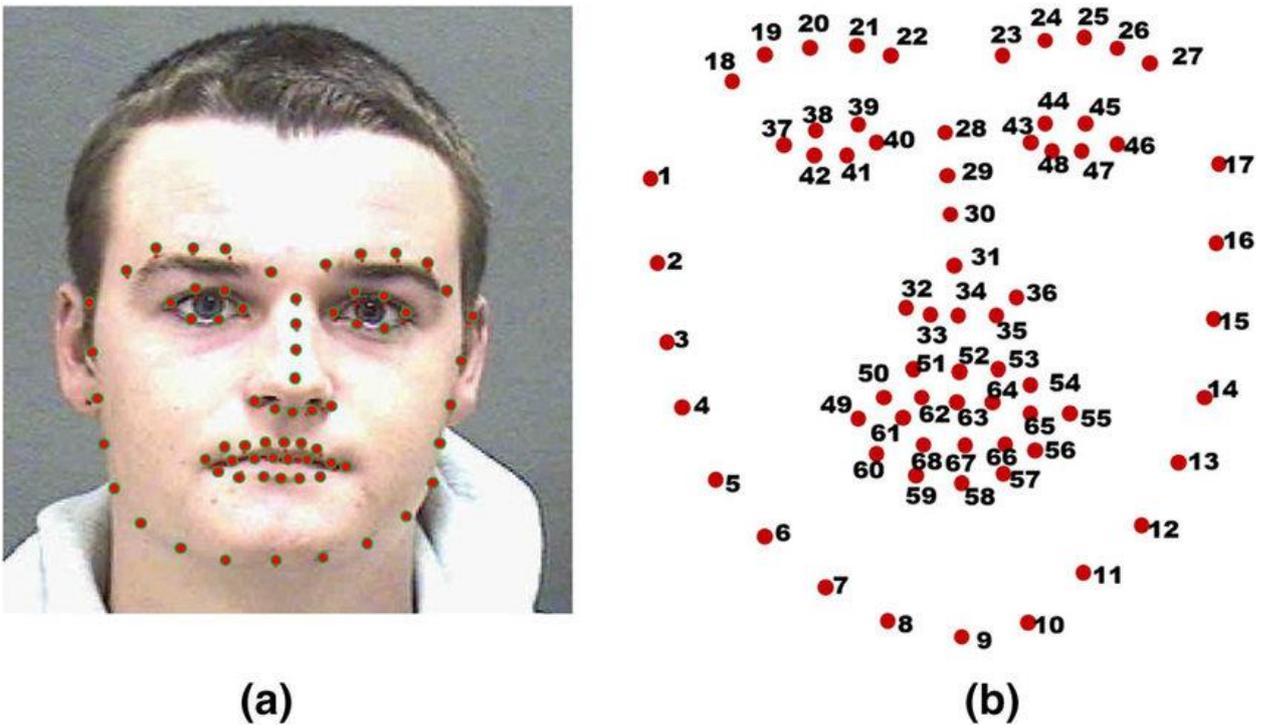


Рисунок 2.3 – Демонстрація бібліотеки dlib

dlib добре підходить для завдань, де потрібна висока точність і швидкодія, а також для програм, які можуть працювати як у реальному часі, так і офлайн. Важливою перевагою цієї бібліотеки є її універсальність – вона підтримує не лише детекцію обличчя, а й трекінг, розпізнавання осіб та інші функції. Бібліотека також популярна через свою відкритість та доступність для широкого кола дослідників і розробників [22].

Порівняння методів

- Haar cascades підходять для простих додатків у реальному часі, проте є чутливими до умов зйомки та положення обличчя;
- Глибокі нейронні мережі забезпечують високу точність і гнучкість для роботи з різними типами зображень, але вимагають більше обчислювальних ресурсів і даних для навчання;
- dlib – це ефективне рішення для задач детекції облич з високою точністю та продуктивністю, особливо коли необхідно використовувати методи трекінгу або розпізнавання облич.

Кожен з цих методів має свої переваги залежно від вимог проекту, обчислювальних можливостей та особливостей вхідних даних.

2.4 Класифікація емоцій нейронними мережами

Класифікація емоцій нейронними мережами є ключовим етапом у системах розпізнавання емоцій, які використовують обробку зображень обличчя для виявлення та ідентифікації емоційних станів. Цей процес базується на аналізі виявлених рис обличчя та їхньому подальшому класифікуванні за допомогою глибоких нейронних мереж (CNN).

Основи процесу класифікації емоцій: процес класифікації емоцій можна розділити на кілька етапів: спочатку відбувається виявлення обличчя на зображенні, потім виділення основних рис обличчя, таких як форма рота, положення брів та очей. Після цього нейронна мережа аналізує ці риси, розпізнає патерни (повторювані закономірності) та зіставляє їх із попередньо визначеними класами емоцій.

Архітектура нейронної мережі: глибокі нейронні мережі, зокрема згорткові нейронні мережі (CNN), є найпоширенішими підходами для класифікації емоцій. CNN добре підходять для аналізу зображень, оскільки вони здатні автоматично виділяти ключові риси, важливі для розпізнавання емоцій. Ці мережі складаються з кількох шарів:

- Згорткові шари відповідають за виділення ознак зображень, таких як лінії, контури та текстури;
- Шари об'єднання (Pooling layers) зменшують розмірність даних, зберігаючи важливі ознаки, що допомагає зменшити обчислювальну складність;
- Повнозв'язані шари обробляють виділені ознаки та приймають рішення про те, яка емоція представлена на зображенні [23].

Більшість систем розпізнають обличчя в контрольованих умовах, наприклад, без волосся на обличчі/окулярів, без різких рухів головою, перший кадр повинен бути нейтральною емоцією і т.д., і тому в даний час довільне

розпізнавання облич привертає велику увагу. Зазвичай розпізнавання облич здійснюється двома способами. При цілісному підході обличчя визначається як єдине ціле, в той час як при аналітичному підході виявляються лише деякі важливі риси обличчя. Після того, як обличчя виявлено, є 2 способи виділити риси. У цілісній моделі обличчя використовується метод на основі шаблонів.

В аналітичній моделі обличчя використовуються методи, засновані на рисах, щоб відстежувати риси обличчя, коли людина показує вираз обличчя. Система складається з частини відстеження обличчя, яка виводить вектор особливостей руху певних ділянок обличчя. Ці характеристики використовуються як вхідні дані для класифікатора.

Трекер обличчя використовує підхід на основі моделі, де будується явна 3D-структурна модель обличчя. У першому кадрі послідовності зображень інтерактивно виділяються характерні риси обличчя, такі як куточки очей і куточки рота. Потім загальна модель обличчя деформується відповідно до обраних рис обличчя.

Модель обличчя складається з 16 ділянок поверхні, вбудованих в об'єми Безьє. Визначені таким чином ділянки поверхні гарантовано є безперервними і гладкими. Змінюючи розташування контрольних точок в об'ємі Безьє, можна змінювати форму сіті. Почнемо з кривої Безьє. Для набору контрольних точок відповідна крива Безьє (або крива Бернштейна-Безьє) має вигляд:

$$=$$

де форма кривої контролюється контрольними точками b_i та u , що знаходяться в діапазоні $[0,1]$ [24].

При переміщенні контрольних точок отримується нова форма відповідно до поліномів Бернштейна $B_n(u)$ у наведеному вище рівнянні. Переміщення точки на кривій можна описати в термінах лінійних комбінацій переміщень контрольних точок.

Об'єм Безьє є прямим продовженням кривої Безьє і визначається рівнянням $V=BD$, записаним у матричній формі. У цьому рівнянні V – це

переміщення вузлів сітки, D – матриця, стовпці якої є векторами переміщень контрольних точок об'єму Безьє, а B – відображення в термінах поліномів Бернштейна. Іншими словами, зміна форми моделі обличчя може бути описана в термінах деформацій в D .

Після того, як модель побудована і налаштована, можна відстежувати рух голови і локальні деформації рис обличчя, таких як брови, повіки і рот. Спочатку вимірюються рухи 2D-зображення за допомогою зіставлення шаблонів між кадрами з різною роздільною здатністю. Для більш надійного відстеження використовуються шаблони зображень з попереднього кадру і з найпершого кадру.

Виміряні 2D-рухи зображення моделюються як проекції справжніх 3D-рухів на площину зображення. За 2D-переміщеннями багатьох точок на сітці можна оцінити 3D-переміщення, розв'язавши надмірно визначену систему рівнянь проекційних переміщень за методом найменших квадратів. Відновлені рухи представлені в термінах величин деяких попередньо визначених рухів різних рис обличчя. Кожному руху риси відповідає проста деформація на обличчі, визначена в термінах параметрів керування об'ємом Безьє.

Ми називаємо ці вектори руху одиницями руху (Motion-Units, MU). Зауважте, що вони схожі, але не еквівалентні AU Екмана і є числовими за своєю природою, представляючи не тільки активацію області обличчя, але й напрямок та інтенсивність руху. MU, що використовуються в трекері обличчя, показані за допомогою Рис. 2.4 [25].

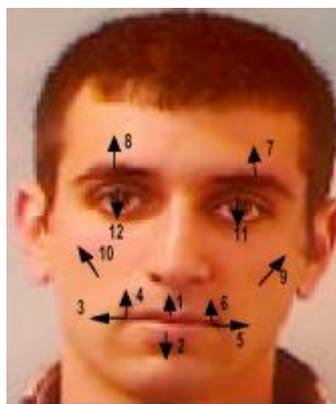


Рисунок 2.4 – Демонстрація вимірювання рухів обличчя

На етапі навчання мережа проходить через тисячі чи мільйони зображень облич із попередньо анотованими емоціями (щастя, сум, злість, страх, здивування, відраза, нейтральний стан тощо). У процесі навчання вона оптимізує свої ваги, щоб правильно класифікувати нові зображення, яких вона раніше не бачила.

Процес навчання: для того щоб мережа навчилася правильно класифікувати емоції, використовуються великі набори даних із маркованими зображеннями облич. Кожне зображення супроводжується певною емоційною міткою, що дозволяє моделі вивчати зв'язки між конкретними рисами обличчя та відповідними емоціями.

Процес навчання включає такі кроки:

1. Передача зображень через мережу: кожне зображення пропускається через згорткові шари, які виділяють різні ознаки обличчя (наприклад, положення очей, кути рота, вирази лоба);
2. Аналіз ознак: після виділення ознак обличчя, ці дані передаються до повнозв'язаних шарів, де модель намагається зіставити ці ознаки з певною емоцією;
3. Оновлення ваг: після кожної ітерації навчання модель порівнює свої передбачення з фактичними результатами, і якщо виникають помилки, ваги мережі коригуються, щоб поліпшити точність наступних передбачень.

Використання попередньо навчених моделей: оскільки навчання з нуля великих нейронних мереж вимагає значних обчислювальних ресурсів і часу, часто використовують попередньо навчені моделі. Такі моделі, як VGGFace або ResNet, попередньо навчаються на великих наборах даних і можуть бути донавчені (fine-tuned) на нових даних для класифікації емоцій. Цей підхід значно зменшує час навчання і покращує точність [26].

Отже, класифікація емоцій за допомогою нейронних мереж є потужним інструментом у сучасних системах розпізнавання облич. Глибокі нейронні

мережі, завдяки своїй здатності автоматично виділяти складні ознаки зображень, забезпечують високу точність та надійність класифікації. Попри певні виклики, такі як зміни освітлення та індивідуальні відмінності, цей підхід є найефективнішим для точного розпізнавання емоцій на зображеннях обличчя.

2.5 Алгоритми класифікації

Алгоритми класифікації широко використовуються для задач розпізнавання емоцій за рисами обличчя, голосом чи іншими даними. Хоча нейронні мережі займають лідируючі позиції у цій сфері, класичні алгоритми машинного навчання, такі як Support Vector Machines (SVM), Random Forests (RF) і K-Nearest Neighbors (KNN), також часто застосовуються. Ці методи є більш легкими в реалізації, потребують менше обчислювальних ресурсів і можуть бути ефективними для менших наборів даних. Давайте розглянемо їх основні характеристики та застосування в контексті класифікації емоцій.

(див. Рис. 2.5) – це один із найпопулярніших алгоритмів для задач класифікації, зокрема для класифікації емоцій. SVM працює за принципом пошуку оптимальної гіперплощини, яка розділяє дані на два або більше класів. У випадку класифікації емоцій, модель SVM намагається знайти межі між різними емоційними станами, такими як щастя, сум чи гнів, на основі виявлених ознак обличчя або інших сигналів.

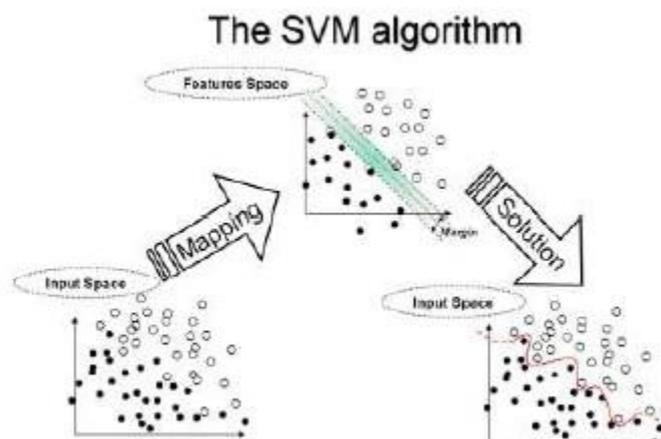


Рисунок 2.5 – Проста схема роботи Support Vector Machines

SVM може працювати з нелінійними даними, використовуючи ядра (kernels), такі як поліноміальні або радіальні базисні функції (RBF) (див. Рис. 2.5.2), що дозволяє йому ефективно розпізнавати складні патерни емоцій [27].

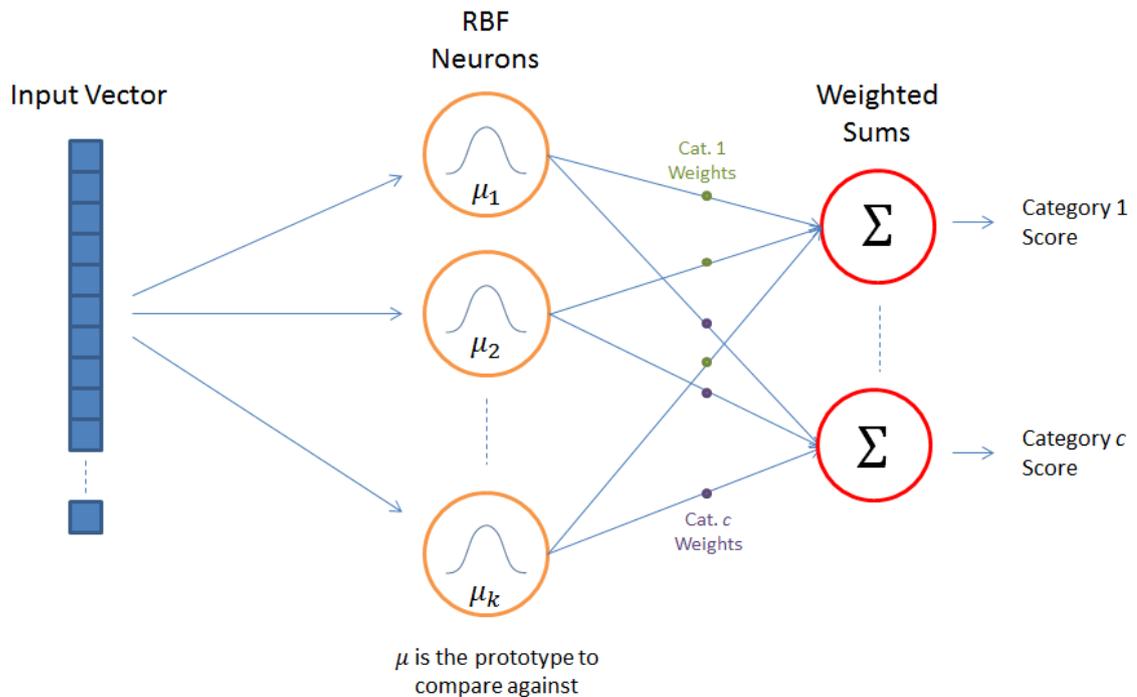


Рисунок 2.5.2 – Деякі радіальні базисні функції

(див. Рис. 2.6) – це ансамблевий метод машинного навчання, що складається з великої кількості вирішувальних дерев (decision trees). Кожне дерево у моделі класифікує дані, а результат визначається голосуванням більшості дерев. Random Forest є стійким і надійним методом для задач класифікації емоцій, оскільки він ефективно обробляє високовимірні дані і добре працює навіть при наявності шуму [28].

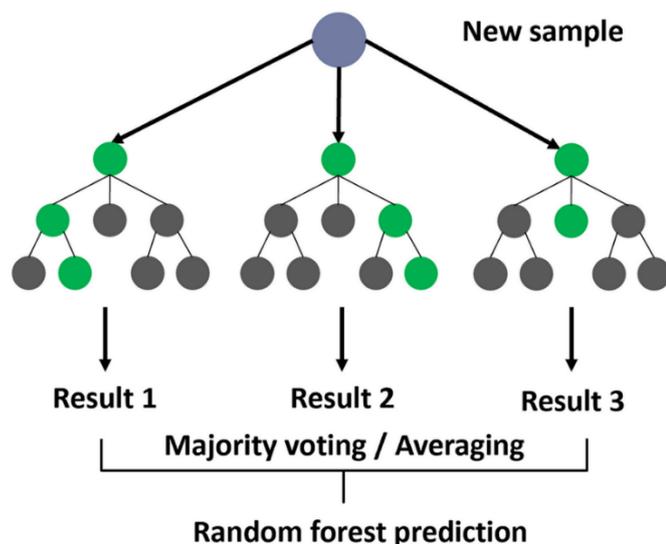


Рисунок 2.6 – Демонстрація Random Forests

Для класифікації емоцій Random Forest може використовувати різні ознаки, такі як положення м'язів обличчя, текстури чи інші фізіологічні параметри, щоб навчити модель розрізняти емоції. Завдяки використанню випадкової вибірки ознак і даних для кожного дерева, RF зменшує ймовірність надмірного навчання і підвищує загальну точність.

(див. Рис. 2.7) – це простий, але ефективний алгоритм для класифікації, який базується на відстанях між точками у багатовимірному просторі ознак. Для класифікації нової точки (наприклад, зображення з обличчям) алгоритм знаходить K найближчих сусідів у тренувальних даних і класифікує її на основі більшості емоційних міток серед цих сусідів [29].

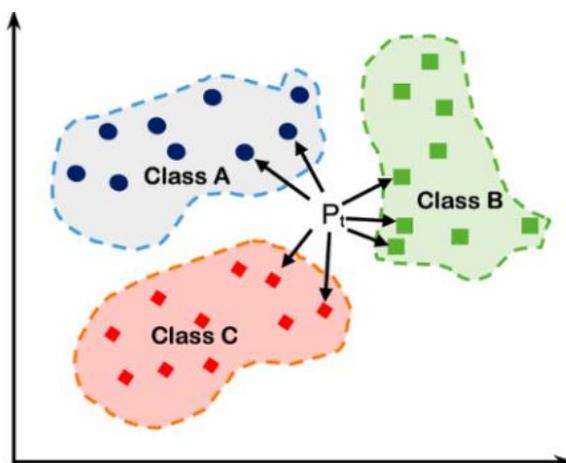


Рисунок 2.7 – Демонстрація K-Nearest Neighbors

KNN часто використовується для класифікації емоцій завдяки простоті його реалізації. Він не потребує складного процесу навчання: лише зберігає всі тренувальні дані і проводить класифікацію під час запиту. KNN добре підходить для невеликих наборів даних, де патерни емоцій можна визначити за допомогою відстаней між різними ознаками обличчя або іншими сигналами [30].

Порівняння цих алгоритмів показано у вигляді Таблиці 2.1.

Таблиця 2.1 – Порівняння популярних алгоритмів класифікації

| Алгоритм | Принцип роботи | Переваги | Недоліки | Застосування |
|-------------------------------|------------------------------------------------------------------------|---------------------------------------------------------|----------------------------------------------------------|---------------------------------------------------------------------------|
| Support Vector Machines (SVM) | Знаходить оптимальну гіперплощину для розділення класів | - Висока ефективність на малих наборах даних | - Вимогливий до обчислювальних ресурсів на великих даних | Класифікація емоцій на основі рис обличчя чи інших ознак |
| | | - Може працювати з нелінійними даними за допомогою ядер | - Не підходить для багатокласової класифікації | |
| Random Forests (RF) | Використовує ансамбль дерев рішень для прийняття рішення | - Стійкість до шуму | - Важко інтерпретувати модель | Класифікація емоцій за різними ознаками обличчя, фізіологічними сигналами |
| | | - Висока точність | - Потребує більше часу на навчання | |
| | | - Стійкість до надмірного навчання | | |
| K-Nearest Neighbors (KNN) | Класифікує нові дані на основі найближчих сусідів у тренувальних даних | - Простота реалізації | - Чутливість до шуму | Класифікація емоцій для невеликих наборів даних |
| | | - Не потребує навчання моделі | - Низька продуктивність на великих наборах даних | |
| | | - Гнучкий вибір метрики для вимірювання відстаней | | |

Таким чином, Support Vector Machines, Random Forests і K-Nearest Neighbors є популярними класичними алгоритмами для класифікації емоцій. Вибір між ними залежить від особливостей завдання, обсягу даних і вимог до точності. Підсумовуючи:

1. SVM підходить для невеликих і середніх наборів даних з нелінійними патернами та високовимірними ознаками;
2. Random Forests добре справляється зі складними, шумними даними, забезпечуючи високу точність завдяки використанню ансамблів дерев;
3. KNN є простим і ефективним для невеликих наборів даних, але має обмеження щодо продуктивності при роботі з великими обсягами інформації.

Кожен з цих алгоритмів може бути успішно використаний для задач класифікації емоцій, залежно від конкретних умов та характеристик вхідних даних.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

3.1 Огляд інструментів та технологій

У даному проєкті для розпізнавання емоцій на зображеннях було використано широкий набір інструментів та технологій, що забезпечили високу ефективність і зручність роботи. Основою розробки стала мова програмування Python, відома своєю гнучкістю та простотою. Для обробки та аналізу зображень були задіяні бібліотеки OpenCV і DeepFace, що дозволяють здійснювати детекцію облич і класифікацію емоцій. Створення графічного інтерфейсу реалізовано за допомогою бібліотеки tkinter, що забезпечує інтерактивну взаємодію з користувачем, тоді як для візуалізації результатів застосовано matplotlib, яка надає зручні засоби для графічного представлення даних.

1. Мова програмування: Python

Python було обрано для розробки програмного забезпечення через його універсальність, простоту у використанні та наявність великої кількості бібліотек, що підтримують обробку зображень, машинне навчання та створення графічних інтерфейсів. Однією з головних причин популярності Python є його відкритий вихідний код та активна спільнота розробників, яка постійно підтримує і розвиває бібліотеки та інструменти.

Python відрізняється простим і лаконічним синтаксисом, який дозволяє швидко писати й тестувати код, що особливо важливо при розробці наукових проєктів. Оскільки розробка програмного забезпечення для розпізнавання емоцій на зображеннях вимагає складної обробки даних та інтеграції з різними технологіями, Python забезпечує високу продуктивність при виконанні таких завдань.

Завдяки наявності потужних бібліотек, Python дозволяє легко вирішувати завдання комп'ютерного зору та глибокого навчання. Це одна з

ключових переваг у проектах, що стосуються обробки зображень і розпізнавання облич. Python надає широкий спектр інструментів для передобробки зображень, виділення рис обличчя, класифікації емоцій та візуалізації результатів.

Ще однією важливою перевагою Python є його здатність інтегруватися з такими популярними фреймворками, як TensorFlow і Keras, що дозволяє використовувати переваги глибоких нейронних мереж для класифікації емоцій. Крім того, Python підтримує мультиплатформеність, тобто програми, написані на цій мові, можуть працювати на різних операційних системах, таких як Windows, macOS і Linux, що робить його універсальним інструментом для розробників (див. Рис. 3.1) [31].

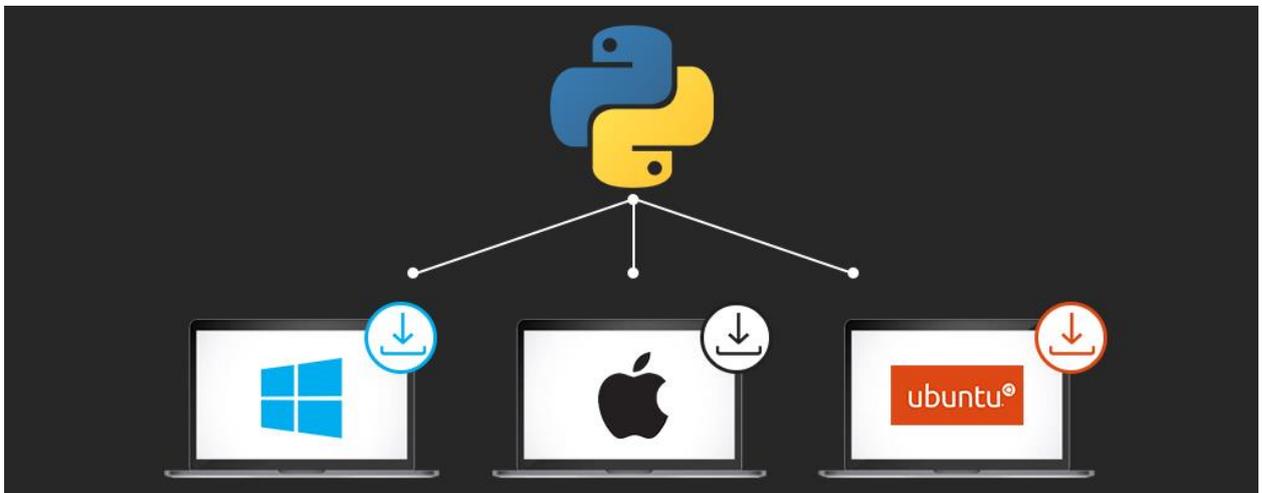


Рисунок 3.1 – Підтримка операційними системами мови Python

Отже, Python є ідеальним вибором для проектів, пов'язаних із розпізнаванням емоцій, оскільки поєднує простоту, гнучкість, широкі можливості та потужну екосистему бібліотек, яка дозволяє швидко реалізувати складні алгоритми.

2. Бібліотеки для аналізу зображень

Аналіз зображень є ключовою частиною процесу розпізнавання емоцій людини, і для цього проекту було використано кілька потужних бібліотек Python, які забезпечують ефективну обробку та аналіз зображень.

Одна з основних бібліотек для роботи із зображеннями в Python – OpenCV (Open Source Computer Vision Library) (див. Рис. 3.2). OpenCV є відкритим проектом із потужним функціоналом, що надає інструменти для обробки зображень та відео в реальному часі. Ця бібліотека дозволяє виконувати широкий спектр завдань комп'ютерного зору, таких як виявлення об'єктів, сегментація зображень, корекція кольорів, фільтрація, зміна розмірів зображень тощо. У контексті розпізнавання емоцій OpenCV використовується для передобробки зображень: зчитування, зміна розмірів та нормалізації зображення перед подачею його на моделі глибокого навчання [32].

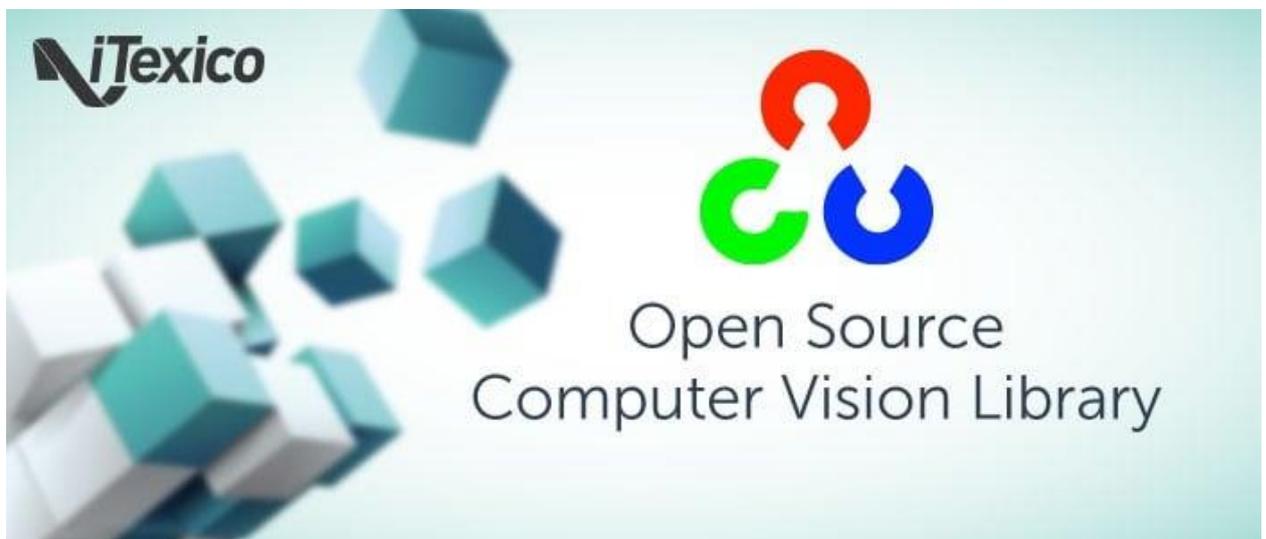


Рисунок 3.2 – Логотип бібліотеки OpenCV

Іншою важливою бібліотекою, що використовується в цьому проекті, є DeepFace – високорівнева бібліотека для аналізу обличчя, яка інтегрує кілька потужних моделей для розпізнавання обличчя та емоцій, таких як VGG-Face, Google FaceNet, OpenFace та Facebook DeepFace. DeepFace використовує глибокі згорткові нейронні мережі (CNN), спеціально навчені для класифікації емоцій за виразами обличчя. Однією з переваг цієї бібліотеки є її здатність швидко інтегрувати готові моделі для точного та швидкого розпізнавання емоцій, що значно зменшує потребу в ручному налаштуванні моделей або навчанні з нуля.

DeerFace виконує автоматичне виявлення обличчя на зображенні, після чого проводить аналіз виразів обличчя для класифікації емоцій, таких як радість, сум, гнів, страх та інші. У цьому проекті вона забезпечує високу точність розпізнавання емоцій завдяки використанню попередньо навчених моделей на великих наборах даних.

Таким чином, комбінація бібліотек OpenCV та DeerFace дозволяє ефективно обробляти зображення, виявляти обличчя та визначати емоції на зображенні. OpenCV надає необхідні інструменти для підготовки зображень і детекції облич, тоді як DeerFace використовує передові методи глибокого навчання для точного аналізу емоцій [33].

3. Бібліотеки для побудови графічного інтерфейсу

Для створення зручного та інтуїтивно зрозумілого графічного інтерфейсу (GUI) у Python широко використовується бібліотека tkinter (див. Рис. 3.3). Це стандартна бібліотека для побудови графічних інтерфейсів, яка постачається разом із Python, що робить її легко доступною і популярною серед розробників.

Рисунок 3.3 – Приклад оформлення графічного дизайну за допомогою бібліотеки tkinter

tkinter забезпечує простий у використанні інструментарій для створення вікон, кнопок, полів введення, текстових віджетів та інших елементів,

необхідних для взаємодії з користувачем. Основна перевага tkinter полягає в його простоті та надійності, що дозволяє швидко створювати графічні інтерфейси без складної конфігурації або використання сторонніх бібліотек.

В даному проекті tkinter використовується для створення вікна програми, яке дозволяє користувачеві вибирати зображення з файлової системи, а також відображати результати розпізнавання емоцій. Завдяки tkinter, програма забезпечує такі можливості:

- Вибір зображення: за допомогою функціоналу файлового діалогу користувач може вибрати зображення зі свого комп'ютера, яке буде використовуватися для аналізу;
- Відображення зображення: tkinter дозволяє вбудовувати зображення прямо в інтерфейс програми, щоб користувач міг переглянути вибране зображення перед проведенням аналізу;
- Багатомовна підтримка: tkinter також дозволяє додати елементи управління для вибору мови інтерфейсу. У проекті реалізовано підтримку української та англійської мов, що робить програму більш гнучкою для користувачів з різних країн [34].

Бібліотека tkinter дозволяє створювати інтуїтивно зрозумілі елементи управління, такі як:

- Кнопки для запуску аналізу зображень;
- Меню та поля вибору, які допомагають користувачам вибирати потрібні опції або мову;
- Інформаційні повідомлення для виведення результатів, що дозволяє програмі надавати зворотній зв'язок користувачеві у зручній та зрозумілій формі.

Однією з переваг tkinter є його простота у налаштуванні та використанні. Оскільки ця бібліотека є частиною стандартного набору Python, вона не вимагає додаткових інсталяцій або залежностей, що робить її зручною для швидкого прототипування і розгортання графічних інтерфейсів. Завдяки

цьому, розробка інтерфейсу програми для розпізнавання емоцій може бути реалізована швидко і без зайвих складнощів.

Загалом, tkinter є ефективним рішенням для побудови графічного інтерфейсу в Python-проектах завдяки своїй простоті, надійності та широким можливостям у створенні інтерактивних додатків. У цьому проекті бібліотека tkinter забезпечує повноцінний користувацький інтерфейс, через який користувач може легко взаємодіяти з програмою, вибирати зображення та переглядати результати аналізу.

4. Бібліотеки для візуалізації результатів

Для візуалізації результатів у програмному забезпеченні для розпізнавання емоцій на зображенні в Python широко використовується бібліотека matplotlib (див. Рис. 3.4). Це одна з найбільш популярних бібліотек для побудови графіків і візуалізації даних, яка надає гнучкий та потужний інструментарій для створення різноманітних типів графічних зображень.

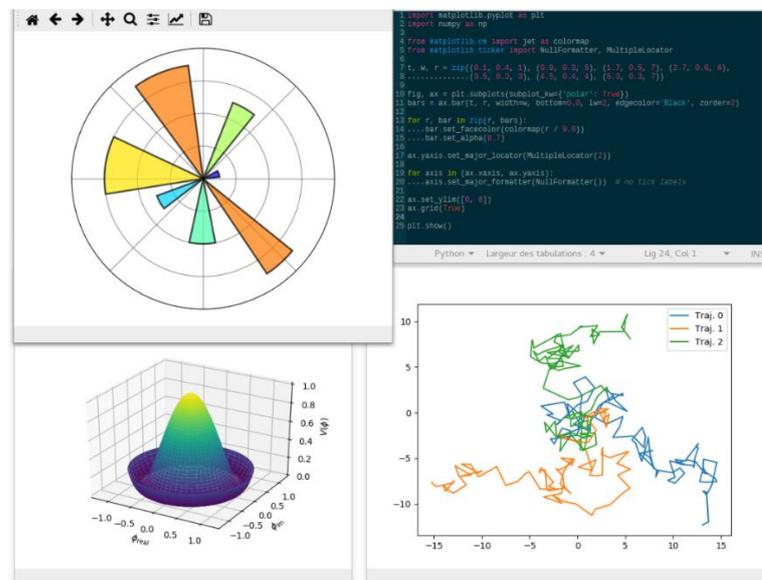


Рисунок 3.4 – Демонстрація можливостей бібліотеки matplotlib

Основною перевагою matplotlib є її здатність швидко та якісно візуалізувати результати аналізу у вигляді графіків, діаграм та інших видів візуальних відображень даних. В контексті проекту, присвяченого розпізнаванню емоцій, ця бібліотека використовується для відображення результатів класифікації емоцій у графічному форматі, що дозволяє

користувачу легко зрозуміти і проаналізувати відсотковий розподіл емоцій, виявлених на зображенні.

В даному проекті `matplotlib` використовується для побудови кругової діаграми (`pie chart`), яка наочно відображає процентний розподіл емоцій. Основні функціональні можливості цієї бібліотеки в межах проекту включають:

- Побудова кругової діаграми: Кожна емоція відображається на круговій діаграмі у вигляді сектора, розмір якого пропорційний проценту цієї емоції серед інших;
- Додавання підписів та легенд: `matplotlib` дозволяє легко додавати текстові підписи та легенди до графіків, що полегшує користувачам розуміння результатів. У нашому проекті підписи відображаються мовою інтерфейсу (українською або англійською), залежно від вибору користувача;
- Настроювання стилю графіків: `matplotlib` надає великий набір можливостей для налаштування кольорів, шрифтів та інших елементів графіка, що дозволяє зробити його більш естетичним і зрозумілим для користувача [35].

Крім того, `matplotlib` легко інтегрується з іншими бібліотеками Python, такими як `tkinter`, що дозволяє відображати графіки безпосередньо в графічному інтерфейсі програми. У нашому проекті `matplotlib` використовується для побудови діаграми всередині інтерфейсу, надаючи користувачам можливість бачити результати розпізнавання емоцій в зручній графічній формі.

Ще однією важливою перевагою `matplotlib` є її гнучкість і можливість візуалізації різних типів даних. Хоча в даному проекті використовується кругова діаграма, бібліотека дозволяє створювати також стовпчасті діаграми, лінійні графіки, гістограми та інші форми візуалізації, що може бути корисним у більш складних завданнях аналізу даних.

Завдяки використанню `matplotlib` у цьому проекті, результати розпізнавання емоцій стають не лише точними, але й наочно представленими,

що полегшує аналіз для користувачів та робить взаємодію з програмою інтуїтивнішою.

Таким чином, для розробки даного програмного забезпечення для розпізнавання емоцій людини на зображенні були використані найбільш правильні та зручні інструменти та технології на думку розробника.

3.2 Функціональні вимоги до ПЗ

Функціональні вимоги до програмного забезпечення для розпізнавання емоцій на зображеннях визначають основні можливості системи, які повинні бути реалізовані для забезпечення її ефективної роботи та взаємодії з користувачем. Нижче описано детальні функціональні вимоги до ПЗ.

1. Завантаження зображення: ПЗ повинно надавати користувачеві можливість вибору зображення для аналізу. Це має бути реалізовано через діалогове вікно файлового менеджера, яке дозволяє обирати зображення різних форматів, таких як JPEG, PNG та інші поширені формати. Після вибору зображення програма повинна відобразити його у вікні інтерфейсу.

2. Виявлення обличчя на зображенні: однією з ключових функцій є автоматичне виявлення обличчя на завантаженому зображенні. ПЗ повинно використовувати алгоритми комп'ютерного зору для розпізнавання облич на зображенні та відокремлення їх від фону. У випадку, якщо обличчя не виявлено, користувач повинен отримати повідомлення про помилку з відповідною інструкцією.

3. Аналіз емоцій: після успішного виявлення обличчя, система повинна здійснити розпізнавання емоцій на основі виразів обличчя. Програма має використовувати попередньо навчені моделі для аналізу емоцій, такі як радість, сум, злість, здивування, страх, нейтральний стан та інші. Кожна емоція повинна бути представлена у вигляді процентного співвідношення, яке демонструє впевненість системи в її визначенні.

4. Виведення результатів: програма повинна надати користувачеві текстовий звіт із результатами аналізу. У звіті повинні бути вказані всі

розпізнані емоції, а також їх процентне співвідношення. Це дозволить користувачу швидко зрозуміти, які емоції домінують на зображенні та як вони розподіляються між собою.

5. Візуалізація результатів: для кращого розуміння результатів аналізу, ПЗ повинно побудувати графік. Ця діаграма повинна відображати процентний розподіл емоцій у зрозумілому візуальному форматі. Кожен сегмент діаграми повинен представляти окрему емоцію, а його розмір відповідати процентному значенню цієї емоції. Діаграма повинна автоматично оновлюватись після кожного нового аналізу зображення.

6. Багатомовна підтримка: програмне забезпечення повинно мати функцію багатомовного інтерфейсу, зокрема підтримувати українську та англійську мови. Користувач повинен мати можливість вибирати бажану мову під час використання програми. Усі текстові підписи, повідомлення, інструкції та результати мають відображатися мовою, вибраною користувачем, щоб забезпечити зручність використання для широкого кола користувачів.

7. Масштабування та відображення зображення: для кращої взаємодії з користувачем програма повинна автоматично масштабувати вибране зображення так, щоб воно відповідало розмірам вікна графічного інтерфейсу. Це дозволить уникнути ситуацій, коли великі зображення займають весь екран або коли маленькі зображення важко розглянути. Зображення має бути розміщене у вікні так, щоб всі елементи управління залишалися доступними для користувача.

8. Повідомлення про помилки: програма повинна обробляти різні помилки, які можуть виникнути під час аналізу. Якщо зображення не містить обличчя, програма повинна повідомити користувача про це і запропонувати вибрати інше зображення. Крім того, якщо формат файлу зображення не підтримується або файл пошкоджений, ПЗ повинно відобразити відповідне повідомлення про помилку.

9. Збереження результатів: програма повинна забезпечити можливість збереження результатів аналізу у вигляді графіків або текстових звітів. Це

дозволить користувачам зберегти важливу інформацію для подальшого використання або порівняння. Наприклад, користувач може зберегти діаграму з відсотковим розподілом емоцій у вигляді зображення або текстовий файл із детальним звітом про результати.

Загалом, ці функціональні вимоги забезпечують повний цикл обробки зображень та аналізу емоцій, включаючи передобробку даних, детекцію облич, аналіз емоцій, візуалізацію та взаємодію з користувачем через зручний графічний інтерфейс.

3.3 Нефункціональні вимоги до ПЗ

Нефункціональні вимоги до програмного забезпечення для розпізнавання емоцій на зображенні включають характеристики системи, які не стосуються її функціональності, але впливають на ефективність, надійність та зручність використання. Нижче наведено ключові нефункціональні вимоги:

1. **Продуктивність:** ПЗ повинно швидко і ефективно обробляти зображення та розпізнавати емоції без значних затримок. Час аналізу одного зображення не повинен перевищувати кількох секунд. Навіть при обробці великих зображень або зображень з великою кількістю облич, система повинна залишатися швидкодіючою.

2. **Масштабованість:** програмне забезпечення повинно бути масштабованим, тобто здатним ефективно працювати як з невеликими, так і з великими зображеннями, а також обробляти одночасно декілька зображень, якщо така функція буде додана. Система має підтримувати додавання нових функцій без значного впливу на існуючу продуктивність.

3. **Надійність:** програма повинна працювати стабільно та без збоїв, навіть у випадку некоректних або непередбачуваних дій користувача, наприклад, спроби завантажити пошкоджене зображення або файл у непідтримуваному форматі. Система повинна коректно обробляти помилки і забезпечувати повідомлення користувача про них.

4. Сумісність: ПЗ повинно бути сумісним з операційними системами, такими як Windows, macOS та Linux. Крім того, система повинна працювати з різними форматами зображень, включаючи JPEG, PNG, BMP та інші, що дозволяє забезпечити зручність використання для широкого кола користувачів.

5. Безпека: ПЗ повинно забезпечувати конфіденційність даних користувача. Всі зображення, що аналізуються, мають оброблятися локально, без відправлення на зовнішні сервери, щоб захистити персональні дані. Важливо забезпечити відсутність зберігання зображень або результатів аналізу без згоди користувача.

6. Зручність використання (юзабіліті): інтерфейс повинен бути інтуїтивно зрозумілим для користувачів різного рівня технічної підготовки. Навігація по програмі має бути простою, а всі основні функції – легко доступними. Програма повинна мати зрозумілі повідомлення про дії та помилки, а також інструкції, як правильно завантажити зображення або вибрати мову.

7. Модифікованість: ПЗ має бути гнучким і дозволяти майбутні зміни або додавання нових функцій з мінімальними затратами часу та ресурсів. Архітектура програми повинна дозволяти легко інтегрувати нові методи розпізнавання емоцій або додаткові інструменти для аналізу зображень.

8. Мультимовність: оскільки система підтримує декілька мов (українську та англійську), вона повинна дозволяти легке додавання інших мов у майбутньому. Мовний інтерфейс повинен коректно відображати всі елементи, незалежно від обраної мови, і не вимагати додаткового перекладу для стандартних елементів, таких як кнопки чи меню.

9. Енергоефективність: ПЗ має бути оптимізованим для використання на комп'ютерах із низькими технічними характеристиками, що передбачає мінімальне споживання ресурсів (оперативної пам'яті, процесора) під час роботи програми. Це особливо важливо для ноутбуків та мобільних пристроїв з обмеженими ресурсами.

10. Точність: програма повинна забезпечувати високий рівень точності при розпізнаванні емоцій на обличчі. Модель, яку використовує ПЗ, повинна бути здатною коректно ідентифікувати різні емоції з мінімальною кількістю помилок. Рівень точності має відповідати сучасним стандартам у галузі комп'ютерного зору.

11. Відповідність стандартам: ПЗ повинно відповідати стандартам якості розробки програмного забезпечення та стандартам у галузі обробки зображень і комп'ютерного зору. Це включає використання перевірених і надійних бібліотек, таких як OpenCV і TensorFlow.

Ці нефункціональні вимоги спрямовані на забезпечення надійності, безпеки, продуктивності та зручності програмного забезпечення, що робить його готовим до реального використання та подальшого вдосконалення.

3.4 Реалізація програмного забезпечення

Більш детальний опис блоків розроблюваного програмного забезпечення для розпізнавання емоцій людини на зображенні:

1. Імпорт бібліотек, як показано на Рис. 3.5.

```

1  import tkinter as tk
2  from tkinter import filedialog
3  from tkinter import messagebox
4  from tkinter import ttk
5  from deepface import DeepFace
6  import cv2
7  from PIL import Image, ImageTk
8  import matplotlib.pyplot as plt
9  import numpy as np
10 import time
11 import psutil # Для отримання інформації про ресурси системи

```

Рисунок 3.5 – Частина програмної реалізації ПЗ

У цьому блоці імпортуються різноманітні бібліотеки, які є критично важливими для функціонування програми. Бібліотека «tkinter» забезпечує графічний інтерфейс користувача (GUI), дозволяючи створювати вікна, кнопки, текстові поля та інші елементи інтерфейсу. Бібліотека «filedialog»

надає можливість відкривати діалоги для вибору файлів, а «messagebox» дозволяє виводити повідомлення про помилки та інші сповіщення.

«DeepFace» – це бібліотека для глибокого навчання, яка здійснює розпізнавання емоцій за зображеннями. «cv2» (OpenCV) використовується для обробки зображень, включаючи завантаження, масштабування та конвертацію зображень. «PIL» (Python Imaging Library) допомагає в обробці та відображенні зображень у Tkinter. Бібліотека «matplotlib» дозволяє створювати графіки та діаграми для візуалізації результатів, а «numpy» використовується для роботи з масивами даних. «psutil» – це бібліотека для моніторингу системних ресурсів, яка дозволяє відстежувати використання ЦП і пам'яті.

2. Глобальні змінні, як показано на Рис. 3.6.

```
13  # Глобальна змінна для вибору мови
14  current_language = "English"
```

Рисунок 3.6 – Частина програмної реалізації ПЗ

Глобальна змінна «current_language» визначає, яка мова наразі використовується в інтерфейсі програми. За замовчуванням вона встановлена на англійську, але користувач може змінити цю мову, вибравши відповідний варіант у комбінованому списку.

3. Переклад емоцій і ресурсів, як показано на Рис. 3.7.

```

16 # Словник для перекладу емоцій на українську
17 emotion_translation = {
18     'happy': 'щасливий',
19     'sad': 'сумний',
20     'angry': 'злий',
21     'surprise': 'здивований',
22     'fear': 'страх',
23     'disgust': 'відраза',
24     'neutral': 'нейтральний'
25 }
26
27 # Словник для перекладу ресурсів
28 resource_translation = {
29     "execution_time": {
30         "English": "Execution Time",
31         "Українська": "Час виконання"
32     },
33     "cpu_usage": {
34         "English": "CPU Usage",
35         "Українська": "Використання ЦП"
36     },
37     "memory_usage": {
38         "English": "Memory Usage",
39         "Українська": "Використана пам'ять"
40     },
41     "most_likely_emotion": {
42         "English": "Most Likely Emotion",
43         "Українська": "Найточніша емоція"
44     }
45 }

```

Рисунок 3.7 – Частина програмної реалізації ПЗ

Цей блок містить два словники. Перший – «emotion_translation» – відповідає за переклад назв емоцій з англійської на українську мову. Це дозволяє програмі відображати результати аналізу емоцій у зручному для користувача форматі. Другий словник – «resource_translation» – використовується для перекладу повідомлень про ресурси, що витрачаються програмою, таких як час виконання, використання ЦП і пам'яті.

4. Функції для роботи з мовою, як показано на Рис. 3.8.

```

47 # Функція для вибору мови
48 def set_language(lang):
49     global current_language
50     current_language = lang
51     update_ui()
52
53 # Функція для оновлення UI відповідно до вибраної мови
54 def update_ui():
55     if current_language == "English":
56         btn_select_image.config(text="Select Image")
57         btn_analyze.config(text="Analyze")
58         lbl_language.config(text="Language:")
59     else:
60         btn_select_image.config(text="Вибрати зображення")
61         btn_analyze.config(text="Аналізувати")
62         lbl_language.config(text="Мова:")

```

Рисунок 3.8 – Частина програмної реалізації ПЗ

Ці функції забезпечують функціональність вибору мови інтерфейсу. «set_language(lang)» змінює мову на ту, що була обрана користувачем, а «update_ui()» оновлює текст на елементах інтерфейсу відповідно до вибраної мови. Це дозволяє програмі бути зручною для користувачів, які говорять різними мовами.

5. Функція для вибору зображення, як показано на Рис. 3.9.

```

64 # Функція для вибору зображення
65 def select_image():
66     try:
67         file_path = filedialog.askopenfilename(
68             filetypes=[("Image files", "*.jpg *.jpeg *.png")]
69         )
70         if file_path:
71             load_image(file_path)
72     except Exception as e:
73         messagebox.showerror("Помилка", f"Неможливо відкрити файл: {str(e)}")

```

Рисунок 3.9 – Частина програмної реалізації ПЗ

Функція «select_image()» відкриває діалогове вікно для вибору зображення. Після вибору зображення програма перевіряє, чи було вибрано файл, і, якщо так, передає його на подальшу обробку. Цей блок також включає

обробку можливих помилок, щоб уникнути збоїв програми при відкритті файлів.

6. Функції для обробки зображення, як показано на Рис. 3.10.

```

75 # Масштабування зображення
76 def resize_image(img, max_size):
77     height, width = img.shape[:2]
78     scaling_factor = min(max_size / width, max_size / height)
79     new_width = int(width * scaling_factor)
80     new_height = int(height * scaling_factor)
81     resized_image = cv2.resize(img, (new_width, new_height), interpolation=cv2.INTER_AREA)
82     return resized_image
83
84 # Завантаження та відображення зображення
85 def load_image(image_path):
86     global img
87     try:
88         img = cv2.imread(image_path)
89         img_resized = resize_image(img, 400) # Обмеження розміру
90         img_rgb = cv2.cvtColor(img_resized, cv2.COLOR_BGR2RGB)
91         img_pil = Image.fromarray(img_rgb)
92         img_tk = ImageTk.PhotoImage(img_pil)
93         lbl_image.config(image=img_tk)
94         lbl_image.image = img_tk
95     except Exception as e:
96         messagebox.showerror("Помилка", f"Помилка завантаження зображення: {str(e)}")

```

Рисунок 3.10 – Частина програмної реалізації ПЗ

Функції «resize_image(img, max_size)» та «load_image(image_path)» відповідають за обробку зображення. «resize_image()» зменшує розмір зображення до максимально допустимого значення, щоб воно відображалось коректно у вікні програми. «load_image(image_path)» завантажує обране зображення, конвертує його в RGB-формат та відображає у графічному інтерфейсі. Важливо, що обробка зображень проводиться ефективно, щоб уникнути затримок у роботі програми.

7. Функція для аналізу емоцій, як показано на Рис. 3.11.

```

98 # Функція для аналізу емоцій
99 def analyze_emotions():
100     if img is None:
101         messagebox.showerror("Помилка", "Спочатку виберіть зображення!")
102         return
103
104     try:
105         start_time = time.time() # Початок вимірювання часу
106         cpu_start = psutil.cpu_percent(interval=None)
107         memory_start = psutil.virtual_memory().used
108
109         result = DeepFace.analyze(img, actions=['emotion'])
110
111         if isinstance(result, list):
112             result = result[0]
113
114         # Вимірювання витрат ресурсів
115         cpu_end = psutil.cpu_percent(interval=None)
116         memory_end = psutil.virtual_memory().used
117         elapsed_time = time.time() - start_time
118
119         display_emotions(result['emotion'])
120
121         # Найточніша емоція
122         most_likely_emotion = max(result['emotion'], key=result['emotion'].get)
123
124         # Переклад тексту про найточнішу емоцію
125         if current_language == "Українська":
126             most_likely_emotion_text = emotion_translation.get(most_likely_emotion, most_likely_emotion)
127             most_likely_emotion_message = f"{resource_translation['most_likely_emotion'][current_language]}: {most_likely_emotion_text}"
128         else:
129             most_likely_emotion_message = f"{resource_translation['most_likely_emotion'][current_language]}: {most_likely_emotion}"
130
131         # Повідомлення про використанні ресурси
132         messagebox.showinfo(
133             "Завершено",
134             f"{resource_translation['execution_time'][current_language]}: {elapsed_time:.2f} сек.\n"
135             f"{resource_translation['cpu_usage'][current_language]}: {cpu_end - cpu_start:.2f}%\n"
136             f"{resource_translation['memory_usage'][current_language]}: {(memory_end - memory_start) / (1024 ** 2):.2f} MB\n"
137             f"{most_likely_emotion_message}"
138         )
139
140     except Exception as e:
141         messagebox.showerror("Помилка", f"Помилка аналізу: {str(e)}")

```

Рисунок 3.11 – Частина програмної реалізації ПЗ

Функція «analyze_emotions()» виконує основну роботу програми – аналіз емоцій на зображенні. Вона викликає метод «DeepFace.analyze()», який обробляє зображення та повертає результати у вигляді емоцій. Функція також вимірює час виконання, використання ЦП і пам'яті, що дозволяє користувачу отримати інформацію про ефективність роботи програми. Результати аналізу виводяться у вигляді гістограми, а також відображається найточніша емоція.

8. Функція для перекладу емоцій, як показано на Рис. 3.12.

```

143 # Функція для перекладу емоцій на українську мову
144 def translate_emotions(emotions):
145     if current_language == "Українська":
146         translated_emotions = {}
147         for emotion, value in emotions.items():
148             translated_emotions[emotion_translation.get(emotion, emotion)] = value
149         return translated_emotions
150     return emotions

```

Рисунок 3.12 – Частина програмної реалізації ПЗ

Функція «translate_emotions(emotions)» відповідає за переклад назв емоцій на обрану мову програми. Вона забезпечує відображення результатів у зручному форматі, щоб користувачі могли легко зрозуміти виведені дані.

9. Відображення результатів, як показано на Рис. 3.13.

```

152 # Відображення графіків з емоціями
153 def display_emotions(emotions):
154     emotions = translate_emotions(emotions)
155
156     fig, axs = plt.subplots(2, 1, figsize=(6, 8))
157
158     # Гістограма
159     emotions_list = list(emotions.keys())
160     emotions_values = list(emotions.values())
161     y_pos = np.arange(len(emotions_list))
162
163     axs[0].barh(y_pos, emotions_values, align='center', alpha=0.7, color='skyblue')
164     axs[0].set_yticks(y_pos)
165     axs[0].set_yticklabels(emotions_list)
166     axs[0].set_xlabel('Відсоток' if current_language == "Українська" else 'Percentage')
167     axs[0].set_title('Розпізнавання емоцій' if current_language == "Українська" else 'Emotion Recognition')
168
169     # Кругова діаграма
170     axs[1].pie(emotions_values, labels=emotions_list, autopct='%1.1f%%', colors=plt.cm.Paired.colors)
171     axs[1].set_title('Співвідношення емоцій' if current_language == "Українська" else 'Emotion Proportion')
172
173     plt.tight_layout()
174     plt.show()

```

Рисунок 3.13 – Частина програмної реалізації ПЗ

Функція «display_emotions(emotions)» візуалізує результати аналізу емоцій у вигляді графіків. Вона створює горизонтальну гістограму, де по осі Y відображаються емоції, а по осі X – їх відсоткове співвідношення. Для цього використовуються можливості бібліотеки Matplotlib, що дозволяє створювати інформативні та зрозумілі графіки.

10. Функція для централізованого відкриття вікна, як показано на Рис. 3.14.

```

176 # Функція для централізованого відкриття вікна
177 def center_window(window, width=600, height=600):
178     screen_width = window.winfo_screenwidth()
179     screen_height = window.winfo_screenheight()
180     x = (screen_width // 2) - (width // 2)
181     y = (screen_height // 2) - (height // 2)
182     window.geometry(f'{width}x{height}+{x}+{y}')

```

Рисунок 3.14 – Частина програмної реалізації ПЗ

Функція «center_window(window, width, height)» відповідає за обчислення координат, необхідних для відкриття вікна програми по центру екрана. Це важливо для поліпшення користувацького досвіду, оскільки вікно буде зручно розташоване на екрані.

11. Графічний інтерфейс програми, як показано на Рис. 3.15.

```

184 # Інтерфейс програми
185 root = tk.Tk()
186 root.title("Emotion Recognition App")
187
188 # Виклик функції для центрованого відкриття вікна
189 center_window(root)
190
191 img = None
192
193 # Віджети інтерфейсу
194 frame_top = tk.Frame(root)
195 frame_top.pack(pady=10)
196
197 lbl_language = tk.Label(frame_top, text="Language:")
198 lbl_language.pack(side=tk.LEFT)
199
200 combo_language = ttk.Combobox(frame_top, values=["English", "Українська"], state="readonly")
201 combo_language.set("English")
202 combo_language.pack(side=tk.LEFT)
203 combo_language.bind("<<ComboboxSelected>>", lambda event: set_language(combo_language.get()))
204
205 btn_select_image = tk.Button(root, text="Select Image", command=select_image)
206 btn_select_image.pack(pady=10)
207
208 lbl_image = tk.Label(root)
209 lbl_image.pack(pady=10)
210
211 btn_analyze = tk.Button(root, text="Analyze", command=analyze_emotions)
212 btn_analyze.pack(pady=10)
213
214 # Запуск інтерфейсу
215 update_ui()
216 root.mainloop()

```

Рисунок 3.15 – Частина програмної реалізації ПЗ

Цей блок створює основне вікно програми за допомогою «tkinter» та запускає цикл подій, що дозволяє користувачу взаємодіяти з інтерфейсом. У вікні розміщені кнопки для вибору зображення та аналізу емоцій, а також комбіновані списки для вибору мови.

12. Взаємодія з користувачем: включає в себе всі елементи управління, такі як кнопки для вибору зображення, кнопки для запуску аналізу емоцій, а також комбіновані списки для вибору мови. Кожен елемент інтерфейсу оптимізований для зручного використання, що дозволяє користувачеві швидко виконувати дії без зайвих зусиль.

13. Обробка помилок: у програмі реалізована обробка можливих помилок, що можуть виникнути під час виконання, таких як помилки при

завантаженні зображення чи аналізі. Ця функціональність забезпечує стабільність програми та запобігає її аварійному завершенню.

Повний вихідний код програмного забезпечення подано у Додатку А.

3.5 Тестування та оцінка ефективності

Перш за все, почнемо з того, що запустимо розроблене програмне забезпечення в нормальних умовах і з правильною взаємодією. Тож після запуску програми маємо відцентроване відносно екрану початкове вікно програми із вибором мови та кнопками вибору зображення й початку аналізу, як продемонстровано на Рис. 3.16-3.17.

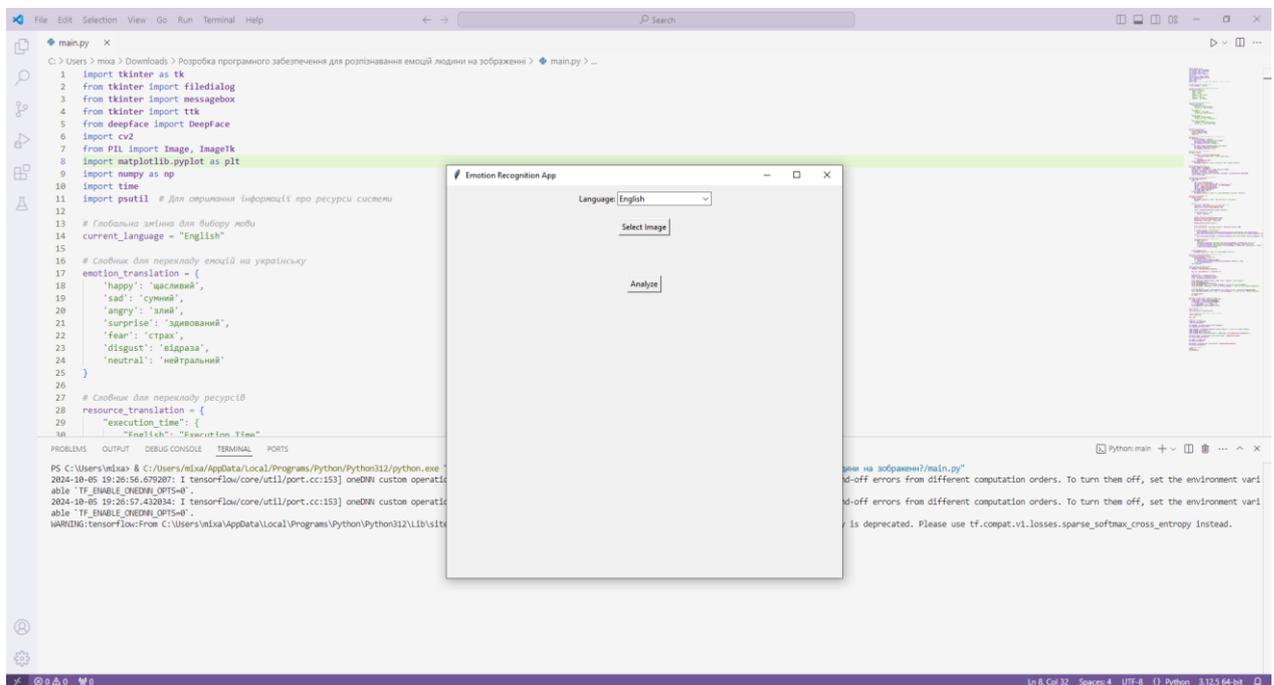


Рисунок 3.16 – Початкове вікно програмного забезпечення

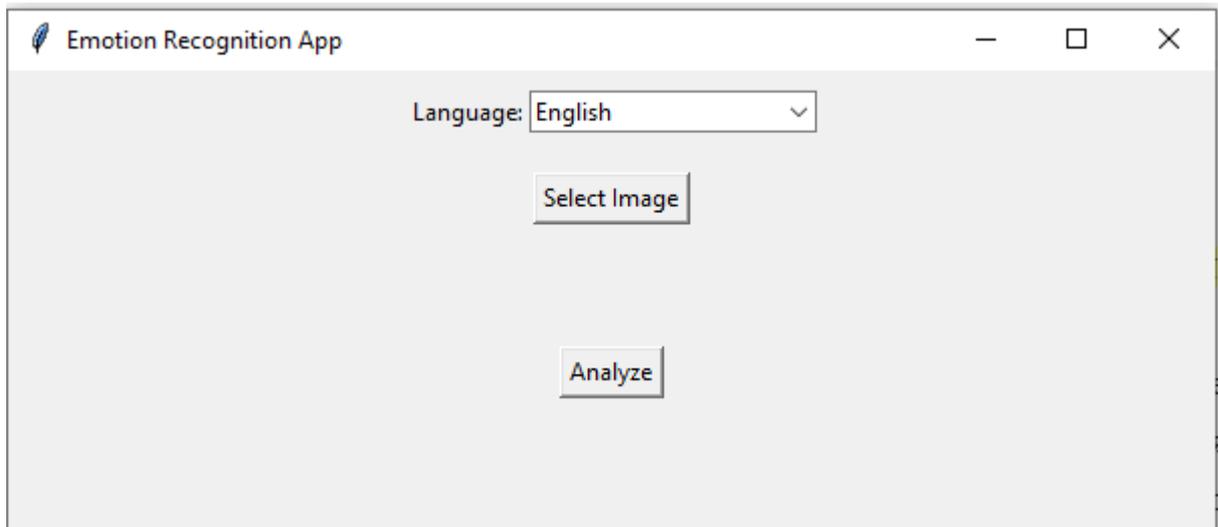


Рисунок 3.17 – Початкове вікно програмного забезпечення

Оберемо зображення для аналізу, натиснувши кнопку «Select image», як продемонстровано на Рис. 3.18.

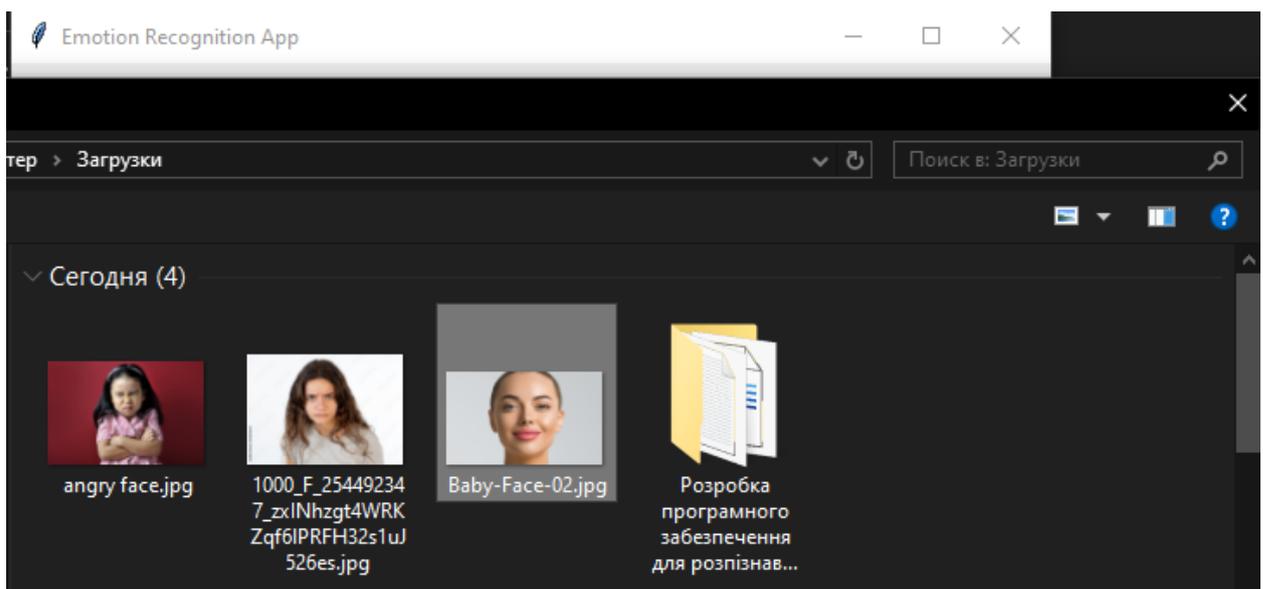


Рисунок 3.18 – Вікно вибору зображення для аналізу

Тепер програма має вигляд, як показано на Рис. 3.19. Обране зображення відображено в програмі й масштабовано так, щоб воно вмістилось у головне вікно програми.

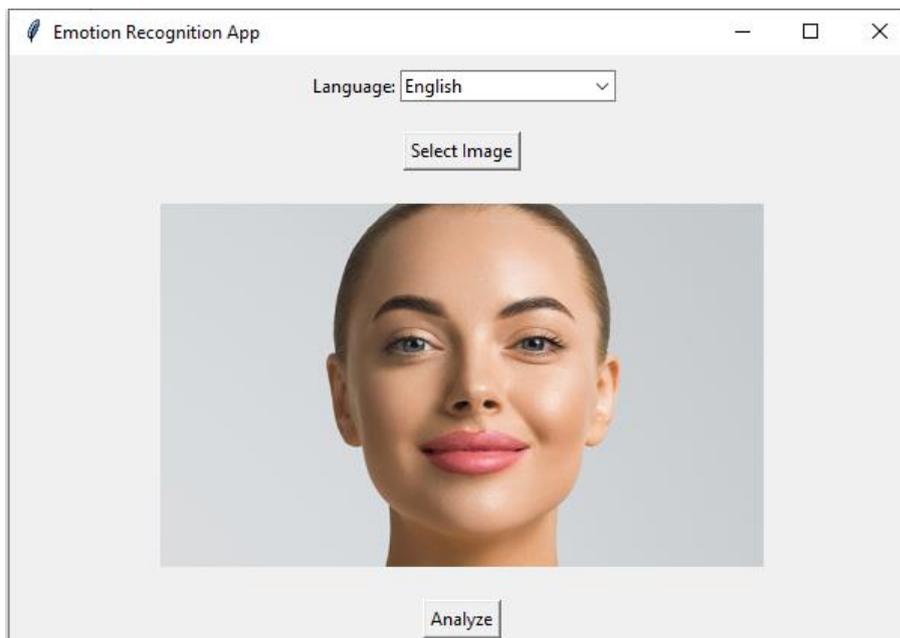


Рисунок 3.19 – Головне вікно програми

Тепер натиснемо кнопку «Analyze», щоб запустити етап аналізу програми. Результат роботи програми продемонстровано на Рис. 3.20-3.21.

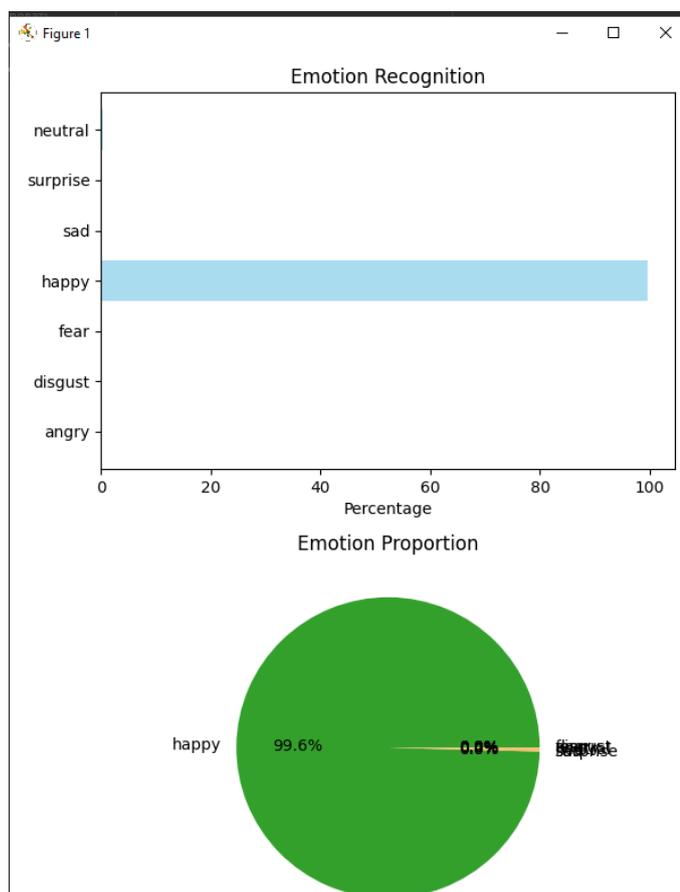


Рисунок 3.20 – Результат аналізу зображення програмою й вивід діаграм-результатів

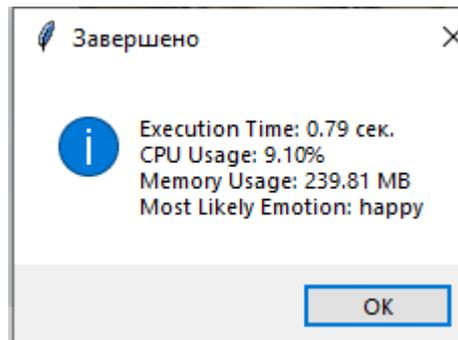


Рисунок 3.21 – Результат роботи програми й вивід інформації

Тож після аналізу зображення програма виводить дві діаграми (кругову й стовпчикову) із результатом розпізнавання емоцій. Після закриття вікна із діаграмами, виводиться повідомлення про результати роботи програмного забезпечення: час виконання, відсоток використання процесора, кількість використаної оперативної пам'яті і найбільш вірогідну емоцію людини із обраного зображення.

На Рис. 3.22-3.25 показано роботу програмного забезпечення на різних зображеннях.

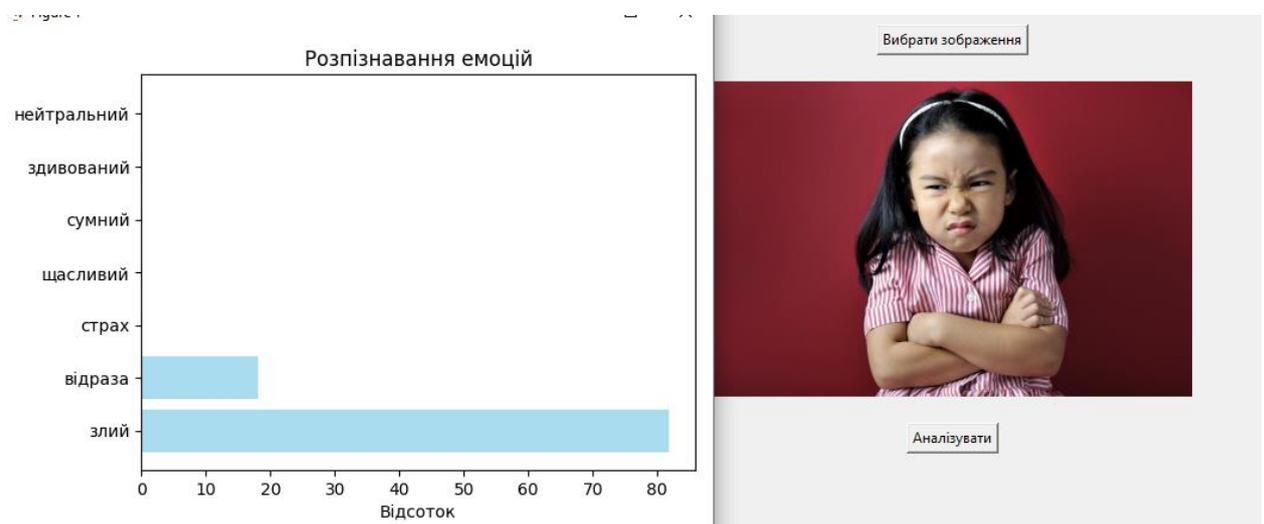


Рисунок 3.22 – Результат розпізнавання емоції програмним забезпеченням

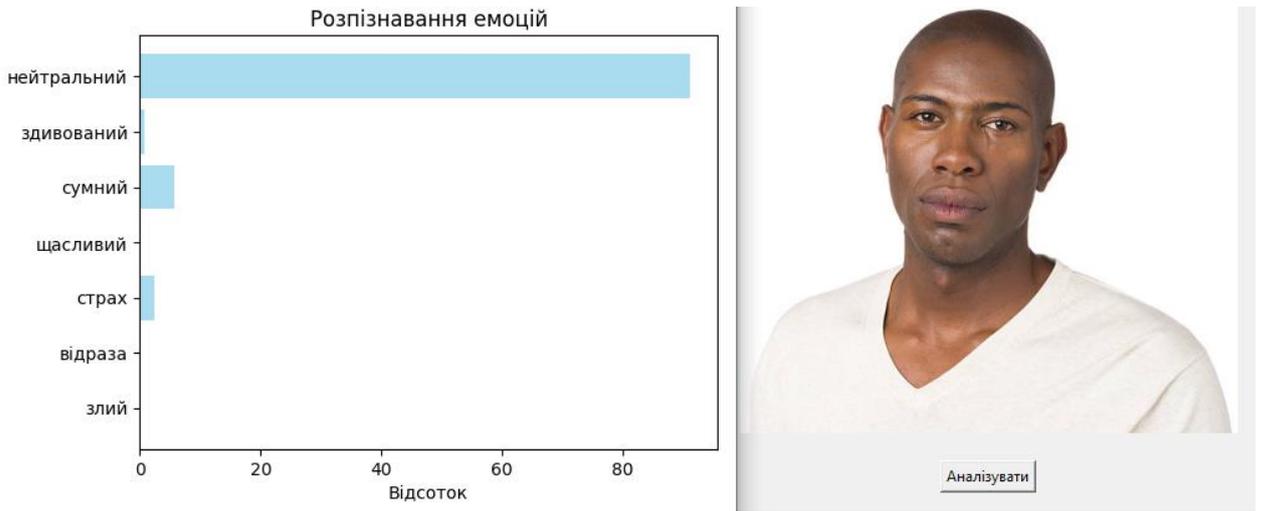


Рисунок 3.23 – Результат розпізнавання емоції програмним забезпеченням

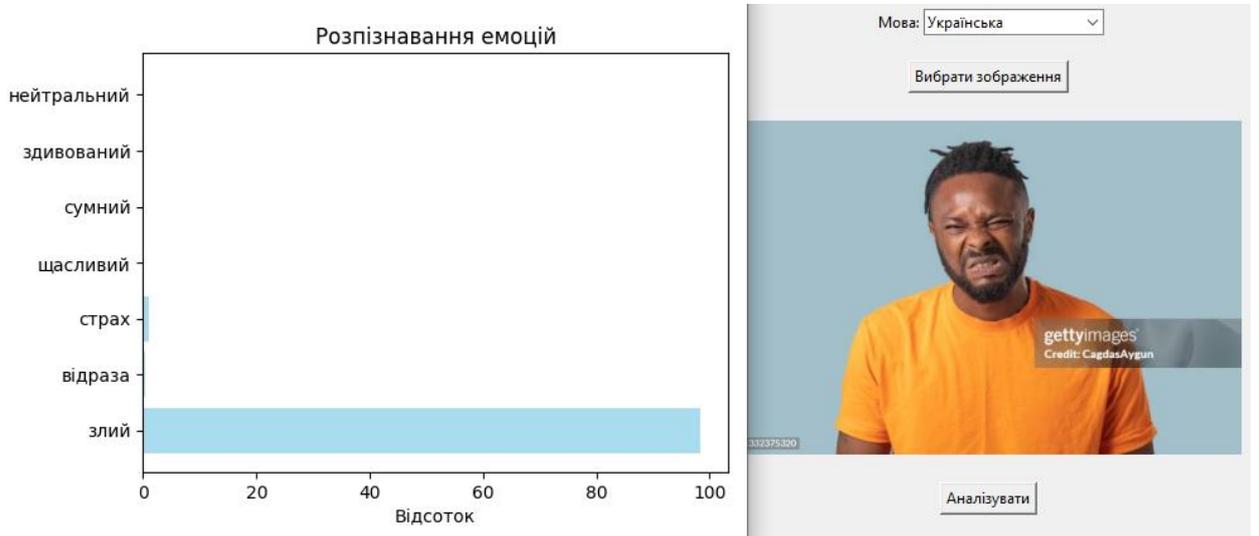


Рисунок 3.24 – Результат розпізнавання емоції програмним забезпеченням

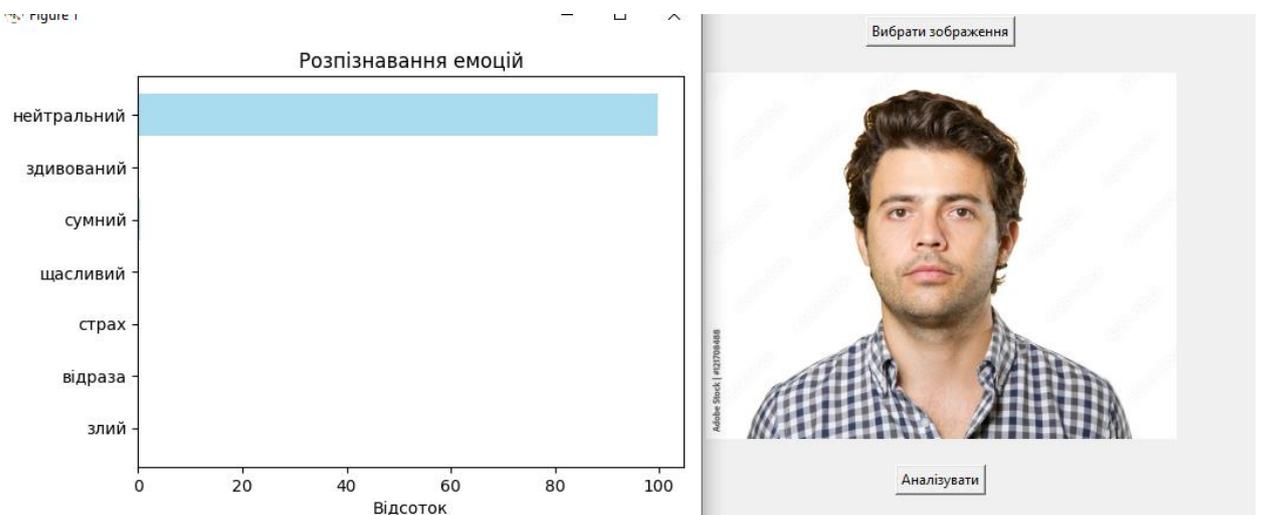


Рисунок 3.25 – Результат розпізнавання емоції програмним забезпеченням

Тестування програмного забезпечення проводитиметься за допомогою тест-кейсів у вигляді Таблиць 3.1-3.10. Тест-кейси – це детальні сценарії для перевірки роботи програмного забезпечення, які містять умови, кроки, очікувані результати і фактичні результати тестування. Вони допомагають виявити помилки, недоліки чи невідповідності в функціональності системи.

Кожен тест-кейс орієнтований на перевірку окремого аспекту програми, наприклад, коректність обробки вхідних даних, реакцію на помилки або перевірку роботи з інтерфейсом. Тест-кейси забезпечують систематичність тестування, підвищують якість ПЗ і гарантують відповідність специфікаціям.

Таблиця 3.1 – Тест-кейс для тестування програмного забезпечення

| Номер | 1 |
|----------------------|---------------------------------------------------------------------------------------------------|
| Назва | Завантаження зображення у програму |
| Опис | Перевірити, чи може користувач вибрати та завантажити зображення через діалогове вікно |
| Передумови | Програма відкрита, немає обраного зображення |
| Кроки | 1. Натиснути "Вибрати зображення" |
| | 2. Вибрати файл зображення |
| Очікуваний результат | Зображення завантажується та відображається в програмі, масштабоване відповідно до розмірів вікна |

Таблиця 3.2 – Тест-кейс для тестування програмного забезпечення

| Номер | 2 |
|----------------------|---------------------------------------------------------------------------------------------------------------------|
| Назва | Аналіз емоцій після вибору зображення |
| Опис | Перевірити, чи коректно аналізуються емоції після завантаження зображення |
| Передумови | Завантажено зображення обличчя |
| Кроки | 1. Натиснути "Аналізувати" після завантаження зображення |
| Очікуваний результат | Виводиться гістограма з емоціями, текстове повідомлення з найточнішою емоцією, час виконання та використані ресурси |

Таблиця 3.3 – Тест-кейс для тестування програмного забезпечення

| Номер | 3 |
|----------------------|------------------------------------------------------------------------|
| Назва | Зміна мови інтерфейсу програми |
| Опис | Перевірити коректність роботи функції зміни мови інтерфейсу |
| Передумови | Програма відкрита, мова за замовчуванням – англійська |
| Кроки | 1. Вибрати "Українська" в меню мов |
| | 2. Натиснути "Аналізувати" після завантаження зображення |
| Очікуваний результат | Інтерфейс, результати та повідомлення відображаються українською мовою |

Таблиця 3.4 – Тест-кейс для тестування програмного забезпечення

| Номер | 4 |
|----------------------|------------------------------------------------------------------------|
| Назва | Вибір некоректного файлу (не зображення) |
| Опис | Перевірити обробку помилок при виборі файлів, які не є зображеннями |
| Передумови | Програма відкрита, користувач намагається завантажити некоректний файл |
| Кроки | 1. Натиснути "Вибрати зображення" |
| | 2. Вибрати некоректний файл (не зображення) |
| Очікуваний результат | Виводиться повідомлення про помилку, зображення не завантажено |

Таблиця 3.5 – Тест-кейс для тестування програмного забезпечення

| Номер | 5 |
|----------------------|------------------------------------------------------------------------------------------------------------------|
| Назва | Виведення інформації про використані ресурси після аналізу |
| Опис | Перевірити, чи виводиться інформація про використані ресурси після завершення аналізу емоцій |
| Передумови | Завантажено зображення, проведений аналіз |
| Кроки | 1. Натиснути "Аналізувати" |
| | 2. Дочекатися завершення аналізу |
| Очікуваний результат | Виводиться інформація про час виконання, використані ресурси ЦП та пам'ять, переклад відповідно до вибраної мови |

Таблиця 3.6 – Тест-кейс для тестування програмного забезпечення

| Номер | 6 |
|----------------------|-----------------------------------------------------------------------------------|
| Назва | Перевірка підтримки форматів зображень (JPEG, PNG) |
| Опис | Перевірити, чи підтримує програма файли зображень у форматах JPEG і PNG |
| Передумови | Програма відкрита, користувач намагається завантажити файл у форматі JPEG або PNG |
| Кроки | 1. Натиснути "Вибрати зображення" |
| | 2. Вибрати файл у форматі JPEG або PNG |
| Очікуваний результат | Файл завантажується без помилок і відображається в програмі |

Таблиця 3.7 – Тест-кейс для тестування програмного забезпечення

| Номер | 7 |
|----------------------|------------------------------------------------------------------------------------------------------------------------------|
| Назва | Обробка помилки при завантаженні текстового файлу |
| Опис | Перевірити, чи обробляє програма помилку, коли користувач вибирає файл у форматі, відмінному від зображення (наприклад, TXT) |
| Передумови | Програма відкрита, користувач намагається завантажити текстовий файл |
| Кроки | 1. Натиснути "Вибрати зображення" |
| | 2. Вибрати текстовий файл (формат .txt) |
| Очікуваний результат | Програма виводить повідомлення про помилку, файл не завантажується |

Таблиця 3.8 – Тест-кейс для тестування програмного забезпечення

| Номер | 8 |
|----------------------|------------------------------------------------------------------------------------------------------------|
| Назва | Запуск аналізу без вибору зображення |
| Опис | Перевірити, чи обробляється помилка, якщо користувач спробує запустити аналіз емоцій без вибору зображення |
| Передумови | Програма відкрита, зображення не вибрано |
| Кроки | 1. Натиснути "Аналізувати", не вибравши зображення |
| Очікуваний результат | Програма виводить повідомлення про помилку, що зображення не вибрано |

Таблиця 3.9 – Тест-кейс для тестування програмного забезпечення

| Номер | 9 |
|----------------------|--------------------------------------------------------------------------------------------------------------------|
| Назва | Перевірка виведення результатів англійською мовою |
| Опис | Перевірити, чи правильно відображаються результати аналізу англійською мовою, коли вибрано англійську в інтерфейсі |
| Передумови | Мова інтерфейсу – англійська, завантажено зображення |
| Кроки | 1. Завантажити зображення |
| | 2. Натиснути "Analyze" |
| Очікуваний результат | Результати аналізу (назви емоцій, ресурси, час виконання) відображаються англійською мовою |

Таблиця 3.10 – Тест-кейс для тестування програмного забезпечення

| Номер | 10 |
|----------------------|--------------------------------------------------------------------------------------------------|
| Назва | Перевірка виведення тексту з найточнішою емоцією |
| Опис | Перевірити, чи правильно виводиться найточніша емоція в кінці аналізу |
| Передумови | Завантажено зображення для аналізу |
| Кроки | 1. Завантажити зображення |
| | 2. Натиснути "Аналізувати" |
| | 3. Дочекатися завершення аналізу |
| Очікуваний результат | Виводиться текст з найточнішою емоцією, перекладений на вибрану мову (англійська або українська) |

Проведене тестування програми за допомогою розроблених тест-кейсів підтвердило її стабільність та правильність роботи. Усі тест-кейси були успішно виконані, що свідчить про коректну обробку зображень, належну роботу з інтерфейсом на обох мовах, а також про здатність програми адекватно реагувати на некоректні дії користувача, такі як завантаження файлів неправильного формату або запуск аналізу без вибору зображення. Це забезпечує високу надійність і функціональність програми у практичному використанні.

Ефективність програми можна оцінити за кількома ключовими критеріями:

1. Швидкість виконання: важливо оцінити, наскільки швидко програма аналізує зображення та виводить результати. Це можна виміряти в секундах для різних розмірів зображень та обчислювальних ресурсів.

2. Використання ресурсів: ефективність роботи програми можна оцінити за допомогою аналізу використання ресурсів, таких як оперативна пам'ять і процесор. Важливо перевірити, як програма працює на різних типах комп'ютерів, зокрема з низькою продуктивністю.

3. Масштабованість: оцінка, як програма справляється з різними типами і розмірами зображень, а також її можливості обробки одночасно великої кількості даних.

Оцінку ефективності розробленого програмного забезпечення подано у вигляді Таблиці 3.5.11 та за допомогою діаграм залежностей характеристик у вигляді Рис. 3.26-3.27.

Таблиця 3.11 – Оцінка ефективності програмного забезпечення

| Ширина зображення, пікселів | Висота зображення, пікселів | Час виконання, с | Використання процесора, % | Використання оперативної пам'яті, Мб |
|------------------------------------|------------------------------------|-------------------------|----------------------------------|---------------------------------------------|
| 256 | 256 | 0,78 | 6,2 | 195 |
| 612 | 408 | 0,05 | 57 | 4 |
| 1000 | 665 | 0,13 | 35 | 91 |
| 1000 | 713 | 0,33 | 25 | 19 |
| 1920 | 864 | 0,65 | 23 | 200 |
| 3840 | 2160 | 2,82 | 20 | 1500 |

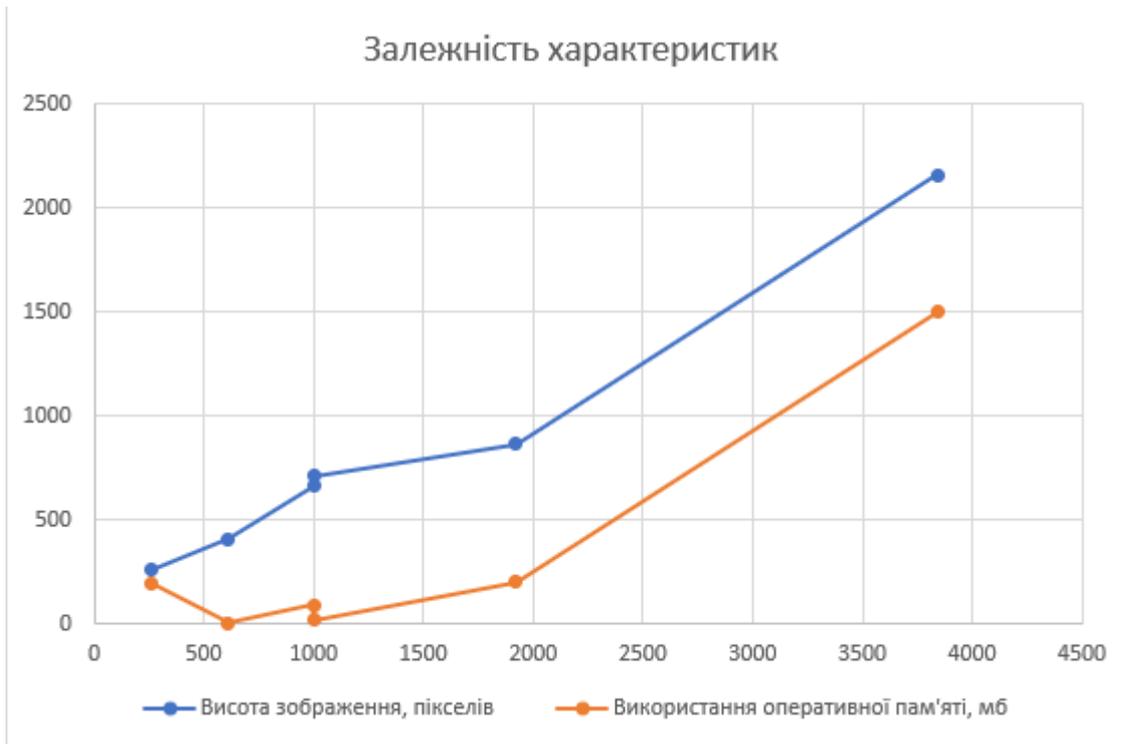


Рисунок 3.26 – Діаграма залежності висоти зображення та використання оперативної пам'яті

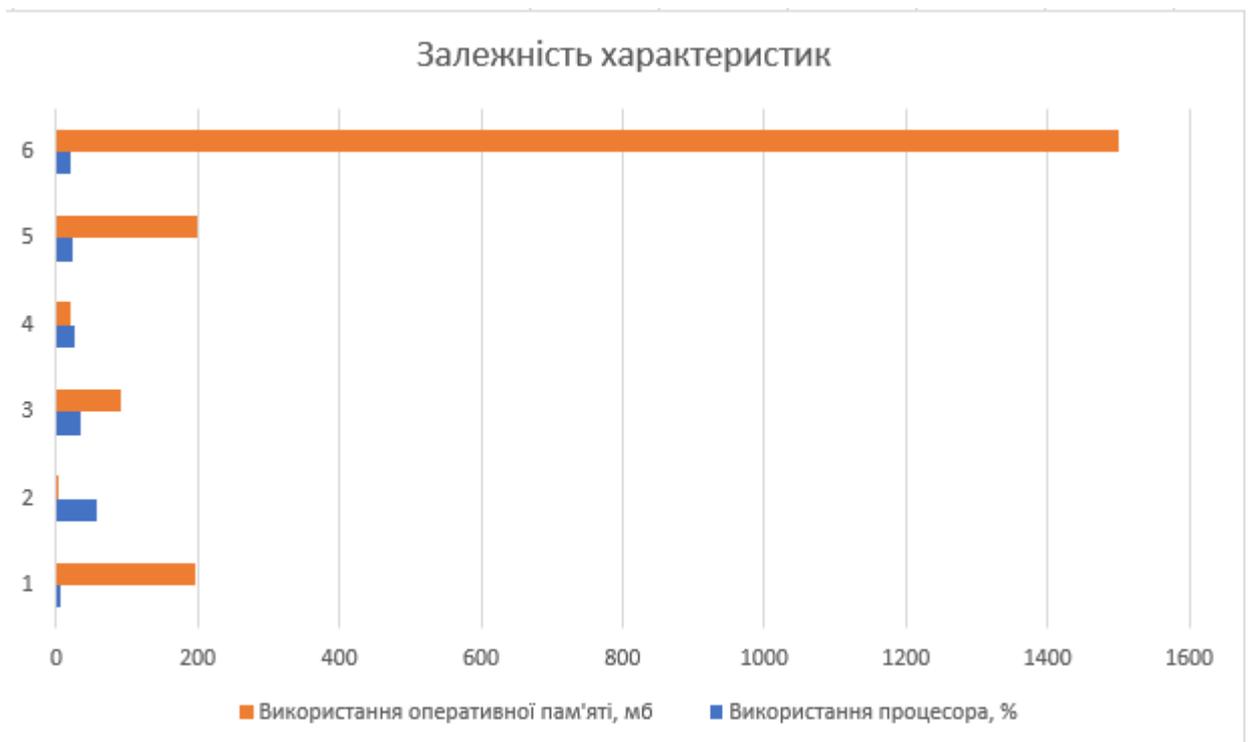


Рисунок 3.27 – Діаграма залежності використання процесора та використання оперативної пам'яті

1. Час виконання: середній час виконання для тестованих зображень становить 0.79 секунд. Це свідчить про те, що програма здатна швидко обробляти зображення невеликих і середніх розмірів, таких як 612x408 пікселів, де час виконання мінімальний (0.05 с). Однак для великих зображень (3840x2160 пікселів) час виконання суттєво зростає до 2.82 с, що є очікуваним для таких обсягів даних. Таким чином, продуктивність програми залишається на прийнятному рівні для більшості користувацьких сценаріїв, але для дуже великих зображень обробка займає більше часу.

2. Використання процесора: середнє використання процесора становить 27.7%. Для невеликих зображень (256x256 пікселів) процесор навантажується лише на 6.2%, що свідчить про ефективність роботи алгоритму в умовах мінімальних обчислювальних ресурсів. Однак для середніх розмірів зображень (612x408 пікселів) навантаження різко зростає до 57%, що може бути результатом збільшення кількості обчислень під час аналізу більш складних зображень. У випадку з найбільшими зображеннями (1920x864 і 3840x2160 пікселів), навантаження на процесор знижується до 23-20%, що може свідчити про оптимізацію розподілу обчислювальних ресурсів для великих файлів.

3. Використання оперативної пам'яті: використання оперативної пам'яті показує значну варіативність: середнє значення становить 334.83 МБ, однак при роботі з невеликими зображеннями (612x408 пікселів) програма споживає мінімум пам'яті – лише 4 МБ. З іншого боку, для обробки великих зображень (3840x2160 пікселів) використання оперативної пам'яті суттєво зростає до 2.1 ГБ, що є типовим для програм, які обробляють великі зображення. Це свідчить про те, що програма здатна адаптуватися до розмірів файлів, але для обробки великих даних необхідна значна кількість пам'яті.

Тож програма демонструє високу ефективність при роботі з зображеннями малих і середніх розмірів, забезпечуючи швидку обробку та оптимальне використання ресурсів. Проте для обробки зображень великої роздільної здатності (4К) спостерігається підвищене споживання пам'яті та

процесорних ресурсів, що слід враховувати при використанні програми на комп'ютерах з обмеженими можливостями. Загалом, програма залишається ефективною і продуктивною, демонструючи хороші показники швидкості і ресурсомісткості.

Комп'ютер, на якому проводилась розробка та тестування програмного забезпечення, має такі технічні характеристики:

Процесор AMD Ryzen 5 3600:

- Кількість ядер і потоків: 6 ядер і 12 потоків завдяки підтримці багатопоточності (Simultaneous Multithreading, SMT), що забезпечує високу продуктивність у багатозадачних операціях;
- Базова тактова частота: 3.6 ГГц, що дозволяє комфортно виконувати складні завдання;
- Максимальна тактова частота: 4.2 ГГц у режимі "Boost", що активується при збільшеній потребі в обчислювальній потужності;
- Техпроцес: 7 нм, що сприяє підвищеній енергоефективності та зниженню тепловиділення;
- Кеш-пам'ять: 32 МБ L3 кеша, що дозволяє швидко обробляти велику кількість даних;
- TDP: 65 Вт, що робить його енергоефективним процесором середнього рівня;
- Сокет: AM4, що робить процесор сумісним з багатьма сучасними материнськими платами [36].

Оперативна пам'ять 32 ГБ DDR4:

- Тип пам'яті: DDR4, що є сучасним стандартом для оперативної пам'яті;
- Об'єм: 32 ГБ, що забезпечує відмінну продуктивність під час запуску ресурсомістких програм, таких як відеоредактори, симулятори та віртуальні машини;
- Частота: 3000 МГц, що є швидкою тактовою частотою для оперативної пам'яті, забезпечуючи швидку передачу даних і зменшення

затримок під час обробки інформації;

- Двоканальний режим: Підтримка двоканального режиму збільшує швидкість передачі даних між пам'яттю і процесором, що робить систему більш ефективною.

Операційна система: Windows 10 LTSM.

Версія Python: 3.12.

ВИСНОВКИ

Під час виконання даної дипломної роботи було розроблене програмне забезпечення, що дозволяє ідентифікувати емоції людини на основі аналізу зображень за допомогою високорівневої мови програмування Python. Це програмне забезпечення має високу точність у різних умовах освітлення та ракурсах, а також має здатність працювати з різними типами емоцій та форматами зображень.

Були розглянуті такі теоретичні аспекти:

- Моделі емоцій;
- Традиційні методи розпізнавання емоцій;
- Огляд наявних наборів даних для розпізнавання емоцій;
- Основи комп'ютерного зору;
- Алгоритми глибокого навчання;
- Передобробка зображень;
- Виявлення обличчя;
- Класифікація емоцій нейронними мережами;
- Алгоритми класифікації.

З іншої сторони, у практичній частині були виконані такі етапи:

- Виконано огляд інструментів та технологій, необхідних для розробки програмного забезпечення;
- Розглянуто функціональні вимоги до ПЗ;
- Розглянуто нефункціональні вимоги до ПЗ;
- Реалізовано програмне забезпечення;
- Проведено тестування та оцінку ефективності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Models of emotion: the affective neuroscience approach [Електронний ресурс] – Режим доступу: [\[https://www.researchgate.net/publication/285907394_Models_of_emotion_the_affective_neuroscience_approach\]](https://www.researchgate.net/publication/285907394_Models_of_emotion_the_affective_neuroscience_approach)
2. Taylor, J. G., Korsten, N. Моделі емоцій на основі нейронних мереж // Sander, D., Scherer, K. R. (ред.). The Oxford Companion to Emotion and the Affective Sciences. Нью-Йорк: Oxford University Press, 2009
3. Тіздейл, Дж. Багаторівневі теорії відносин між когніцією та емоціями // Dalgleish, T., Power, M. (ред.). Handbook of Cognition and Emotion. Чичестер, Великобританія: Wiley, 1999
4. Тагард, П., Оубі, Б. Емоційна свідомість: нейронна модель взаємодії когнітивної оцінки та соматичного сприйняття для створення якісного досвіду // Consciousness and Cognition. 2008. Т. 17. С. 811–834
5. Томкінс, С. С. Афективна свідомість та зображення: Том II. Негативні афекти. Нью-Йорк: Springer, 1963
6. Тубі, Дж., Космідес, Л. До карти еволюційної функціональної організації розуму і мозку // Gazzaniga, M. S. (ред.). The New Cognitive Neurosciences. 2-е вид. Кембридж, Массачусетс: MIT Press, 2000
7. Emotion Recognition Using Different Sensors, Emotion Models, Methods and Datasets: A Comprehensive Review [Електронний ресурс] – Режим доступу: <https://www.mdpi.com/1424-8220/23/5/2455>
8. The Facial Emotion Recognition (FER-2013) Dataset for Prediction System of Micro-Expressions Face Using the Convolutional Neural Network (CNN) Algorithm based Raspberry Pi [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/347959965_The_Facial_Emotion_Recognition_FER-2013_Dataset_for_Prediction_System_of_Micro-Expressions_Face_Using_the_Convolutional_Neural_Network_CNN_Algorithm_based_Raspberry_Pi

9. AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild [Электронный ресурс] – Режим доступа: http://mohammadmahoor.com/wp-content/uploads/2017/08/AffectNet_oneColumn-2.pdf
10. Review on Emotion Recognition Databases [Электронный ресурс] – Режим доступа: https://www.researchgate.net/publication/321531813_Review_on_Emotion_Recognition_Databases
11. Datasets For Emotion Detection [Электронный ресурс] – Режим доступа: <https://analyticsindiamag.com/top-ai-tools/top-8-datasets-available-for-emotion-detection>
12. Розенфельд, А., Как, А. Digital Picture Processing. Academic Press, 1982
13. Баргаут, Л., Лі, Л. В. Perceptual Information Processing System. Патент США № 10/618,543, 2003
14. Ферхерст, М. С. Computer Vision for Robotic Systems. Prentice Hall, 1988
15. What Is Computer Vision? An Introduction [Электронный ресурс] – Режим доступа: <https://onlinedegrees.sandiego.edu/introduction-to-computer-vision>
16. Є, Дж. Ч. Geometry of Deep Learning: A Signal Processing Perspective. Mathematics in Industry (англ.). Сінгапур: Springer Nature, 2022
17. What Is Deep Learning? Definition and Techniques [Электронный ресурс] – Режим доступа: <https://www.linkedin.com/pulse/what-deep-learning-definition-techniques-examples-neil-sahota-%E8%90%A8%E5%86%A0%E5%86%9B>
18. The Complete Guide to Image Preprocessing Techniques in Python [Электронный ресурс] – Режим доступа: <https://medium.com/@maahip1304/the-complete-guide-to-image-preprocessing-techniques-in-python-dca30804550c>

19. Haar Cascades, Explained [Електронний ресурс] – Режим доступу: <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>
20. Deep Neural Network [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/topics/computer-science/deep-neural-network#:~:text=Deep%20neural%20networks%20are%20a,compared%20to%20conventional%20neural%20networks>
21. An Introduction to Convolutional Neural Networks (CNNs) [Електронний ресурс] – Режим доступу: <https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns>
22. Кінг, Д. Е. Dlib-ml: A Machine Learning Toolkit (PDF). J. Mach. Learn. Res., 2009
23. Цай, Дж. Л., Кнутсон, Б., Фунг, Х. Х. Культурні варіації в оцінці афектів // Journal of Personality and Social Psychology. 2006. Т. 90. С. 288–307
24. Вердюн, П., Ван Мехелен, І., Тюрлінкс, Ф. Взаємозв'язок між обробкою подій та тривалістю емоційного досвіду // Emotion. 2011. Т. 11. № 1. С. 20–28
25. Neural Networks for Emotion Classification [Електронний ресурс] – Режим доступу: <https://arxiv.org/pdf/1105.6014>
26. VGG Face Descriptor [Електронний ресурс] – Режим доступу: https://www.robots.ox.ac.uk/~vgg/software/vgg_face/
27. Теобальд, О. Machine Learning For Absolute Beginners. Лондон, Великобританія: Scatterplot Press, 2017
28. Теодорідіс, С. Machine Learning: A Bayesian and Optimization Perspective. Elsevier Ltd, 2020
29. Вольф, Р. 5 Types of Classification Algorithms in Machine Learning // MonkeyLearn Blog. 26 серпня 2020
30. Прабха, Р., Даршан, Г., Рошанрадж, К. Т., Сур'я Пракш, П. Б. Дослідження алгоритмів машинного навчання // International Journal of

Scientific Research in Computer Science, Engineering and Information Technology.

2021. Т. 7. № 4

31. Рамальо, Л. Python. До вершин майстерності. Fluent Python. O'Reilly, 2015

32. Рейтц, К., Шлюссер, Т. Автостопом по Python = The Hitchhiker's Guide to Python, 2017

33. Біслі, Д. М. Python Essential Reference. 4-е вид. Addison-Wesley Professional, 2009

34. Палах, Я. Parallel Programming with Python. Packt Publishing Ltd, 2014

35. Дауні, А. Б. Think Python: Як мислити як науковець-комп'ютерник. Версія 2.2.6. Кембридж: Cambridge University Press, травень 2012

36. Процесор AMD Ryzen 5 3600 (100-000000031) [Електронний ресурс] – Режим доступу: <https://hotline.ua/ua/computer-processor/amd-ryzen-5-3600-100-000000031/>

ДОДАТОК А

ВИХІДНИЙ КОД ОСНОВНИХ КОМПОНЕНТІВ

Файл main.py

```
import tkinter as tk
from tkinter import filedialog
from tkinter import messagebox
from tkinter import ttk
from deepface import DeepFace
import cv2
from PIL import Image, ImageTk
import matplotlib.pyplot as plt
import numpy as np
import time
import psutil # Для отримання інформації про ресурси системи

# Глобальна змінна для вибору мови
current_language = "English"

# Словник для перекладу емоцій на українську
emotion_translation = {
    'happy': 'щасливий',
    'sad': 'сумний',
    'angry': 'злий',
    'surprise': 'здивований',
    'fear': 'страх',
    'disgust': 'відраза',
    'neutral': 'нейтральний'
}

# Словник для перекладу ресурсів
resource_translation = {
    "execution_time": {
        "English": "Execution Time",
        "Українська": "Час виконання"
    },
    "cpu_usage": {
        "English": "CPU Usage",
        "Українська": "Використання ЦП"
    },
    "memory_usage": {
        "English": "Memory Usage",
        "Українська": "Використана пам'ять"
    },
    "most_likely_emotion": {
        "English": "Most Likely Emotion",
        "Українська": "Найточніша емоція"
    }
}
```

```

    }
}

# Функція для вибору мови
def set_language(lang):
    global current_language
    current_language = lang
    update_ui()

# Функція для оновлення UI відповідно до вибраної мови
def update_ui():
    if current_language == "English":
        btn_select_image.config(text="Select Image")
        btn_analyze.config(text="Analyze")
        lbl_language.config(text="Language:")
    else:
        btn_select_image.config(text="Вибрати зображення")
        btn_analyze.config(text="Аналізувати")
        lbl_language.config(text="Мова:")

# Функція для вибору зображення
def select_image():
    try:
        file_path = filedialog.askopenfilename(
            filetypes=[("Image files", "*.jpg *.jpeg *.png")]
        )
        if file_path:
            load_image(file_path)
    except Exception as e:
        messagebox.showerror("Помилка", f"Неможливо відкрити файл: {str(e)}")

# Масштабування зображення
def resize_image(img, max_size):
    height, width = img.shape[:2]
    scaling_factor = min(max_size / width, max_size / height)
    new_width = int(width * scaling_factor)
    new_height = int(height * scaling_factor)
    resized_image = cv2.resize(img, (new_width, new_height),
        interpolation=cv2.INTER_AREA)
    return resized_image

# Завантаження та відображення зображення
def load_image(image_path):
    global img
    try:
        img = cv2.imread(image_path)
        img_resized = resize_image(img, 400) # Обмеження розміру
        img_rgb = cv2.cvtColor(img_resized, cv2.COLOR_BGR2RGB)
        img_pil = Image.fromarray(img_rgb)
        img_tk = ImageTk.PhotoImage(img_pil)
        lbl_image.config(image=img_tk)

```

```

        lbl_image.image = img_tk
    except Exception as e:
        messagebox.showerror("Помилка", f"Помилка завантаження зображення:
{str(e)}")

# Функція для аналізу емоцій
def analyze_emotions():
    if img is None:
        messagebox.showerror("Помилка", "Спочатку виберіть зображення!")
        return

    try:
        start_time = time.time() # Початок вимірювання часу
        cpu_start = psutil.cpu_percent(interval=None)
        memory_start = psutil.virtual_memory().used

        result = DeepFace.analyze(img, actions=['emotion'])

        if isinstance(result, list):
            result = result[0]

        # Вимірювання витрат ресурсів
        cpu_end = psutil.cpu_percent(interval=None)
        memory_end = psutil.virtual_memory().used
        elapsed_time = time.time() - start_time

        display_emotions(result['emotion'])

        # Найточніша емоція
        most_likely_emotion = max(result['emotion'], key=result['emotion'].get)

        # Переклад тексту про найточнішу емоцію
        if current_language == "Українська":
            most_likely_emotion_text =
emotion_translation.get(most_likely_emotion, most_likely_emotion)
            most_likely_emotion_message =
f"{resource_translation['most_likely_emotion'][current_language]}:
{most_likely_emotion_text}"
        else:
            most_likely_emotion_message =
f"{resource_translation['most_likely_emotion'][current_language]}:
{most_likely_emotion}"

        # Повідомлення про використанні ресурси
        messagebox.showinfo(
            "Завершено",
            f"{resource_translation['execution_time'][current_language]}:
{elapsed_time:.2f} сек.\n"
            f"{resource_translation['cpu_usage'][current_language]}: {cpu_end -
cpu_start:.2f}%\n"

```

```

        f"{resource_translation['memory_usage'][current_language]}:
        {(memory_end - memory_start) / (1024 ** 2):.2f} MB\n"
        f"{most_likely_emotion_message}"
    )

    except Exception as e:
        messagebox.showerror("Помилка", f"Помилка аналізу: {str(e)}")

# Функція для перекладу емоцій на українську мову
def translate_emotions(emotions):
    if current_language == "Українська":
        translated_emotions = {}
        for emotion, value in emotions.items():
            translated_emotions[emotion_translation.get(emotion, emotion)] =
value
        return translated_emotions
    return emotions

# Відображення графіків з емоціями
def display_emotions(emotions):
    emotions = translate_emotions(emotions)

    fig, axs = plt.subplots(2, 1, figsize=(6, 8))

    # Гістограма
    emotions_list = list(emotions.keys())
    emotions_values = list(emotions.values())
    y_pos = np.arange(len(emotions_list))

    axs[0].barh(y_pos, emotions_values, align='center', alpha=0.7,
color='skyblue')
    axs[0].set_yticks(y_pos)
    axs[0].set_yticklabels(emotions_list)
    axs[0].set_xlabel('Відсоток' if current_language == "Українська" else
'Percentage')
    axs[0].set_title('Розпізнавання емоцій' if current_language == "Українська"
else 'Emotion Recognition')

    # Кругова діаграма
    axs[1].pie(emotions_values, labels=emotions_list, autopct='%1.1f%%',
colors=plt.cm.Paired.colors)
    axs[1].set_title('Співвідношення емоцій' if current_language == "Українська"
else 'Emotion Proportion')

    plt.tight_layout()
    plt.show()

# Функція для центрованого відкриття вікна
def center_window(window, width=600, height=600):
    screen_width = window.winfo_screenwidth()
    screen_height = window.winfo_screenheight()

```

```

x = (screen_width // 2) - (width // 2)
y = (screen_height // 2) - (height // 2)
window.geometry(f'{width}x{height}+{x}+{y}')

# Інтерфейс програми
root = tk.Tk()
root.title("Emotion Recognition App")

# Виклик функції для центрованого відкриття вікна
center_window(root)

img = None

# Віджети інтерфейсу
frame_top = tk.Frame(root)
frame_top.pack(pady=10)

lbl_language = tk.Label(frame_top, text="Language:")
lbl_language.pack(side=tk.LEFT)

combo_language = ttk.Combobox(frame_top, values=["English", "Українська"],
state="readonly")
combo_language.set("English")
combo_language.pack(side=tk.LEFT)
combo_language.bind("<<ComboboxSelected>>", lambda event:
set_language(combo_language.get()))

btn_select_image = tk.Button(root, text="Select Image", command=select_image)
btn_select_image.pack(pady=10)

lbl_image = tk.Label(root)
lbl_image.pack(pady=10)

btn_analyze = tk.Button(root, text="Analyze", command=analyze_emotions)
btn_analyze.pack(pady=10)

# Запуск інтерфейсу
update_ui()
root.mainloop()

```