

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка системи автоматизованого контент-аналізу
емоцій з використанням штучного інтелекту»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Штучний інтелект
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Аліна ХОТИНЕНКО

Виконала:
здобувачка вищої освіти
група ШД-41

Аліна ХОТИНЕНКО

Керівник:
ст.викладач

Тетяна КИСІЛЬ

Рецензент:

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Штучного інтелекту

Ступінь вищої освіти Бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Штучний інтелект

ЗАТВЕРДЖУЮ

Завідувач кафедри Штучного інтелекту

_____ Ольга ЗІНЧЕНКО

« _____ » _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

_____ Хотиненко Аліні Вікторівні _____

1. Тема кваліфікаційної роботи: Розробка системи автоматизованого контент-аналізу емоцій з використанням штучного інтелекту

керівник кваліфікаційної роботи Тетяна КИСІЛЬ,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» 02.2024р. № 36

2. Строк подання кваліфікаційної роботи «31» травня 2024р.

3. Вихідні дані до кваліфікаційної роботи: наукові статті на тему розпізнавання об'єктів на зображенні, відкриті бази сегментованих зображень осіб з різним емоційним станом, мова розробки додатка – Python.

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

Огляд сучасного стану досліджень

Технічні аспекти систем виявлення емоційних станів

Реалізація системи розпізнавання емоцій

5. Перелік графічного матеріалу: *презентація*

1. Структура штучної нейронної мережі
2. Особливості розпізнавання емоцій людини
3. Бібліотеки для реалізації проекту
4. Тестування проекту

6. Дата видачі завдання «27» лютого 2024 р

КАЛЕНДАРНИЙ ПЛАН

<i>№ з/п</i>	<i>Назва етапів кваліфікаційної роботи</i>	<i>Строк виконання етапів роботи</i>	<i>Примітка</i>
1	Аналіз наявної науково-технічної літератури	28.02-03.03.24	
2	Огляд методів вирішення задачі розпізнавання емоцій людини	04.03-17.03.24	
3	Аналіз існуючих аналогів	18.03-27.03.24	
4	Пошук та аналіз набору даних	29.02-10.03.24	
5	Вибір технологій для вирішення задачі	11.03-25.03.24	
6	Розробка і навчання моделі	26.03-13.04.24	
7	Програмна реалізація системи	14.04-16.05.24	
8	Розробка документації	17.05-31.05.24	

Здобувачка вищої освіти _____

Аліна ХОТИНЕНКО

Керівник

кваліфікаційної роботи _____

Тетяна КИСІЛЬ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: стор.59, табл. 1, рис. 12, джерел 20.

Мета роботи –підвищення ефективності роботи системи розпізнавання емоцій. Аналіз існуючих програм та виявлення наявних недоліків, щоб в подальшому удосконалити свою систему. За допомогою отриманих знань створити систему, яка розпізнає емоції.

Об'єкт дослідження – процес роботи системи розпізнавання емоцій осіб та підвищення їх якості.

Предмет дослідження – система розпізнавання емоцій людини з використанням штучного інтелекту.

Короткий зміст роботи: У роботі проведено огляд існуючих технологій розробки системи розпізнавання емоцій людини. В ході досліджень було вирішено обрати найбільш ефективний метод розпізнавання емоцій на фотографіях, шляхом їх вивчення та порівняння між собою. У дипломному проєкті розроблено систему, яка розпізнає такі емоції за виразом обличчя: злість, страх, щастя, спокій (нейтральний вираз), сум та здивування у режимі реального часу на відео з веб-камери. Для виявлення обличчя застосовано бібліотеку OpenCV, яка слугує для аналізу і обробки зображення обличчя у режимі реального часу на відео з веб-камери ноутбуку. Побудовано згорткову нейронну мережу для розпізнавання емоцій людини.

КЛЮЧОВІ СЛОВА: РОЗПІЗНАВАННЯ ЕМОЦІЙ, НЕЙРОННА МЕРЕЖА, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА.

ЗМІСТ

1 ОГЛЯД СУЧАСНОГО СТАНУ ДОСЛІДЖЕНЬ.....	10
1.1 Вивчення сучасних методів аналізу текстів	11
1.2 Аналіз психологічних станів за допомогою тексту	14
1.3 Аналіз існуючих програмних рішень.....	15
1.4 Загальні аспекти систем розпізнавання емоцій	23
1.5 Висновки до розділу	32
2 ТЕХНІЧНІ АСПЕКТИ СИСТЕМ ВИЯВЛЕННЯ ЕМОЦІЙНИХ СТАНІВ.....	33
2.1 Загальні поняття штучних нейронних мереж	33
2.2 Глибинне навчання нейронних мереж	36
2.3 Згорткові штучні нейронні мережі: загальні аспекти.....	36
2.4 Висновки до розділу	41
3 РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ.....	42
3.1 Вибір мови програмування для реалізації системи розпізнавання емоцій....	42
3.2 Аналіз пакетів обраної мови програмування Python.....	43
3.3 Сутність програмних засобів, використаних при реалізації проєкту	44
3.4 Програмна реалізація системи розпізнавання емоцій.....	53
3.5 Тестування створеного програмного забезпечення	53
3.6 Висновки до розділу	57
ПЕРЕЛІК ПОСИЛАНЬ	60
ДОДАТОК А. ПРОГРАМНИЙ КОД ПРОЄКТОВАНОЇ СИСТЕМИ	63
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	79

ВСТУП

Штучний інтелект став потужною силою змін у багатьох галузях. Зараз кількість контенту у світі швидко зростає, що створює виклик для традиційних методів аналізу.

Актуальність теми визначається рядом факторів. По-перше, швидке зростання обсягів інформації в Інтернеті та соціальних медіа створює потребу в ефективному і швидкому аналізі великої кількості даних. По-друге, бізнесу необхідно розуміти реакції своєї аудиторії на їхні продукти і послуги, щоб адаптувати маркетингові стратегії. По-третє, розвиток технологій машинного навчання дає можливість створювати дедуктивні та точні системи аналізу емоцій.

Для України ця тема також актуальна, оскільки зростаючі темпи діджиталізації суспільства створюють необхідність в ефективному аналізі великої кількості україномовного контенту в мережі.

Ручний аналіз настроїв у текстовому чи візуальному вмісті займає багато часу та є неефективним. Важливо розуміти людські емоції та поведінку, оскільки це є ключовим аспектом розвитку інтелектуальних систем. Одним із способів вирішення цієї проблеми є розробка систем, які розпізнають емоції за виразом обличчя. Такі системи можуть аналізувати міміку та виявляти емоції без втручання людини.

Метою дослідження є створення системи, яка автоматизовано аналізує емоційний відгук на текстовий, візуальний та аудіо-візуальний контент у онлайн середовищі за допомогою штучного інтелекту. Завдання включають розробку алгоритмів машинного навчання, створення бази даних для тренування та тестування, реалізацію програмного забезпечення, порівняння з існуючими рішеннями

Розпізнавання емоцій за виразом обличчя є складним завданням через індивідуальність вираження емоцій, що залежить від різних факторів. Тому при вираженні емоцій важливо враховувати різноманітні аспекти, такі як міміка

обличчя, положення очей, рухи губ та інші зміни фізіономії. Розгляд і аналіз цих аспектів допоможе вам отримати точніші результати.

Виявлено, що розпізнавання емоцій за виразом обличчя має велике значення у різних сферах життя, таких як психологія, медицина, реклама, розваги та безпека. Наприклад, психологічне дослідження допомагає виявити емоції та психічний стан людини. У медицині такі системи можна використовувати для виявлення патологічних станів, таких як депресія і тривога. У рекламі та розвагах розпізнавання емоцій допомагає аналізувати реакцію споживачів на рекламу та взаємодію з віртуальним середовищем. У сфері безпеки такі системи можуть виявляти та запобігати насильству, аналізуючи емоції людей у громадських місцях.

Об'єктом дослідження є процес аналізу емоційного відгуку на контент, що здійснюється в онлайн середовищі. Цей процес включає в себе автоматизовану обробку текстового, візуального та аудіо-візуального матеріалу з метою виявлення та класифікації емоцій, викликаних споживачами контенту. Основними характеристиками об'єкта є постійний потік інформації, яка генерується у соціальних мережах, новинах, форумах та інших джерелах, а також необхідність швидкого та точного аналізу цього обсягу даних з метою отримання корисної інформації для бізнесу, науки та суспільства загалом.

Отже, встановлено, що створення системи, яка може впізнавати емоції за виразом обличчя, наразі є актуальним та перспективним завданням. Сучасні методи глибокого навчання, такі як згорткові нейронні мережі, відкривають нові можливості для поліпшення точності та швидкості процесу розпізнавання емоцій. Основним завданням даної роботи є розробка надійної та точної системи, здатної миттєво розпізнавати людські емоції за виразом обличчя. Під час роботи розглядаються доступні методи та алгоритми розпізнавання емоцій. А сама модель базується на розробленій і навченій згортковій нейронній мережі.

Дане дослідження має велике значення для подальшого розвитку методик, пов'язаних з контент аналізом емоцій. Результати можна застосувати в різних сферах, відкриваючи нові можливості для соціального розвитку та покращення якості життя людей.

1 ОГЛЯД СУЧАСНОГО СТАНУ ДОСЛІДЖЕНЬ

В сучасному світі більшість людей має власні профілі у соціальних мережах. Багато хто використовує свої сторінки для висловлення власної думки. З цього всього, користь в випадку виявлення психологічного стану отримують кілька зацікавлених сторін.

Аналіз висловлювання пацієнта у соціальних мережах, дозволяє лікарю отримати додаткову інформацію про емоційний стан, психологічні проблеми та ставлення до хвороби, що допомагає у збагаченні анамнезу та точній діагностиці. Інтелектуальна система може автоматично виявляти ключові ознаки депресії, тривоги та інших психічних розладів у текстах пацієнтів.[20] Нейромережа може аналізувати висловлювання пацієнта без емоційного впливу, що дозволяє отримувати об'єктивні результати. Це може бути особливо корисно в ситуаціях, коли пацієнт не може чітко висловити свої симптоми або коли їхні слова можуть бути спотворені емоціями або стресом. Керуючись цими даними, людина розуміє, коли потрібна додаткова підтримка або консультація.[19]

Кожен спеціаліст хоче якомога краще розуміти клієнтів, їх настрої, потреби та думки – це дозволить покращити якість лікування та зменшити виникнення серйозних захворювань.[7]

На сьогоднішній день в використанні комп'ютерних методів для аналізу текстових даних маємо деякі ключові напрямки та досягнення:

1. Обробка природної мови (Natural Language Processing, NLP): Удосконалення технологій для роботи з природною мовою дозволяє автоматично аналізувати та розуміти текстові дані. Використання методів NLP включає токенізацію, лематизацію, визначення частин мови, виявлення іменованих сутностей тощо.

2. Машинне навчання та глибоке навчання: Методи машинного навчання, такі як класифікація та кластеризація, використовуються для автоматичної обробки та аналізу текстових даних. Глибокі нейронні мережі дозволяють створювати складніші моделі для прогнозування та класифікації.

3. Використання соціальних медіа та Інтернету: Дослідження активно

використовують дані з соціальних медіа та Інтернету для аналізу психологічних станів. Це дозволяє отримувати великий обсяг даних, які відображають реальні висловлювання та емоції користувачів.

4. Розвиток додатків для моніторингу психічного здоров'я: З'являються додатки, які використовують аналіз текстів для моніторингу психічного здоров'я користувачів. Вони можуть виявляти зміни в настрої, емоціях та поведінці, що може бути корисним для самостійного виявлення проблем та підтримки.[18]

1.1 Вивчення сучасних методів аналізу текстів

Векторизація слів, тематичне моделювання та машинне навчання в аналізі тексту - це ключові методи, які дозволяють отримувати цінну інформацію з текстових даних:

Векторизація слів - цей процес полягає в перетворенні текстових даних у вектори числових значень, які можуть бути оброблені комп'ютером. Найпоширенішим методом векторизації є "bag-of-words".[1]

Мішок слів (Bag of Words) – це метод отримання характеристик з тексту для моделювання. Цей підхід простий і гнучкий, що дозволяє легко його змінювати та адаптувати. Основна концепція моделі полягає у представленні тексту як вектора, що відображає частоту входження кожного слова. У цьому методі не враховується інформація про порядок чи структуру слів. Його називають мішком слів, оскільки він лише визначає наявність слова в документі, не враховуючи його місцезнаходження.[6]

Для застосування цієї моделі потрібно:

- сформувані словник відомих токенів (слів);
- визначити ступінь їх присутності.

Інші методи включають TF-IDF (term frequency-inverse document frequency) та Word2Vec.

У сфері машинного навчання та обробки природної мови, тематична модель використовується для статистичного аналізу текстової інформації з метою визначення абстрактних "тем", які присутні в колекції документів.

Тематичне моделювання —це процес аналізу текстових даних, спрямований на виявлення основних тем або концепцій, які присутні в наборі документів, дозволяє нам ефективно аналізувати великі обсяги тексту, групуючи документи в групи. Один із найпопулярніших алгоритмів тематичного моделювання - це Latent Dirichlet Allocation (LDA). LDA розглядає кожен документ як суму різних тем і намагається знайти невидимі "латентні" теми, які краще пояснюють набір документів.

Машинне навчання включає в себе використання алгоритмів, які навчаються на основі набору даних, щоб розуміти, класифікувати або прогнозувати властивості тексту. Наприклад, класифікація текстів за емоційним тоном (позитивний, негативний, нейтральний), виявлення ключових тем або категорій, автоматичне розпізнавання іменованих сутностей тощо. [17]

Загалом існують два типи машинного навчання: індуктивне навчання, або навчання за прецедентами, та дедуктивне навчання. Оскільки дедуктивне навчання зазвичай відносять до експертних систем, терміни «машинне навчання» і «навчання за прецедентами» часто вважаються синонімами.

Загалом, традиційні методи аналізу текстів мають свої переваги та обмеження. Вони дозволяють швидко та ефективно виконувати базовий аналіз тексту, але можуть бути не досить точними у випадку складних або невідомих структур текстів.

Вибір ідеальної системи обробки природної мови (NLP) є критично важливим рішенням для компаній і розробників, які працюють над завданнями, пов'язаними з мовою. Дві популярні фреймворки NLP — spaCy і TensorFlow, кожна з яких пропонує унікальний набір функцій, переваг і випадків використання. Також зараз існує безліч інтелектуальних систем, які використовуються для аналізу текстової інформації. Розглянемо можливості SpaCy і TensorFlow.

Призначення SpaCy. SpaCy - це програмна бібліотека для обробки природної мови (NLP), що надає функціонал для розпізнавання та аналізу тексту. Розроблений з урахуванням промислових випадків, він швидкий, ефективний і легко інтегрований у ваші проекти. [3]

Деякі ключові функції SpaCy включають:

- попередньо навчені моделі;
- розширені функції з коробки;
- блискавична обробка.

SpaCy пропонує менші, попередньо навчені моделі, які оптимізовані для різних завдань NLP, таких як токенізація, тегування частин мови та розпізнавання іменованих об'єктів.

Функції вбудовані в бібліотеку без необхідності додаткового програмного коду, SpaCy надає більш розширені та точні функції без потреби значного налаштування.

Продуктивність є ключовим напрямком розробки SpaCy, що робить його придатним для великомасштабних застосувань.

Призначення Tensor Flow. TensorFlow, розроблений Google Brain, — це бібліотека машинного навчання з відкритим кодом, яка широко використовується для різноманітних програм штучного інтелекту (ШІ). Хоча TensorFlow має ширші варіанти використання, ніж SpaCy, він також часто використовується для завдань НЛП. [3]

Деякі важливі моменти щодо TensorFlow включають:

- гнучкість : TensorFlow можна використовувати для різних завдань, від обробки й аналізу даних до моделей глибокого навчання для розпізнавання зображень і розуміння природної мови.
- масштабованість : TensorFlow розроблено для масштабування від невеликих моделей, що працюють на мобільних пристроях, до складних багаторівневих нейронних мереж, що працюють у розподілених системах.
- підтримка спільноти : TensorFlow має величезну екосистему з численними

інструментами, бібліотеками та посібниками, доступними для підтримки вашої роботи в машинному навчанні та ШІ.

Підводячи підсумок, SpaCy розроблено спеціально для НЛП і відрізняється обробкою, ефективністю та простотою використання, особливо для промислових застосувань. TensorFlow, з іншого боку, — це універсальна бібліотека, зосереджена на машинному навчанні та штучному інтелекті, яка включає NLP як один із багатьох випадків використання. Вибираючи ідеальну структуру NLP для ваших проєктів, враховуйте конкретні потреби вашої програми та переваги, які пропонує кожна бібліотека.

1.2 Аналіз психологічних станів за допомогою тексту

Sentiment Analysis - це процес аналізу тексту з метою визначення та класифікації емоційного тону або настрою, що виражений у ньому. Ця технологія аналізує створений користувачами контент, такий як текстові коментарі, твіти, огляди та відгуки клієнтів, часто класифікуючи тексти на позитивні, негативні або нейтральні емоційні категорії.

Аналіз настроїв широко використовується для розуміння й оцінки емоційної реакції людей на продукти, послуги, бренди чи події. Компанії можуть відстежувати задоволеність клієнтів, розуміти сприйняття бренду та визначати стратегії розвитку продукту, проводячи аналіз соціальних мереж. Крім того, його можна використовувати в політичних опитуваннях або аналізі ЗМІ для розуміння суспільних емоційних тенденцій.[2]

Основні кроки аналізу настроїв:

- Збір даних: текстові дані збираються для аналізу. Ці дані можуть включати відгуки, твіти або коментарі клієнтів, серед іншого.
- Попередня обробка тексту: текстові дані очищаються та обробляються.
- Етапи попередньої обробки можуть включати перетворення тексту на нижній регістр, видалення спеціальних символів, видалення знаків пунктуації та видалення стоп-слів (тобто безглуздих слів).

- *Вилучення функцій*: текстові дані перетворюються на числові функції. Цей крок передбачає структурування тексту в числові вектори, які можуть бути використані алгоритмами машинного навчання.

- *Машинне навчання*: модель машинного навчання навчається на функціях, отриманих із навчальних даних. Алгоритми класифікації можуть бути використані для прогнозування емоційних категорій (позитивні, негативні, нейтральні) тексту.

- *Оцінка результату*: навчена модель оцінюється на основі даних тестування, а результати оцінюються за допомогою таких показників, як точність, точність і запам'ятовування.[15]

Аналіз настроїв використовується в різних сферах:

а) маркетинг і бренд-менеджмент: компанії можуть зрозуміти емоційну реакцію на продукти чи послуги та керувати репутацією бренду, відстежуючи відгуки клієнтів.

б) аналіз соціальних медіа: Компанії можуть оцінити задоволеність клієнтів і запобігти потенційним кризам, розуміючи, що користувачі говорять про їхні продукти чи послуги на платформах соціальних мереж.

в) дослідження ринку: Емоційні дані, зібрані за допомогою опитувань або коментарів клієнтів, можна використовувати для розуміння споживчих тенденцій і ринкових уявлень.

г) аналіз ЗМІ та політики: Аналіз настроїв можна використовувати в політичних опитуваннях або аналізі ЗМІ, щоб зрозуміти емоційні тенденції суспільства.

Аналіз настроїв є потужним інструментом для розуміння емоційних реакцій користувачів і широко використовується в різних галузях для підвищення рівня задоволеності клієнтів і розробки ефективних стратегій.

1.3 Аналіз існуючих програмних рішень

Активна розробка найперспективніших рішень відкритого доступу почалася лише в 2020 році. Оскільки платні продукти для бізнесу надають обмежену

інформацію звичайним користувачам, здійснити їх детальний аналіз було доволі складно.

Існує широкий вибір рішень на ринку для ідентифікації облич. Деякі з цих рішень не вимагають спеціальних навичок під час використання з методами машинного навчання, тоді як інші потребують певних знань та зусиль.

Можна виділити три ключові категорії послуг розпізнавання обличь, кожен з яких має свої переваги та недоліки:

- Програмне забезпечення для виявлення облич (software as a service - SaaS). Провайдер послуг ідентифікації облич зобов'язується відповідати за управління та технічну допомогу високонавантажених серверів, а також слідкує за новітніми технологічними розробками машинного навчання.[12]

- Користувачу потрібно інтегрувати додаток зі своїми ІТ-системами через API-інтерфейс. Ці рішення пропонують широкі можливості, але мають свої проблеми.

По-перше, такі сервіси є дорогими, оскільки постачальник бере на себе всі аспекти обслуговування. Крім того, вам необхідно забезпечити безперервний доступ до глобальної мережі, оскільки зображення потрібно відправляти на сервер. Також, ви надаєте особисті дані третій стороні і не контролюєте, як вони використовуються, тому є певні області щодо безпеки, які можуть викликати побоювання.

- Варіанти REST API, розміщені на власному хостингу. Ці технології можуть бути розгорнуті як локально, так і в хмарі. Вони позбавлені типових проблем, властивих SaaS-продуктам. Ви маєте можливість контролювати місце зберігання даних, розміщуючи їх на своїх серверах або в приватній хмарі, а також можете створити офлайн-рішення. Однак, ви будете змушені власноруч займатися адмініструванням серверів. Зазвичай сервери працюють у контейнерах Docker, що полегшує їх організацію. Хоча самостійне розміщення коштує менше, ніж використання SaaS, воно все одно залишається досить витратним. На щастя, створюються різні рішення самостійного розміщення REST API з публічними

ресурсами. Вони виглядають досить перспективно, хоча ще не настільки розвинені, як альтернативні підходи.

- Фреймворки та бібліотеки з відкритим вихідним кодом. Зазвичай вони доступні безкоштовно, оскільки багато дослідників прагнуть ділитися вихідним кодом власних передових підходів. Для використання цього типу програмного забезпечення вам, безумовно, знадобляться певні знання з машинного навчання. Якщо ви хочете інтегрувати ці рішення з вашим власним програмним забезпеченням, доведеться докласти зусиль для його RESTифікації. Однак головна перевага в тому, що перед вами буде високоякісна технологія, з якою ви вже маєте певний досвід.

Розуміння власних можливостей - важлива річ. Колись, використання можливості виявлення облич серед звичайних людей була надзвичайно дорога. Цінова політика з розпізнавання облич для відео в реальному часі становить від \$86,40 на день і може перевищувати \$30,000 на камеру на рік. Тому, в основному, дана технологія набуває популярності лише у разі її впровадження державними установами або високотехнологічними підприємствами.

Однак, з появою все більшої кількості безкоштовних варіантів, це вже не є такою значною проблемою, як раніше. Незважаючи на те, що вони перебувають на різних етапах розвитку, малі та середні підприємства вже можуть їх використовувати. Ці технології більше не призначені лише для нішевих ентузіастів. Точність була головним критерієм при виборі програмного забезпечення для розпізнавання облич. Рішення мають змогу продемонструвати свої високі стандарти за допомогою різноманітних тестів.

Кожного року на ринку з'являються нові технології розпізнавання облич, які встановлюють нові стандарти продуктивності. Через це порівнювати минулі рішення з новітніми розробками може викликати складність. Проте є один дуже давній, але досі широко застосовуваний бенчмарк: Labeled Faces in the Wild (LFW).

1. *Ageitgey/face_recognition (GitHub Repository)*. Отримавши 45 тисяч схвальних відгуків на GitHub, ця бібліотека, ймовірно, є найбільш популярною для

безкоштовного розпізнавання облич. Ви можете вибрати між використанням їх API на Python або їх бінарної програми командного рядка. Встановлення підтримується на всіх основних платформах, і навіть є Docker-образ для швидшої інсталяції. Попри те, що бібліотека має значну популярність, варто врахувати кілька суттєвих недоліків, якщо ви вирішите її використовувати. По-перше, хоча репозиторій все ще оновлюється, з 2018 року не було значних змін (а три роки у сфері ШІ – це великий проміжок). По-друге, алгоритм розпізнавання облич трохи застарілий і забезпечує точність лише 99,38% на LFW. Крім того, інтеграція з цим додатком може бути складною, оскільки він не має REST API.

2. **CompreFace**. Це рішення з'явилося на GitHub у липні 2020 року і наразі має близько 2000 зірок, але воно здається досить перспективним. CompreFace: це відкрите програмне забезпечення з відкритим вихідним кодом, розроблене компанією Exadel. Його можна легко інтегрувати в різноманітні додатки та системи завдяки простому API. CompreFace використовує передові алгоритми машинного навчання для досягнення високої точності розпізнавання облич і працює з різними типами зображень, включаючи фотографії в реальному часі. Він також забезпечує можливість роботи в режимі офлайн та масштабування для великих обсягів даних. Загалом, це розширений інструмент розпізнавання облич, який може використовуватися в різноманітних сценаріях, включаючи системи безпеки, відеоспостереження, біометричні системи контролю доступу та багато іншого. CompreFace пропонує розширений функціонал для ідентифікації осіб, включаючи визначення віку, статі, емоційного стану тощо.

3. **DeepFace**. Ця бібліотечна структура, яка має спільну назву з алгоритмом розпізнавання облич Facebook DeepFace і була завантажена на GitHub у лютому 2020 року, має щонайменше 4100 відгуків. Додатково платформа включає інші алгоритми розпізнавання. Більше того, REST API, яке можна використовувати, обмежується перевіркою методів і не має функціоналу для створення колекцій облич та пошуку вашого обличчя серед них.

Якщо ви знаєте мову Python, почати працювати досить легко, але для новачків це може бути складно.

4. **FaceNet**. Ця система базується на концепції вбудованого векторного простору, де кожне обличчя представлене у вигляді унікального вектора, який знаходиться в просторі. Вона використовує глибокі нейронні мережі для навчання моделі, яка здатна генерувати подібні вектори для однакових облич та віддалені вектори для різних облич. Це дозволяє ефективно використовувати цю систему для ідентифікації та верифікації осіб у реальному часі.

FaceNet забезпечує можливість створення векторного представлення для кожного обличчя, яке потім може бути використане для порівняння облич та визначення їхньої ідентичності. Бібліотека Python з відкритим кодом на основі FaceNet була розроблена експертами Google для розпізнавання облич. Багато посібників використовують репозиторій з 12 600 відгуками як головну бібліотеку. Цей алгоритм має досить високу ефективність, що, звісно, не погано, але не дотягує до найкращих результатів. Однак, він має кілька недоліків: відсутність REST API та відсутність оновлення вже тривалий час.

5. **InsightFace**. Ще одним публічно доступним пакетом на мові Python, який набрав 12,1 тисячі відгуків, є InsightFace. Цей пакет базується на RetinaFace, одній з найсучасніших і найнадійніших систем розпізнавання та ідентифікації облич (SubCenter-ArcFace). Цей репозиторій залишається дуже актуальним. На наборі даних LFW алгоритм досягає точності 99,86%. Основним недоліком поки є складність його використання. Розгляньте CompreFace та InsightFace-REST, якщо вам потрібні рішення, що використовують InsightFace, пропонують зручний REST API і можуть працювати у докерних контейнерах.

6. **InsightFace-REST**. Хоча цей багатообіцяючий репозиторій був створений ще в 2019 році, активний розвиток розпочався лише в жовтні 2020 року. Це програмне середовище на базі Docker, яке пропонує зручний REST API, подібний до CompreFace. Основною перевагою є триразове збільшення швидкості розпізнавання у порівнянні з InsightFace. Однак є кілька недоліків. По-перше, вам

доведеться розробити власний класифікатор, оскільки репозиторій надає лише вбудовування облич і не має API для безпосереднього розпізнавання осіб. По-друге, репозиторій поки що не має власної ліцензії, тому для його використання потрібно отримати дозвіл від творця.

Також використовують програмне забезпечення для розпізнавання облич (в алфавітному порядку):

1. **Amazon Rekognition.** Це послуга обробки зображень та відео, надана компанією Amazon Web Services (AWS). Вона надає можливості для розпізнавання облич, аналізу настроїв та емоційного стану на зображеннях, виявлення та аналізу об'єктів, а також ідентифікації та визначення діяльностей на відео. Amazon Rekognition може використовуватися в різних сферах, включаючи системи безпеки, відеоспостереження та аналізу зображень у медицині. Зазначений ресурс використовували різні урядові установи США, такі як Міграційна та митна служба США та підприємства приватного сектору. Rekognition має дві основні можливості: алгоритми, навчені на даних Amazon, і алгоритми, які користувачі можуть навчати на своїх власних наборах даних.

До класу навчальних моделей належать такі можливості:

- визначення відомих людей на зображеннях;
- розпізнавання облич із встановленням гендеру, вікової групи, емоційного стану, наявності бороди або вусів, окулярів для зору чи сонцезахисних окулярів, а також визначення усмішки;
- візуальний трекінг осіб на відеозаписі, скажімо, для розбору дій учасників гри на стадіоні після матчу;
- виявлення символів на знімках;
- ідентифікація неприйняттого або недоречного контенту.

2. **Deep Vision AI.** Наявні SaaS-версії та версії Deep Vision AI для розміщення на власному хостингу. Втім, відсутність цін на веб-сайті, ймовірно, означає, що послуги орієнтовані лише на корпоративний сегмент. Відомостей про

підтримувані мови програмування для SDK теж відсутні. Окрім ідентифікації облич, є інструменти для встановлення гендеру та вікової групи.

3. **FaceFirst.** Є світовим лідером у високоефективних системах зіставлення облич для роздрібних торговців, лікарень, казино, аеропортів, стадіонів і арен. Програмне забезпечення FaceFirst використовує штучний інтелект і людський нагляд для запобігання насильству, крадіжкам і шахрайству. За допомогою FaceFirst можна створити безпечніше середовище для своїх цінних клієнтів, пацієнтів, гостей, співробітників і партнерів.

4. **Face++.** Компанія була заснована у 2011 році групою китайських підприємців. З того часу вона стала однією з провідних у світі компаній у сфері розпізнавання облич і машинного зору. Нестандартні обмеження безкоштовної програми Face++ означають, що не можна гарантувати стабільну роботу протягом тривалого часу. Компанія надає необмежену кількість запитів, але лише три на секунду, які рівномірно діляться між усіма споживачами на безоплатному тарифному плані. Крім того, слід врахувати, що платна підписка вдвічі вища порівняно з іншими провідними компаніями, такими як Amazon і Microsoft, які встановлюють плату 1 долар за кожну тисячу розпізнавань.

Face++ включає різноманітні додаткові сервіси, зокрема, розпізнавання емоцій і статі, виявлення ключових точок обличчя та визначення віку. Сервіси SDK для розпізнавання облич можуть бути доступні для різних мов програмування та платформ. Хоча додаток має своє коріння в Китаї, вони також мають клієнтів та партнерів по всьому світу. Їхні технології використовуються в різних країнах для різноманітних цілей.

5. **FaceX.** Це індійська компанія, заснована у 2018 році, яка пропонує як SaaS версії, так і версії для самостійного розміщення. Безкоштовний тарифний план відсутній; єдина цінова опція – щомісячна підписка вартістю 15 доларів США, яка включає 300 запитів щодня. Хоча SDK для мов програмування не згадуються, FaceX пропонує низку інших послуг, таких як ідентифікація орієнтирів обличчя, визначення віку та статі.

6. ***Kairos***. Компанія була заснована у 2012 році в Маямі, Флорида. З того часу вона стала відомою у галузі інноваційних рішень у сфері розпізнавання облич. Програма надає можливість скористатися безкоштовною пробною версією тривалістю 14 днів, в межах якої можна здійснити до 10 000 запитів. Після цього доведеться платити за підписку, яка починається від 19 доларів на місяць плюс 20 доларів за кожні 1 000 розпізнавань. Це досить дорого порівняно з конкурентами. Однак сервіс надає набір інструментів, бібліотек, документації та прикладів, які допомагають розробникам створювати програмне забезпечення для конкретної платформи, операційної системи чи середовища, а також інші послуги, такі як ідентифікація орієнтирів обличчя, встановлення вікової групи і статі. Існують рішення, які можна встановити самостійно на власному сервері, а також версії, які працюють за моделлю послуги (SaaS).

7. ***Machine Box***. Щоб отримати ліцензію на використання продуктів Machine Box, вам потрібно звернутися безпосередньо до їхньої команди або офіційного представника. Вони зможуть надати вам інформацію про доступні варіанти ліцензування та процес отримання ліцензії відповідно до ваших потреб та використання. SDK для мови Go є додатковою перевагою, але вона корисна лише для тих, хто використовує Go. Для масштабування потрібна бізнес-ліцензія. Однак, безкоштовна версія є більш базовою порівняно з деякими іншими масштабованими програмами розпізнавання облич, які доступні безкоштовно для комерційного використання.

8. ***Microsoft Azure Cognitive Services Face API***. При обмеженні в 20 запитів на хвилину, безкоштовний план від Microsoft є великодушним, дозволяючи безкоштовно відправляти 30 000 запитів щомісяця. Проте варто мати на увазі, що в такому випадку відсутній SLA. Платні запити (зі SLA) коштують від 1 долара за 1 000 розпізнавань, залежно від щомісячного обсягу. Підтримувані SDK: .Net, Python, Java, Node.js та Go. Microsoft також надає послуги ідентифікації емоцій, розпізнавання орієнтирів, визначення віку та статі, а також SaaS-версії та варіанти для власного хостингу.

9. **Paravision.** Компанія, що спеціалізується на розпізнаванні облич, призначена для корпоративного сектору і називається Paravision. Їхній перелік послуг і тарифів не розголошується, і вони пропонують лише власні рішення, не SaaS. Paravision надає SDK для Python і C++, проте інформація щодо цін за їхні послуги не оприлюднюється, що нехарактерно для схожих компаній.

10. **Trueface.** Інша нова компанія, який спеціалізується на розпізнаванні облич для бізнесу. Ця компанія зосереджується лише на системах розпізнавання облич, які розташовані на їхньому власному сервері, з різноманітними можливостями, додаток пропонує продукти, які допомагають клієнтам ідентифікувати та аналізувати обличчя на зображеннях і відео, а також забезпечують інші біометричні можливості, такі як виявлення емоцій, вікового діапазону тощо. Підтримується Python та C++ SDK. Щодо вартості Trueface, інформація ще недоступна.

Наразі ця сфера надалі швидко розвивається. Використання застарілих сервісів, які вже не підтримуються оновленнями, втрачає свою доцільність, тому що механізм для виявлення облич все більш удосконалюється. Процеси розпізнавання обличчя можуть працювати за наносекунди. Trueface також наголошує на пом'якшенні упередженості, щоб забезпечити справедливі результати для всіх етнічних груп і статей.

1.4 Загальні аспекти систем розпізнавання емоцій

Часто важко визначити емоції людини лише за виразом обличчя. Дослідження показують, що група осіб сприймають різні емоції, глянувши на ідентичну міміку. Додатком розробленим на базі нейронних мереж дана задача виявляється більш проблематичною. Науковці та розробники з усіх країн активно дискутують про рівень точності сучасних програм емоційного аналізу.

1.4.1 Проблематика ідентифікації емоцій на зображеннях та відео

Існують чинники, які додають складності до процесу аналізу емоцій, і вони поділяються на технічні та психологічні групи.

Технічні. Технічні фактори включають обмеження обладнання та програмного забезпечення, які використовуються для аналізу візуальних даних: розпізнавання об'єктів, неочікувана поведінка та інші фактори. Розглянемо основні технічні виклики, що стоять перед системами виявлення емоцій, та способи їх вирішення:

1. Розширення бази матеріалів

Програми розпізнавання емоцій, як і інші методи машинного та глибокого навчання, вимагають значного масиву інформації для навчання. Ці дані повинні включати відеоматеріали з різною кадровою частотою, зняті з різних кутів зору, на різному фоні, з особами різних національностей, статей та етнічних груп.

Проте більшість наборів даних, доступних у відкритому доступі, є неповними. Наприклад, їм може бракувати різноманіття за кольором шкіри чи статтю, або вони можуть бути обмеженими за обсягом.

Є три варіанти вирішення даної проблеми:

а) створення власного датасету: цей метод є найбільш витратним і часомістким, але він дозволяє створити ідеальний датасет, необхідний для вирішення конкретної задачі.

б) об'єднання різних датасетів: це дозволяє перевірити ефективність вашого програмного алгоритму на різних наборах даних.

в) модифікація даних під час виконання завдання: наприклад, деякі дослідники пропонують модифікувати використані фільми, змінюючи їх освітленість, збільшуючи або зменшуючи швидкість, додавання додаткового шуму.

2. Проблеми з візуальними перешкодами та освітленням обличчя

Перешкоди, такі як зміна положення тіла, часто ускладнюють розпізнавання обличчя людей на відеокадрах, особливо коли мова йде про необроблені відеодані.

Зазвичай для подолання цієї проблеми використовується фронталізація, що дозволяє встановити ознаки обличчя та створити опорні точки для тривимірного зображення. Також досить поширеними є випадки змінених умов освітлення та

чіткості за обмежених можливостей. Задля поліпшення точності розпізнавання розробники часто застосовують методи коригування освітлення.[11]

Деякі дослідники рекомендують включати інфрачервоний або близько інфрачервоний прошарок у інформаційний потік. Даний шар мало вразливий до змін у освітленні та дозволяє збирати інформацію про зміни температурні показники тіла, що відображає емоційний стан особи.

3. Аналіз особливостей обличчя

Обличчя обробляються на наявність особливостей зовнішності за допомогою системи аналізу емоцій. Така обробка може викликати труднощі з наступних причин:

а) Різноманітність відображень: Люди можуть відображати широкий спектр емоцій та мімічних проявів, що може ускладнити процес ідентифікації. Наприклад, одна й та ж емоція, така як радість, може виражатися різними способами відповідно від особистості, культурних особливостей та інших факторів.

б) Зміна в умовах освітленості: Зміна в освітленості може вплинути на вигляд обличчя та вираження емоцій. Наприклад, тіні, які створюються внаслідок яскравого світла або недостатнього освітлення, можуть змінювати форму та контраст обличчя, що ускладнює процес ідентифікації рис.

в) Різноманітність поз і кутів зйомки: Люди можуть перебувати у різних позах та різних кутах зйомки на зображеннях або відео. Це здатне викликати зміну форми та пропорцій обличчя, що робить процес ідентифікації більш складним.

г) Зміна волосся та аксесуарів: Зміна зачіски, наявність окулярів, багатофункціональності або інших аксесуарів також може впливати на вигляд обличчя та емоції, що ускладнює ідентифікацію рис.

д) Зображувальна якість: Погана роздільна здатність знімку, така як розмитість, шум, артефакти стиснення, може ускладнити процес виявлення рис обличчя та ідентифікації емоцій. Враховуючи ці фактори, розробники програм

ідентифікації рис обличчя постійно вдосконалюють свої алгоритми та методи для забезпечення точності та ефективності у різних умовах.

4. Виявлення часткових емоційних станів

Більшість програм, які спрямовані на розпізнавання емоцій, часто фокусуються на виявленні виражених піків емоційної активності, ігноруючи менш виразні вирази. Цей підхід може призвести до неточного розпізнавання емоцій у випадках, коли особа приховує свої почуття, наприклад, в культурах з традиціями збереження емоцій.

Додавання більшої кількості ознак у базу даних дозволяє програмі тренуватися на розпізнавання виразів емоцій з більш різноманітних сигналів. Для цього, крім маркування емоцій та зовнішніх ознак, слід включити в набір ознак міміку, яка відображає глибину емоцій.

Впровадження глибокої пікової нейронної мережі також розв'язує цю проблему. Вона аналізує як пікові, так і менш виражені вирази однакових емоцій і мінімізує частоту оновлення кадрів, що дозволяє отримати більш точні результати.

5. Ідентифікація емоційного контексту

Задній план, а також позиція тіла надають стільки ж інформації про реакцію, як і обличчя. Проте видалення контексту є складним завданням, оскільки потребує аналізу всього кадру, а не лише обличчя. Емоції, що виражаються через рухи, також менш вивчені, ніж міміка.

Моделі нейронних мереж, орієнтовані на контекст, розділяють потокове відео на декілька груп: обличчя (одне або декілька) та його оточення (все, що знаходиться поза людиною, такі як рухи, предмети на зображенні). Сама технологія також розділяється на декілька частин: перша - аналізує міміку обличчя, друга - аналізує оточення. Кінцевий результат відображає дві характеристики методу.

6. Різниця на основі расової приналежності

Ми часто стикаємося з труднощами у розпізнаванні емоцій оточуючих, особливо представників різних расових або культурних груп. Найновітніші досягнення в галузі інтелектуальних систем теж можуть мати складності у цій

області. Наприклад, ще у 2015 році система розпізнавання Google Photos не могла правильно ідентифікувати темношкірих людей.

Проблема полягає передусім у недостатній розмаїтості даних та нерівномірному складі науково-дослідницьких груп, що розробляють програмні рішення та технології. Більшість публічних наборів даних мають широкий спектр соціальних аспектів, що стосуються білих представників чоловічої статі але дуже мало про жінок чи людей іншого кольору шкіри. Це сталося через те, що дослідницькі групи, які ведуть такі дослідження, складаються головним чином з чоловіків білого пігменту шкіри, які в більшості випадків використовують власний досвід для перевірки отриманих результатів.

Інша можлива стратегія для розв'язання цього питання - це більш докладна перевірка масивів даних перед тестуванням. Також можна розглядати варіант формування більш різноманітних та представницьких груп дослідників.

Психологічні. Психологічні аспекти в контексті аналізу емоцій включають унікальність сприйняття та вираження емоцій кожною особою, вплив культурних та соціальних чинників на їх прояв, захисні механізми, що можуть приховувати справжні почуття, внутрішні конфлікти, що впливають на емоційний стан, а також індивідуальні особливості та життєвий досвід кожної людини, які впливають на їх емоційну реакцію.

1. Культурні відмінності в вираженні емоцій

Це є однією з ключових проблем, які виникають у дослідженнях виразу обличчя та прояву емоцій. Ця проблема полягає в тому, що способи вираження та сприйняття емоцій можуть суттєво відрізнитися залежно від культурного контексту. Наприклад, те, що вважається прийнятним або навіть нормальним в одній культурі, може сприйматися як неприйнятне або незвичайне в іншій.

Це має прямий вплив на розпізнавання емоцій за виразом обличчя, оскільки алгоритми розпізнавання можуть бути неправильно налаштовані або неоптимізовані для розпізнавання емоцій, які виявляються відповідно до культурних стандартів різних груп.

Наприклад, емоційний вираз, який вважається виразним або характерним для однієї культури, може бути менш помітним або зрозумілим для представників іншої культури. Такі відмінності у вираженні емоцій можуть впливати на точність алгоритмів розпізнавання емоцій і спричиняти помилки в їх роботі.

2. Виявлення емоцій у дітей

Діти в ранньому віці, особливо немовлята, демонструють свої почуття інакше, ніж це роблять дорослі. Вони можуть бути більш усвідомленими по відношенню до своїх емоцій, ніж здатні їх передати голосом, і реагувати різноманітними виразами обличчя на оточуючі події. Крім того, діти не завжди можуть контролювати свої реакції.

Однак встановлення емоційних станів у немовлят становить виклик для дослідників. Не завжди зрозуміло, чи виразні рухи обличчя вказують на їхній емоційний стан. Ще однією складністю є розуміння дорослими емоційного вираження у дітей. Первинні дослідження емоційних реакцій немовлят, в контексті, можуть неправильно інтерпретувати їхні реакції, вважаючи, що дитина виражає злість, коли насправді вона просто дивиться. Такі помилкові співвідношення між виразами обличчя і емоціями ставлять під сумнів точність досліджень емоцій у дітей на ранніх стадіях їхнього розвитку.

3. Недостовірні показники емоцій

Алгоритми можуть недостатньо точно ідентифікувати ключові елементи емоцій у виразі обличчя через різні особливості фізіономії та варіації виразів. Деякі програмні алгоритми можуть недооцінювати або надто підсилювати певні ознаки, що призводить до неточних результатів.

Використання недостатньо репрезентативних даних для навчання моделей може призвести до помилкових результатів. Якщо набір даних не включає в себе достатньої різноманітності етнічних груп, вікових категорій або культурних контекстів, модель може недооцінювати або надмірно виділяти певні емоційні вирази, особливо в контексті різних культурних нюансів.

Крім того, недостатня обробка контексту може призвести до неточного розпізнавання емоцій. Штучні інтелектуальні системи можуть не враховувати важливі фактори, такі як фоновий шум, мовні особливості, тон голосу та інші субтільні сигнали, що впливають на вираження емоцій.

Тому, ретельно стежте за виникненням нових наукових відкриттів у галузі вираження емоцій та систематично вдосконалюйте свої навички прийняття рішень на підставі аналізу отриманих даних.

1.4.2 Противники систем розпізнавання обличчя

Сьогодні, технології ідентифікації осіб вже вкоренилися в різні сфери нашого повсякденного життя, виявляючи широке застосування.

Користуються нею також державні структури, включаючи департаменти поліції. Згідно з даними Джорджтаунського університету, поліцейські бази включають зображення більшості повнолітніх американців. Урядові установи в міжнародному масштабі використовують систему розпізнавання людей для виявлення осіб, які порушують закон, та відстеження їх руху. Наприклад, поліцейські в Гонконгу використовують цю технологію для відстеження учасників протестів.

Щоб протистояти цій технології, люди різних професій включаючи активістів розробляють одяг та інші ресурси, що надають можливість, використовувати для ухилення від розпізнавання облич. Учасники протестів у різних країнах, від Гонконгу до США, використовують маски для збереження анонімності. Крім того, платформа для захищеної комунікації Signal почала роздавати безкоштовні засоби маскування для ухилення від розпізнавання протестувальникам, таких як протести після смерті Джорджа Флойда.[4]

Технологія розпізнавання обличчя використовується програмами для швидкого розпізнавання обличчя чи будь-яких людських фігур. Однак її можна обдурити, використовуючи спеціальний одяг, який ускладнює розпізнавання. Інші

методи заважають правильній ідентифікації, наприклад, шляхом надання фотографій підставних осіб.

На даний момент такі розробки переважно використовуються у наукових або мистецьких проектах. Проте з розповсюдженням технологій розпізнавання облич вони можуть здобути популярність у галузі технологій. Важливо відзначити, що технології розпізнавання облич не є ідеальними, і вже працюють над алгоритмами для їх уникнення.[5]

1.4.3 Особливості розпізнавання емоцій людини

Людські емоції проявляються через вирази брів, очей та рота. Метод Віолі-Джонса(Рис.1.1) застосовується для виділення цих елементів на портретних знімках. Щоб оптимізувати процес розпізнавання та зменшити кількість помилкових спрацьовувань, обличчя ділять на окремі зони: рот зазвичай розташований у нижній частині, а брови та очі — у верхній.[13] Ключовий етап у розпізнаванні емоцій полягає у визначенні ключових точок на цих елементах. Для точної ідентифікації емоцій важливий аналіз кількох таких точок, процес визначення яких виконується наступним чином:

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

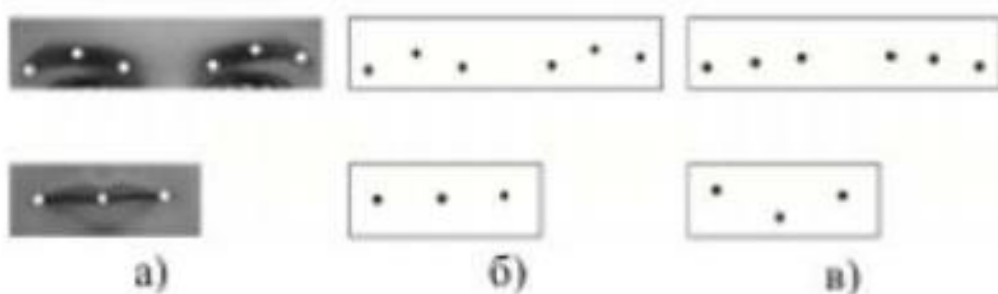


Рис. 1.1 Комбінація точок брів та рота (а – зображення брів та рота; б – комбінація точок брів та рота; в – комбінація ключових точок, яка відповідає іншим положенням розглянутих елементів на зображенні)

Конвертація кольорового зображення в градації сірого відбувається на ранньому етапі визначення обличчя. Для створення бінарного зображення використовують адаптивний поріг. Потім, за допомогою градієнтної маски, на бінарному зображенні формується контурний образ аналізованого елемента. Локалізація ключових точок здійснюється шляхом визначення встановленої кількості точок на контурі цього елемента(Рис.1.2). Тільки після ідентифікації ключових точок проводиться класифікація емоції.

Давайте розглянемо, як поєднання рухів брів та рота виражає різні емоції:

- смуток: коли людина сумує, кінчики її брів та рот опускаються вниз;
- щастя: виявляючи радість, брови та куточки рота піднімаються вгору;
- гнів: характеризується опущенням і зморщуванням брів, а рот розкривається, і його куточки опускаються;
- образа: при відчутті образи брови опускаються, а рот піднімається, з куточками, що опускаються вниз;
- страх: в стані страху брови підіймаються, а рот відкривається і розширюється;
- здивування: при здивуванні брови підіймаються, а рот відкривається.

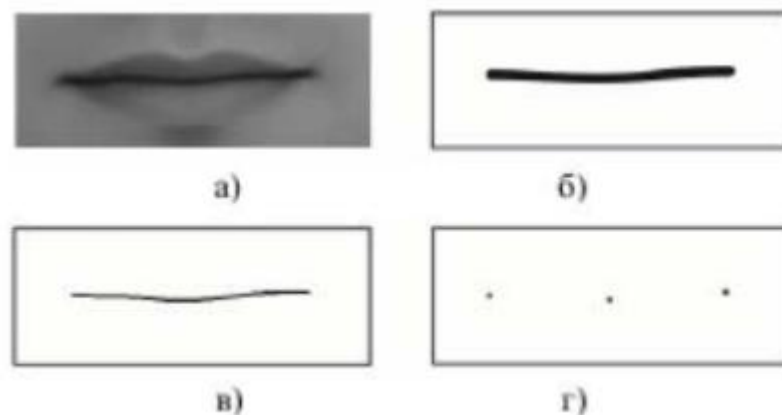


Рис. 1.2 Процес виділення ключових точок (а – напівтонове зображення; б – бінарне зображення; в – застосування градієнтної маски; г – локалізація ключових точок)

1.5 Висновки до розділу

В даному розділі було досліджено еволюцію алгоритмів та методів, які використовувалися та досі застосовуються у системах розпізнавання облич. Починаючи з примітивних підходів, які спробували здійснювати автоматичну ідентифікацію, наступні системи стали все більш точними. Протягом останніх десяти років відбувся значний прорив. За допомогою використання глибокого навчання, системи розпізнавання досягли такого рівня точності, що зрівнюється з можливостями людини у впізнаванні інших осіб. Це стало можливим завдяки різноманітним факторам, таким як розвиток потужних графічних процесорів (GPU) та спеціалізованих процесорів, розширенням доступного обсягу навчальних даних, присутність обширної бази знімків та записів в мережі для створення датасетів, зокрема впровадження передових та продуктивних алгоритмів.

Наприклад, DeepFace використовується для розпізнавання 3D-моделей обличчя. Для розробки та опрацювання цих моделей застосовується аналітична геометрія та традиційні методи комп'ютерного зору.

Проведеним дослідженням визначено, що для розпізнавання емоцій людини одним з найпоширеніших є метод Віоли – Джонса, який забезпечує досить хорошу швидкість і високу точність виявлення заданих об'єктів на зображеннях. Тому розробники й дослідники поєднують різні методи задля підвищення ефективності роботи та використовують сучасні методи для вирішення завдань розпізнавання обличчя.

2 ТЕХНІЧНІ АСПЕКТИ СИСТЕМ ВИЯВЛЕННЯ ЕМОЦІЙНИХ СТАНІВ

2.1 Загальні поняття штучних нейронних мереж

Штучні нейронні мережі - це математичні моделі, які імітують роботу нервової системи живих організмів. Вони складаються з взаємопов'язаних штучних нейронів, які співпрацюють для вирішення різних завдань машинного навчання, таких як класифікація, прогнозування або вирішення задач обробки природної мови. Їх назва та форма навіяні людським мозком, і вони повторюють спосіб спілкування реальних нейронів між собою. Штучні нейронні мережі (ШНМ) — це масивні паралельні системи, що складаються з величезної кількості взаємопов'язаних базових процесорів.

2.1.1 Структура штучної нейронної мережі

Комп'ютерна модель, що використовує структуру та принципи роботи біологічних нейронних мереж, має назву штучна нейронна мережа (ШНМ).

Найпопулярніший вид штучної нейронної мережі складається з трьох наборів одиниць або шарів: шар «вхідних» блоків, з'єднаних із шар «прихованих» одиниць, з'єднаних із шаром «вихідних» одиниць (Рис.2.1). Дані надходять у мережу на входах і проходять мережею шар за шаром, поки не досягнуть виходів.

В архітектурі штучної нейронної мережі є три рівні:

- *Вхідний рівень.* Вхідний рівень нейронної мережі складається з колекція штучних вхідних нейронів. Вони посилають інформацію з початкових нейронних шарів до системи для обробки. Вхідний рівень нейронної мережі ініціює робочий процес.
- *Прихований рівень.* Прихований рівень штучної нейронної мережі складається з вхідного та вихідного рівнів, а вхідні та вихідні дані штучних нейронів зважені за кількістю входів.
- *Вихідний рівень.* Останній шар нейронів є вихідним рівнем у штучній нейронній мережі. мережа, яка забезпечує певні виходи в програматорі. Оскільки

вони є останніми «виконавчими» вузлами мережі, нейрони на вихідному рівні можуть бути створені та оброблятися по-іншому.

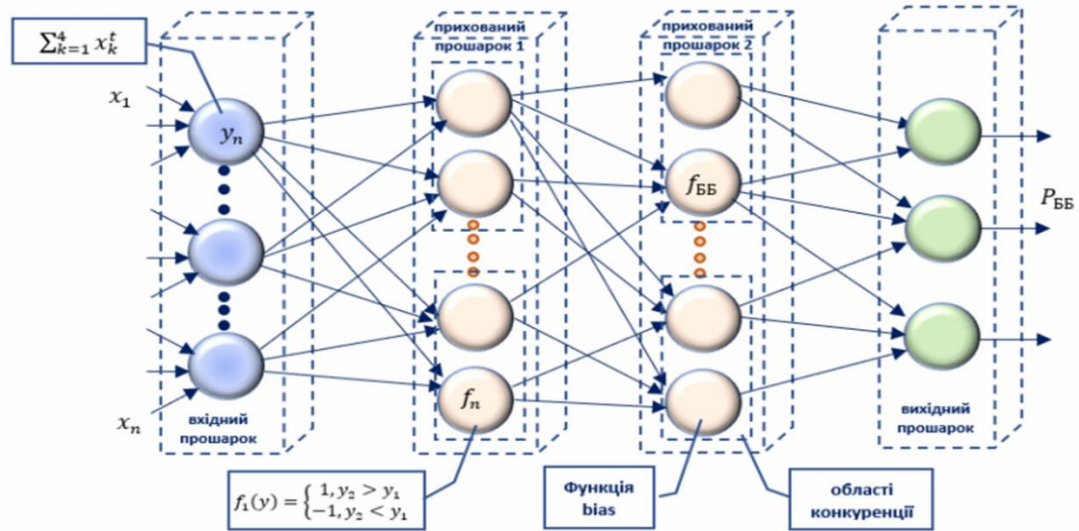


Рис. 2.1 Структура багатошарової штучної нейронної мережі

Нейронні мережі, завдяки своїй надзвичайній здатності до аналізу складних або неоднозначних даних, можуть використовуватися для виділення патернів і виявлення тенденцій, які є настільки складними, що їх важко помітити людям або іншими алгоритмічними методами.

Навчену нейронну мережу можна порівняти з "експертом" у конкретній області, на основі інформації, яку вона аналізувала. Цього "експерта" можна потім використовувати для передбачення нових ситуацій та відповідей на запитання "що буде, якщо...".

Додаткові переваги полягають:

- адаптивне навчання: можливість вивчати завдання на основі наданих для навчання даних або попереднього досвіду;
- самоорганізація: ШНМ може організувати або представляти інформацію, яку вона здібна отримувати під час процесу навчання, самостійно;
- робота в режимі реального часу: обчислення ANN можуть виконуватися паралельно, і створюються та створюються спеціальні апаратні компоненти, які використовують переваги цієї функції;

- відмовостійкість: за допомогою кодування надлишкової інформації, часткова втрата продуктивності спричиняється недоліком мережі.

2.1.2 Навчання штучної нейронної мережі

Машинне навчання — це підсфера штучного інтелекту, яка працює з розпізнаванням образів і моделюванням. Алгоритми машинного навчання обробляють величезні обсяги даних. Навчальні дані, які ви пропонуєте, впливають на поведінку та властивості моделі машинного навчання. Після того, як він навчився, він може передбачити результат, якщо йому нададуть порівняльні дані.[8]

Що стосується навчання, існує дві основні категорії:

- регресія: регресія працює з прогнозуванням числових значень для заданих вхідних даних.
- класифікація: процес класифікації даних у заздалегідь визначені класи (набори) на основі обчислень.

Існує два типи машинного навчання:

а) Контрольоване навчання;

Нам надається набір даних, і ми вже знаємо, як має виглядати наш правильний вихід у контрольованому навчанні, припускаючи, що існує зв'язок між входом і результатом. Проблеми «регресії» та «класифікації» — це два типи контрольованих навчальних завдань. Ми намагаємося передбачити результати в безперервному виході в задачі регресії, що означає, що ми намагаємося зіставити вхідні змінні з деякою безперервною функцією. Натомість ми прагнемо щоб передбачити дискретний результат у завданні класифікації. Іншими словами, ми намагаємося класифікувати вхідні змінні.

б) Неконтрольоване навчання.

Цю структуру можна отримати шляхом кластеризації інформації відповідно до зв'язків між змінними в даних. У неконтрольованому навчанні немає зворотного зв'язку на основі прогнозованих результатів, тому немає вчителя, який би вас виправив.

2.2 Глибинне навчання нейронних мереж

Глибинне навчання є спеціалізованою галуззю машинного навчання, яка використовує складні нейронні мережі з численними шарами, відомими як глибокі нейронні мережі, для аналізу великих масивів даних і виявлення складних патернів. Такі моделі адаптують свої ваги і з'єднання для зменшення помилок у прогнозах і областей, таких як розпізнавання мови, комп'ютерне бачення, рекомендаційні системи і автономні транспортні засоби, де необхідна здатність машин ефективно опанувати складні дані.[9]

Глибинна нейронна мережа (deep neural network, DNN) – це вид штучної нейронної мережі, яка складається з численних шарів обробки інформації. Вона перетворює вхідні дані на вихідні, поступово виділяючи і збираючи характеристики, що дозволяє збільшувати рівень абстракції даних з вхідних до вихідних. На відміну від поверхневих нейронних мереж, ГНМ мають більшу кількість нейронів і зв'язків, що забезпечує вищу обчислювальну потужність та здатність моделювати складні залежності.

Спеціалізація шарів і висока ієрархічність обробки даних роблять ГНМ ефективними для обробки та інтерпретації. Крім того, спеціалізовані шари ГНМ дозволяють інтегрувати апріорні знання про предметну область у мережеву модель.

Проте, конкретні архітектури глибоких нейронних мереж зазвичай мають вузьке застосування у специфічних задачах. Скажімо, деякі архітектури ефективні переважно для розпізнавання зображень і можуть не підходити для інших видів завдань.

2.3 Згорткові штучні нейронні мережі: загальні аспекти

Згорткова нейронна мережа (CNN) — це вид нейромережі глибокого навчання, яка добре підходить для аналізу зображень і відео.

CNN містить численні рівні, включаючи повністю зв'язаний рівень, рівень нелінійності, рівень об'єднання та згортковий рівень. Програми та результати буфера пам'яті були чудовими, включаючи найбільший набір даних класифікації

зображень (Illustration Net), комп'ютерне бачення та обробку природної мови (NLP). Десятки чи сотні шарів CNN можуть навчитися розпізнавати різноманітні характеристики зображення. Вихідні дані кожного згорнутого зображення використовуються як вхідні дані для наступного шару, а кожне навчання обробляється за допомогою фільтрів різної роздільної здатності. Фільтри можуть бути змінені, від простих критеріїв, таких як яскравість і краї, до атрибутів, які характеризують виключно компонент.

2.3.1 Архітектура згорткових нейронної мережі

2.3.1.1 Базова архітектура CNN

Базова архітектура згорткової нейронної мережі (CNN) ефективно обробляє візуальні дані за допомогою послідовності конволюційних, пулінгових і повнозв'язних шарів. Конволюційні шари автоматично виявляють ключові ознаки зображень, пулінг зменшує розміри даних, зберігаючи при цьому важливі деталі, а повнозв'язні шари класифікують дані, використовуючи нелінійність функцій активації для розпізнавання складних шаблонів. Це забезпечує гнучкість і точність у задачах комп'ютерного зору.

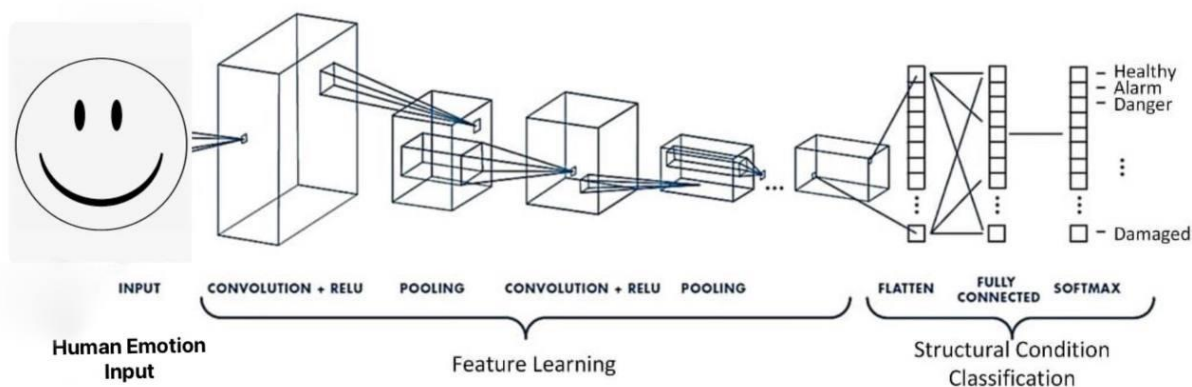


Рис. 2.2 Загальна архітектура згорткової нейронної мережі

Процес роботи зі згортковими нейронними мережами включає наступні етапи:

- вхідні дані, які можуть бути зображеннями або відео, подаються на вхідний рівень;

- згорткові шари виявляють різноманітні ознаки у вхідних даних, використовуючи фільтри для розпізнавання різних ознак у вхідних даних, виявляючи контури, форми, текстури та інші особливості;
- після кожного згорткового шару використовується функція активації ReLU, яка надає нелінійність та поліпшує продуктивність мережі;
- шар пулінгу зменшує розмірність вихідних даних після згорткових шарів, зберігаючи важливу інформацію, зменшує кількість параметрів і обчислень у мережі, зменшує ризик перенавчання і покращує загальну ефективність, виділяючи найважливіші ознаки;
- повнозв'язні шари об'єднують всі нейрони з попереднього шару з кожним нейроном наступного шару, формуючи єдину велику матрицю з'єднань. Вони використовуються для остаточної класифікації або регресії, перетворюючи виділені ознаки на вихідні значення.

2.3.2 Архітектура згорткової нейронної мережі LeNet

Для реалізації проєкту була створена згорткова нейронна мережа яка базується на структурі моделі LeNet з деякими модифікаціями.

Архітектура LeNet 5 — це «Hello World» у сфері згорткових нейронних мереж. Правило зворотного поширення вперше було застосовано до всіх розумних застосувань у 1989 році Янном Лекуном та його колегами з Bell Labs. Вони також стверджували, що шляхом накладення обмежень із області завдань гнучкість узагальнення мережі може бути значно посилена. ЛеКун встановив, що одношарові мережі справді мають тенденцію демонструвати слабкі навички узагальнення, пояснюючи скромну аномалію ідентифікації рукописних цифр в іншій публікації того ж року, навіть припускаючи, що проблема є лінійно роздільною. Багатошарова, неприродна мережа може функціонувати надзвичайно добре, коли аномалію буде усунено за допомогою детекторів незмінних ознак. Він вважав, що ці відкриття доводять, що зменшення кількості вільних параметрів у нейронній мережі може покращити її здатність до узагальнення.

LeNet-5 символізує появу CNN і окреслює його ключові компоненти. Однак у той час це не було популярним через брак апаратного забезпечення, особливо GPU (блок графічного процесора, цільовий електронний компонент розроблений для модифікації пам'яті з метою прискорення створення зображень під час буферизації для виведення на екран демонстрації) та альтернативних алгоритмів, такі як SVM, які могли б виконувати ефекти, подібні до ефектів LeNet або навіть кращі за них.

Особливості LeNet-5:

- кожен згортковий рівень включає три частини: згортку, об'єднання та нелінійній функції активації;
- використання згортки для виділення просторових характеристик (спочатку) згортка називалася рецептивними полями);
- середній шар об'єднання використовується для підвибірки;
- « \tanh » використовується як функція активації;
- використання багаторівневого перцептрона або повністю підключених шарів як останнього класифікатора;
- рідкий зв'язок між шарами зменшує складність обчислень.

Архітектура LeNet-5 CNN має сім рівнів. Три шари згортки, два шари підвибірки та два пов'язані шари складають композицію шару(Рис. 2.3).

1. Перший шар

Зображення розміром 32x32 у відтінках сірого служить вхідним для LeNet-5 і обробляється першим згортковим шаром, який включає шість карт або фільтрів. Від 32x32x1 до 28x28x6 розміри зображення змінюються.

2. Другий шар

Потім, використовуючи розмір фільтра 22 і крок 2, LeNet-5 додає середній шар об'єднання або шар підвибірки. 14x14x6 буде зменшеним розміром остаточного зображення.

3. Третій шар

Потім присутній другий згортковий шар із 16 картами функцій розміром 55 і кроком 1. Лише 10 із 16 карт об'єктів у цьому шарі пов'язані з шістьма картами об'єктів у шарі нижче.

4. Четвертий шар

З розміром фільтра 22 і кроком 2 четвертий шар (S4) знову є середнім шаром об'єднання. Результат буде зменшено до $5 \times 5 \times 16$, оскільки цей шар ідентичний другому шару (S2), але має 16 карт функцій.

5. П'ятий шар

П'ятий шар (C5) містить 120 карт об'єктів, кожна розміром 1×1 , і є повністю зв'язаним згортковим шаром. Усі 400 вузлів ($5 \times 5 \times 16$) четвертого рівня, S4, підключені до кожного з 120 одиниць у 120 одиницях C5.

6. Шостий шар

Повністю пов'язаний шар (F6) із 84 одиницями складає шостий шар.

7. Вихідний рівень

Рівень виводу SoftMax, який має 10 потенційних значень і відповідає цифрам від 0 до 9, є останнім шаром.

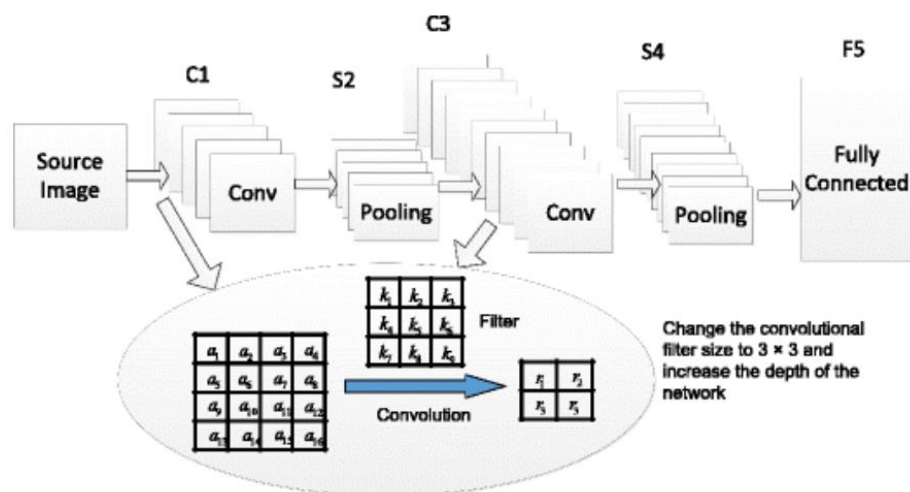


Рис. 2.3 Архітектура LeNet-5

Таким чином, LeNet-5 використовується в основному для базових задач обробки зображень і розпізнавання символів. Вона залишається важливою моделлю для розуміння основ згорткових нейронних мереж і їх застосувань.

2.4 Висновки до розділу

В даному розділі була описана нейронна мережа, на основі якої була побудована модель. Було розглянуто основні компоненти та методи, що використовуються для створення ефективних систем розпізнавання емоцій за виразом обличчя. Зокрема, було проаналізовано використання згорткових нейронних мереж та інших алгоритмів глибокого навчання, які забезпечують високу точність та швидкість розпізнавання.

В ході розробки системи автоматизованого контент-аналізу емоцій з використанням штучного інтелекту, архітектура LeNet-5 використана для аналізу вхідних даних, які можуть включати тексти, зображення або відео, з метою виявлення емоційних ознак.

Архітектура LeNet-5 є досить відомою згортковою нейронною мережею, спочатку розробленою для розпізнавання рукописних цифр. Вона включає у себе шари згортки, пулінгу та повнозв'язані шари, які допомагають виявляти та класифікувати ознаки у зображеннях.

На основі проведеного аналізу можна зробити висновок, що для підвищення ефективності систем виявлення емоційних станів необхідно збільшувати обсяг та різноманітність навчальних даних, а також вдосконалювати алгоритми розпізнавання для більш точного врахування нюансів людських емоцій. Майбутні наукові дослідження та інновації в цій області обіцяють значні перспективи з метою покращення якості систем розпізнавання емоцій та їх застосування в різних сферах.[17]

3 РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ

3.1 Вибір мови програмування для реалізації системи розпізнавання емоцій

Для створення системи була обрана мова програмування Python, оскільки вона має багато переваг.

Python — це мова програмування, яку легко вивчити. Вона вважається високорівневою мовою програмування загального призначення, відома своєю простотою та читабельністю коду. Підтримує різні парадигми програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне програмування. Завдяки великій кількості бібліотек і фреймворків, Python широко використовується в наукових дослідженнях, розробці веб-додатків, аналізі даних, машинному навчанні та багатьох інших областях.[16]

Переваги використання мови Python:

- легко вивчати та використовувати: Python має просту та доступну структуру, граматику легко зрозуміти початківцям, а читання коду дозволяє швидко зрозуміти та писати програми;
- величезна бібліотека стандартних функцій: Python має величезну бібліотеку стандартних функцій, бібліотека, що містить модулі для обробки таких завдань, як обробка тексту, обробка документів, взаємодія з Інтернетом, обробка даних тощо;
- великий вибір сторонніх бібліотек і фреймворків: на вибір є велика кількість сторонніх бібліотек і фреймворків, бібліотеки функцій і фреймворки для різних областей застосування, таких як NumPy і SciPy для наукових обчислень, Django і Flask для веб-розробки, TensorFlow і PyTorch для машинного навчання;
- портативність і кросплатформенність: Python може працювати на різних платформах, такі платформи, як Windows, macOS, Linux тощо, дозволяють розробникам писати програми, які можуть працювати на цих платформах без зміни програмного коду;
- режим інтерактивного програмування: Python підтримує інтерактивне

програмування, (REPL) режим дозволяє розробникам тестувати та змінювати код на льоту, що значно прискорює процес розробки та виправлення помилок;

- підтримка різних концепцій програмування: це забезпечує гнучкість у різних підходах до вирішення проблем;
- активна та велика спільнота: існує багато ресурсів, форумів, документації та навчальних матеріалів, це допомагає розв'язувати проблеми та вдосконалювати свої навички програмування;
- застосування в різних галузях: Python використовується в різних галузях, таких як, веб-розробка, наукові дослідження, обробка даних, автоматизація систем, машинне навчання, штучний інтелект та багато інших;
- легка інтеграція з іншими мовами та технологіями: Python інтегрований з іншими мовами програмування, такими як C, C++ і Java, тому його можна використовувати в складних проектах;
- розвиток високої продуктивності: Python надає різноманітні бібліотеки, які дозволяють розробникам писати код швидше, скорочуючи час розробки та звільняючи ресурси для інших завдань.

3.2 Аналіз пакетів обраної мови програмування Python

Індексатор пакетів Python, також відомий як PyPI, є офіційним репозиторієм програмного забезпечення Python. Це дозволяє розробникам легко знаходити, встановлювати та ділитися пакетами Python. PyPI є невід'ємною частиною екосистеми Python і дозволяє встановлювати та розгортати бібліотеки та інші програми.

PyPI є провідною частиною екосистеми Python, що значно полегшує створення, розгортання та розгортання програм. Його інтеграція з pip робить керування пакетами простим і ефективним процесом, що дозволяє розробникам зосередитися на написанні коду замість вирішення проблем із сумісністю.

Pip — це стандартний менеджер пакетів для Python. Він дозволяє встановлювати та керувати пакетами, які не входять до вбудованої бібліотеки

Python. Pip є важливим інструментом для інсталяції пакетів Python, розпочинаючи з Python 3.4 та 2.7.9. Це обов'язковий інструмент для будь-якого користувача Python, тому що його застосовують в багатьох проектах Python. Для людей, які займаються програмуванням на інших мовах, концепція менеджера пакетів буде знайома: Ruby застосовує gem, JavaScript використовує npm, а .NET використовує NuGet. Зараз Pip є менеджером пакетів Python.

Python часто називають мовою «вбудованих акумуляторів», оскільки його стандартна бібліотека містить великий вибір пакетів і модулів, які допомагають розробникам у їхніх проектах. Крім того, активна спільнота Python збагачує бібліотеку численними пакетами через PyPI, індекс пакетів Python. PyPI містить багато пакетів, включаючи фреймворки розробки, інструменти та бібліотеки, які забезпечують зручні інтерфейси для розширення функціональності стандартної бібліотеки Python.

3.3 Сутність програмних засобів, використаних при реалізації проєкту

3.3.1 OpenCV

OpenCV - це відкрита бібліотека, яка створена для обробки зображень, комп'ютерного зору та чисельних алгоритмів. OpenCV широко застосовується в продуктах, зокрема в системі OCR Solutions, що дозволяє розпізнавати техпаспорти, закордонні паспорти та інші документи.

Взагалі, OpenCV є важливою складовою нашого технологічного стеку для розробки систем штучного інтелекту та комп'ютерного зору. Ця бібліотека реалізована мовами програмування C/C++ і має підтримку для великого переліку мов програмування.

Можливості OpenCV:

- кадрування;

Одна з ключових функціональних можливостей, якою ми часто користуємось, це зрозумілий та простий API OpenCV. Якщо нам потрібно вирізати певну частину

з вихідного зображення, ми просто вказуємо координати, обрізаємо зображення і продовжуємо нашу роботу.

- модифікація розміру;

Бібліотека дозволяє виконувати цю операцію з великою гнучкістю та точністю. Вона включає методи розмірної зміни, інтерполяції та обрізання зображень. Крім того, OpenCV дозволяє змінювати розмір кожного каналу окремо для кольорових зображень та працювати з різними типами зображень, включаючи високоякісні формати.

- реалізація обертання зображення;

Можливість змінювати орієнтацію початкового зображення у будь-якому напрямку, віддзеркалювати його та інші подібні операції.

- конвертування кольорового зображення у відтінки сірого або чорно-білого;

Існує кілька стандартних методів, які дозволяють виділити ключові деталі зображення, дотримуючись точних налаштувань для порогу відсіву. Цей підхід у нашій роботі використовується досить часто.

- розмиття/ згладжування;

Зрозуміло, що ця операція слугує для створення розмиття на початковій відзнятій картинці. Проте, вона не призначена для відновлення різкості у вже розмитих зображень, для цього використовуються інші методи.

- малювання прямокутників і ліній;

Ми використовуємо прямокутники для визначення областей об'єктів під час маркування набору даних для систем розпізнавання. Ми додаємо мітки, щоб ілюструвати, які об'єкти присутні на зображенні, їх розташування та кількість. Також можемо намалювати різноманітні геометричні фігури, а не лише прямокутники, змінювати їх кольори, товщину ліній та зафарбовування.

- написи (текст на зображенні);

Дана функція використовується під час роботи з відеоматеріалами, коли необхідно динамічно відображати певні параметри або те, що відбувається в даний момент на екрані. OpenCV дозволяє виявляти текст на зображеннях за допомогою

методів виявлення контурів, фільтрації зображень, машинного навчання та аналізу текстур і кольорових функцій.

- розпізнавання обличчя.

OpenCV має вбудовані інструменти для розпізнавання обличчя, які дозволяють виявляти, вирізати та аналізувати обличчя на зображеннях та відео. Також забезпечує можливість виявлення обличчя на зображеннях та відео, включаючи функції вирізання, визначення ключових точок обличчя та аналізу емоцій та виразів, використовуючи алгоритми машинного навчання та комп'ютерного зору для точного вирізання та аналізу обличчя.[14]

3.3.2 NumPy

NumPy - це популярна бібліотека мови Python, що використовується для операцій з масивами. Крім цього, вона містить інструменти для розв'язання завдань у лінійній алгебрі і роботи з матрицями.

NumPy був створений Тревісом Оліфантом у 2005 році і є проектом з відкритим вихідним кодом, що дозволяє його вільне використання.

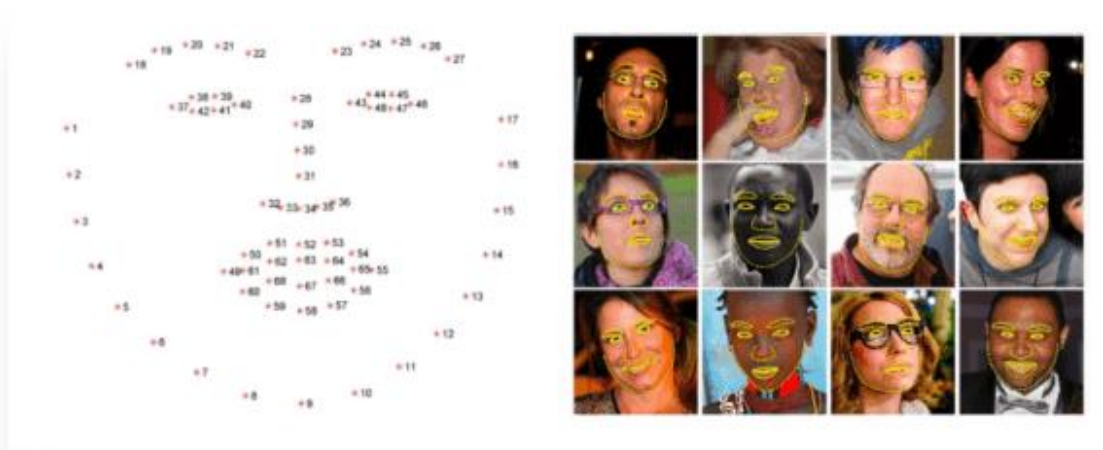


Рис. 3.1 Система ідентифікує обличчя

У Python є списки, що використовуються для роботи зі множинами даних, але їх обробка може бути повільною. NumPy пропонує альтернативу у вигляді об'єкту масиву, який функціонує набагато швидше, чим звичайні списки Python. Цей об'єкт масиву у NumPy відомий як ndarray і має багато додаткових функцій, які роблять роботу з ним дуже зручною.

Переваги використання даної бібліотеки:

- швидкість: операції з масивами у NumPy виконуються швидше, ніж звичайні списки у Python, що робить її ідеальним інструментом для обробки великих обсягів даних;
- ефективність використання пам'яті: NumPy оптимізує використання пам'яті, що дозволяє ефективно працювати з великими обсягами даних, зменшуючи накладні витрати;
- широкий функціонал: бібліотека має велику кількість функцій для роботи з масивами, включаючи математичні операції, лінійну алгебру, обробку зображень, обробку сигналів та багато іншого;
- удосконалені операції індексації та зрізів: NumPy надає потужні засоби для індексації та вибірки даних з масивів, що спрощує роботу з ними;
- широке співпрацювання з іншими бібліотеками: NumPy легко інтегрується з іншими популярними бібліотеками, такими як SciPy, Pandas, Matplotlib тощо, що розширює його можливості;
- підтримка векторизованих операцій: масивні операції можуть бути векторизовані, що дозволяє виконувати їх над цілими масивами даних без необхідності використання циклів;
- висока надійність і стабільність: NumPy має широку спільноту користувачів і активно підтримується, що забезпечує стабільну роботу та швидке виправлення можливих помилок.

3.3.3 Matplotlib

Matplotlib — це потужна бібліотека графічних зображень у Python, яка використовується для створення статичних, анімованих та інтерактивних візуалізацій. Основна мета Matplotlib — надати користувачам інструменти та функціональність для графічного представлення даних, що полегшує їх аналіз і розуміння. Він був спочатку розроблений Джоном Д. Хантером у 2003 році і зараз підтримується великою спільнотою розробників.

Основні характеристики Matplotlib:

- універсальність : Matplotlib може створювати широкий спектр графіків, включаючи лінійні діаграми, точкові діаграми, стовпчасті діаграми, гістограми, секторні діаграми тощо.
- налаштування : пропонує широкі можливості налаштування для керування кожним аспектом графіка, таким як стилі ліній, кольори, маркери, мітки та анотації.
- інтеграція з NumPy : Matplotlib легко інтегрується з NumPy, що полегшує безпосередню побудову масивів даних.
- якість публікації : Matplotlib створює високоякісні сюжети, придатні для публікації, з детальним контролем над естетикою сюжету.
- розширюваність : Matplotlib дуже розширюваний, із великою екосистемою додаткових наборів інструментів і розширень, таких як Seaborn, функції побудови графіків Pandas і Basemap для географічного побудови.
- кросплатформенність : можливість роботи на різних операційних системах, таких як Windows, macOS і Linux, незалежно від платформи.
- інтерактивні графіки : Matplotlib підтримує інтерактивне малювання за допомогою віджетів і обробки подій, що дозволяє користувачам динамічно досліджувати дані.

Matplotlib користується популярністю завдяки своїй простоті у використанні, обширній документації та широкому спектру можливостей побудови. Він пропонує гнучкість у налаштуваннях та підтримує різні типи графіків.

Matplotlib є підходящим вибором для різноманітних завдань візуалізації даних, у тому числі дослідницького аналізу даних, наукового побудови графіків і створення графіків якості публікації. Він чудово підходить у сценаріях, коли користувачам потрібен детальний контроль над налаштуванням сюжету та створення складних або спеціалізованих візуалізацій.

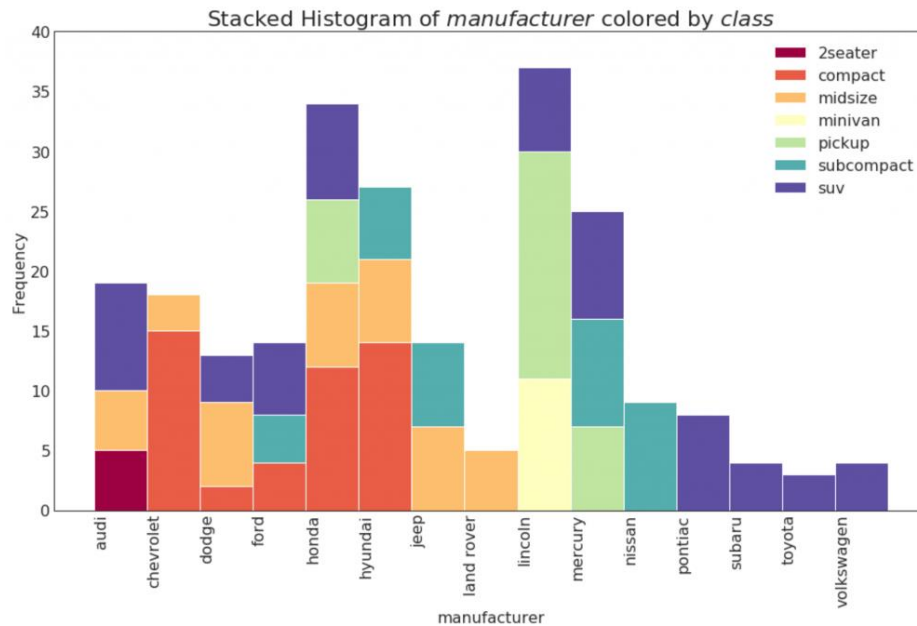


Рис. 3.2 Приклад гістограми побудованої за допомоги Matplotlib

3.3.4 SciPy

SciPy на Python - це високорівнева бібліотека, яка надає функції для числових обчислень та наукових областей, таких як оптимізація, апроксимація, обробка сигналів, статистика, розв'язання диференціальних рівнянь та інше. Вона розширює базовий функціонал NumPy, надаючи більш складні алгоритми та методи для вирішення.

Можливості SciPy:

- SciPy містить функції для числової оптимізації, включаючи методи мінімізації та пошуку коренів функцій.
- бібліотека має інструменти для апроксимації функцій та даних за допомогою різних методів, таких як найменших квадратів.
- SciPy надає функції для обробки сигналів, включаючи фільтрацію,

згладжування, аналіз спектра та інше.

- різноманітні статистичні функції для обчислення різних параметрів розподілів, тестів гіпотез, аналізу дисперсії та інших операцій.
- SciPy надає методи для чисельного розв'язання різних типів диференціальних рівнянь, включаючи звичайні та часткові.
- інтегрування: бібліотека має функції для чисельного обчислення інтегралів, включаючи одновимірні, багатовимірні та квадратури по замкнених кривих.
- SciPy має інструменти для роботи з розрідженими матрицями та розв'язання систем лінійних рівнянь з розрідженими матрицями.
- обробка зображень: бібліотека містить функції для обробки та аналізу зображень, включаючи фільтрацію, сегментацію, морфологічні операції та інші.

SciPy - це відкрита бібліотека на Python, що використовується для наукових обчислень у математиці, інженерії та інших технічних областях. Вона містить різноманітні модулі, які допомагають вирішувати широкий спектр завдань у цих галузях. SciPy інтегровано з NumPy, що спрощує роботу з числовими обчисленнями.[16]

3.3.5. TensorFlow

TensorFlow - це відкрита платформа машинного навчання, розроблена компанією Google. Вона надає інструменти для створення, навчання та виконання нейронних мереж та інших моделей машинного навчання. Спочатку розроблялась для великих числових обчислень, але виявилась надзвичайно корисною і для глибокого навчання.

Розроблена командою Google Brain, TensorFlow була вперше представлена громадськості у 2015 році. Вона об'єднує численні моделі та алгоритми машинного і глибокого навчання (також відомих як нейронні мережі) і надає їх у зручній формі через програмні метафори. TensorFlow використовує Python або JavaScript для зручного API, а виконання здійснюється на високопродуктивній мові C++.

TensorFlow також пропонує велику бібліотеку попередньо навчених моделей для використання у власних проектах.

TensorFlow дає змогу користувачам генерувати графи обчислень, які відображають, як дані проходять через послідовність вузлів. Кожен вузол у графі являє собою математичну операцію, а зв'язок між вузлами - багатовимірний масив даних або тензор.

Ця архітектура дозволяє легке виконання коду TensorFlow на кластері комп'ютерів, зокрема при використанні графічних процесорів, може бути розподілено.

Основні етапи роботи з TensorFlow включають в себе створення обчислювальних графів, визначення функцій втрат та оптимізаційних алгоритмів, навчання моделі на навчальних даних, оцінку її результатів та застосування моделі для передбачень на нових даних.

3.3.6 Keras

Keras — зазвичай працює поверх бібліотеки низькорівневого машинного навчання, такої як TensorFlow або Theano. Він надає доступний, високопродуктивний інтерфейс для вирішення проблем машинного навчання (ML) з акцентом на сучасне глибинне навчання.

Keras - це комплексний інструмент, який включає в себе всі етапи процесу машинного навчання, починаючи з підготовки даних і завершуючи налаштуванням параметрів моделей та їх розгортанням. Основними структурами даних Keras є шари та моделі. Шар — це просте перетворення вводу/виводу, а модель — орієнтований ациклічний граф (DAG) шарів.

Keras відкриває широкі можливості масштабування та кросплатформенного використання завдяки інтеграції з TensorFlow. Ви можете розгорнути Keras на потужних TPU Pod або великих кластерах GPU, експортувати моделі Keras для запуску у браузері або на мобільних пристроях не викличуть ніяких труднощів. Ви також можете обслуговувати моделі Keras через веб-API.[10]

Keras призначений для зменшення когнітивного навантаження шляхом досягнення наступних цілей:

- пропонуйте прості, узгоджені інтерфейси;
- мінімізуйте кількість дій, необхідних для стандартних випадків використання;
- надайте зрозумілі, дієві повідомлення про помилки;
- дотримуйтеся принципу поступового розкриття складності: легко почати, і ви можете завершити розширені робочі процеси, навчаючись по ходу;
- допомогти вам написати стислий, читабельний код.

Keras виконує роль програмного інтерфейсу, який спрощує дії.

3.3.7 Pandas

Бібліотека Pandas - це відкрита бібліотека на мові програмування Python, яка надає зручні та потужні інструменти для обробки та аналізу даних. Вона дозволяє ефективно працювати з табличними даними, виконувати операції фільтрації, сортування, групування, об'єднання та багато іншого. Pandas дозволяє зручно завантажувати, зберігати та маніпулювати даними з різноманітних джерел, таких як CSV файли, бази даних SQL, Excel таблиці та інші.

Бібліотека Pandas надає широкий набір можливостей для роботи з даними у форматі табличних структур. Зокрема, вона дозволяє зчитувати та записувати дані з різних джерел, виконувати операції фільтрації, сортування, групування та об'єднання даних. Pandas також надає зручний доступ до окремих елементів та рядків даних, а також можливість виконувати операції зі змінними усіх типів, включаючи числа, рядки, дати та час. Вона підтримує роботу з багатовимірними даними та має вбудовані функції для статистичного аналізу, візуалізації даних та роботи з відсутніми даними.

Pandas також має інструменти для роботи з часовими рядами, агрегації даних та роботи з великими обсягами інформації. Вона стала однією з найпопулярніших бібліотек для аналізу даних у середовищі Python завдяки своїй простоті використання та потужним можливостям.

Бібліотека відзначається високою продуктивністю завдяки оптимізації критичних частин коду, які реалізовані у мові Python та C.

3.4 Програмна реалізація системи розпізнавання емоцій

Для реалізації проєкту була обрана мова програмування Python. Для взаємодії зі згортковими нейронними мережами були обрані наступні бібліотеки: Pandas; OpenCV; Numpy; Pillow; Matplotlib; H5py; Keras; Tensorflow; Scipy.

Заради того, щоб система була здатна відтворювати емоції, вона потребує бази даних. В даний час, знайти таку базу не є складною задачею, оскільки ця галузь досить стабільно прогресує. Я скористалася FER-2013, яка можливо є однією з найбільших баз даних у цьому напрямку. У цій базі даних міститься приблизно 36 тисяч сірих зображень, які представляють сім основних емоцій.

За допомогою функції `read_csv` з пакету бібліотеки `pandas` було завантажено дані з бібліотеки FER-2013 у форматі `*.csv`. За допомогою функцій з бібліотеки `OpenCV`, алгоритм може отримувати та обробляти відеодані з камери. За допомогою класифікатора `Haar Cascade`, що також входить до бібліотеки `OpenCV`, програма розпізнає обличчя людей. Після цього проводиться обробка даних та виведення результату у формі кадру (`Frame`) на екран. Функція `draw_text` з бібліотеки `Pillow` відповідає за відображення тексту на екрані. Обробка даних проводиться за допомогою згорткової нейронної мережі, яка була легко реалізована завдяки використанню `Keras API`, яке базується на `TensorFlow`.

3.5 Тестування створеного програмного забезпечення

Тепер настав час для проведення тестування створеного проєкту. Для початку, ми маємо виконати запуск програми, це здійснюється через командний рядок. Додаток аналізує емоції в реальному часі. Після запуску відкриється вікно, в якому ви клацаєте по зображенню на камері комп'ютера. Система розпізнає вираз обличчя та емоційний стан людини, наприкінці відображає результати на екрані. Відображення результатів змінюється відповідно до зміни зовнішнього вигляду обличчя. Давайте подивимося, як система розпізнає різні емоції:

- а) злість(Рис.3.3);
- б) сум(Рис.3.4);
- в) щастя(Рис.3.5);
- г) спокій (нейтральний вираз)(Рис.3.6);
- д) здивування(Рис.3.7).

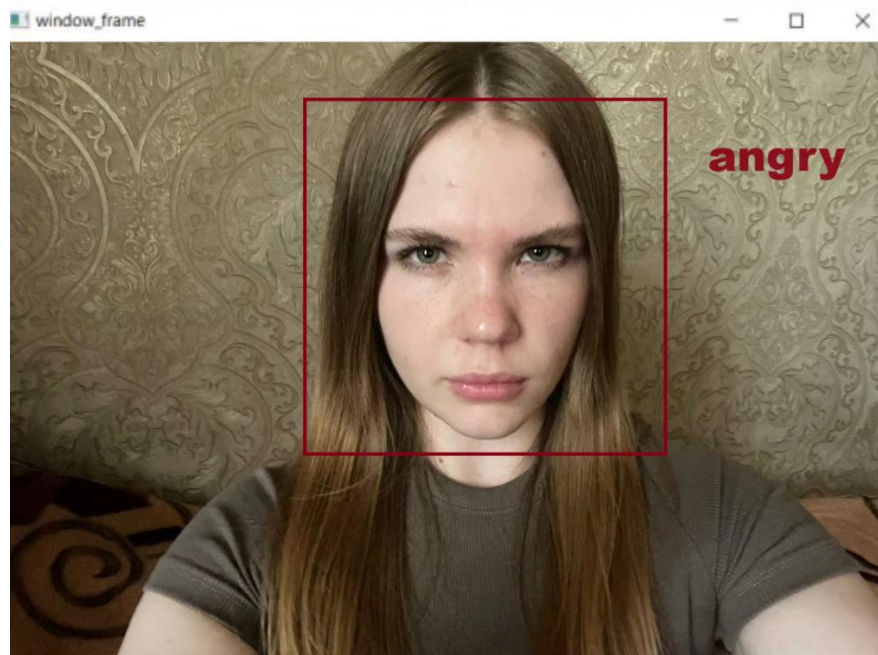


Рис.3.3 Результат емоції «Злість»

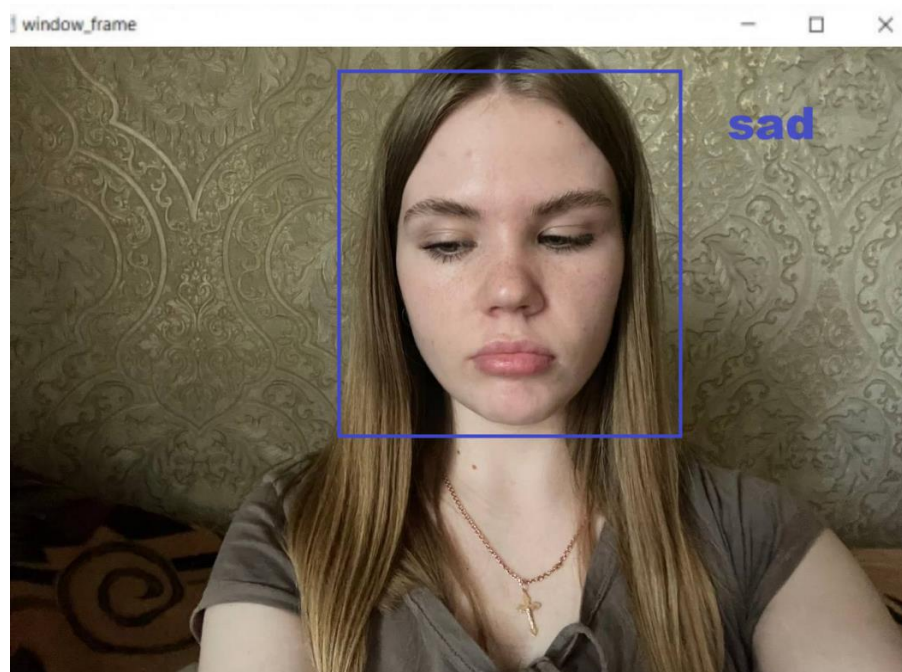


Рис. 3.4 Результат емоції «Сум»

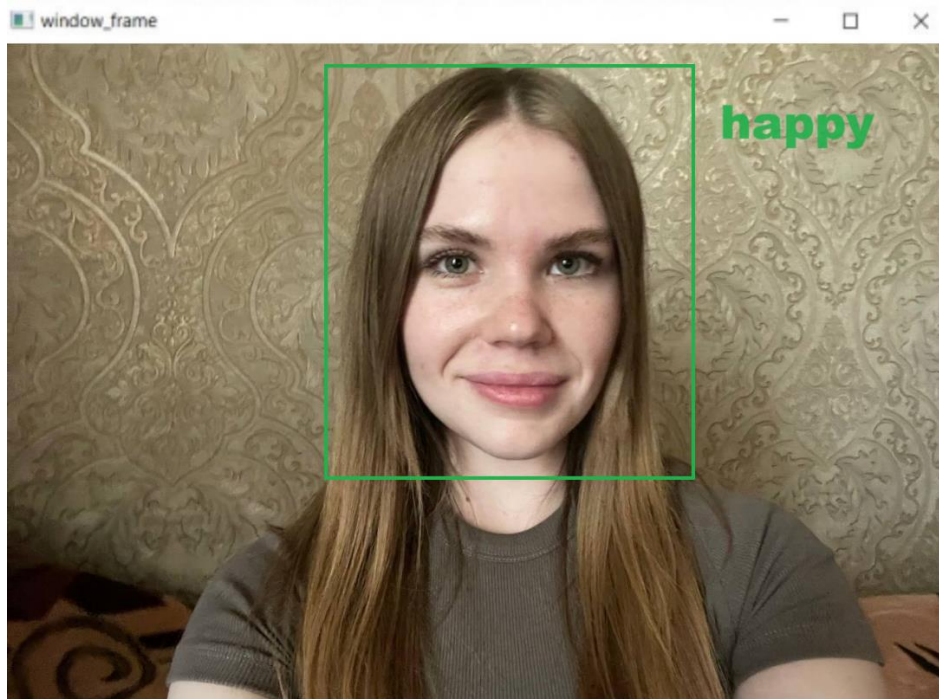


Рис.3.5 Результат емоції «Щастя»

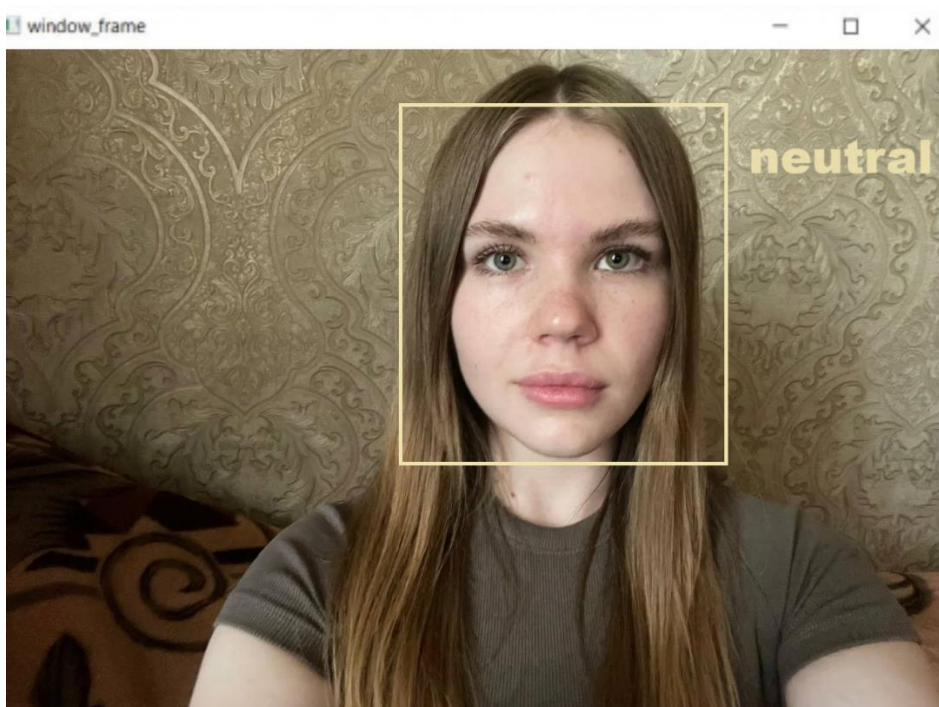


Рис. 3.6 Результат емоції «Спокій»

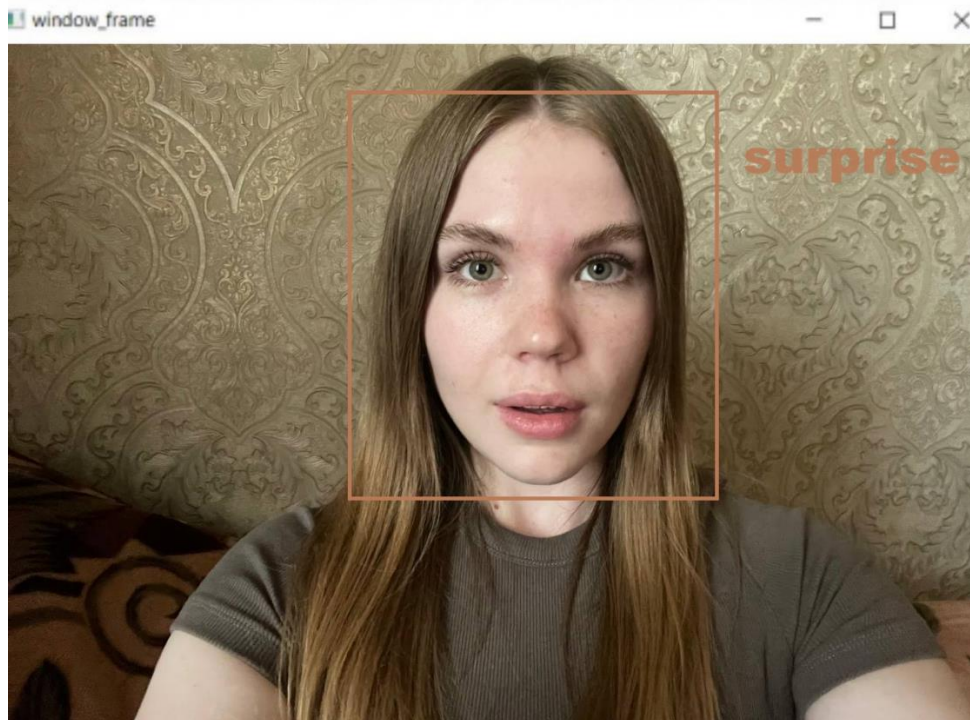


Рис. 3.7 Результат емоції «Здивування»

Метрики precision, recall та f-score використовувалися для оцінки ефективності методу. Давайте розглянемо ці метрики більш детально.

1. Accuracy (точність) - це відповідність результатів вимірювань до еталонного значення або істинної величини.
2. Precision (влучність) - це відношення кількості правильно визначених елементів до загальної кількості визначених елементів, що дозволяє оцінити точність виявлення певного класу.
3. Recall (повнота) - це відношення кількості правильно визначених елементів до загальної кількості дійсних елементів певного класу, що вказує на здатність системи виявляти всі випадки певного класу.
4. F-score (F-міра) - це гармонічне середнє між точністю і повнотою, яке дозволяє оцінити збалансованість між цими двома метриками в одному числі. Для визначення ефективності методу застосовуються метрики влучності та повноти. Влучність відображає відношення правильно визначених позитивних випадків до загальної кількості визначених позитивних випадків, включаючи помилкові визначення. Повнота, у свою чергу, вимірюється як відношення правильно

визначених позитивних випадків до загальної кількості позитивних випадків, які могли б бути визначені.

Таблиця 3.1

Тестування та оцінка ефективності розпізнавання

<i>№</i>	<i>Емоція</i>	<i>Кількість фото для тестування</i>	<i>Оцінка ефективності розпізнавання</i>
1.	Злість	10	67%
2.	Страх	10	52%
3.	Щастя	10	67%
4.	Спокій(Нейтральний вираз обличчя)	10	65,7%
5.	Сум	10	64%
6.	Здивування	10	52%

Після тестування системи виявилось, що такі емоції, як злість, щастя, спокій та сум, розпізнаються точніше, оскільки містять достатньо інформації для навчання моделі та конкретних облич (положення брів, губ). З іншого боку, система іноді неправильно ідентифікує та плутає такі емоції, як страх та здивування через схожі машині обличчя (брови, губи).

3.6 Висновки до розділу

В даному розділі був описаний процес реалізації системи. Під час досліджень та практичної реалізації встановлено, що сучасні методи глибокого навчання, такі як згорткові нейронні мережі, демонструють високу точність у розпізнаванні емоцій. Ці алгоритми ефективно ідентифікують основні емоції, такі як злість, щастя, спокій та сум, завдяки наявності значної кількості тренувальних даних і характерних ознак.

Однак було виявлено, що недостатня кількість даних для тренування моделі значно знижує точність розпізнавання певних емоцій, наприклад, здивування. Це підкреслює важливість формування збалансованих і різноманітних наборів даних для всіх категорій емоцій. Крім того, система іноді плутає емоції, які мають подібні вирази обличчя, такі як, злість та сум. Це свідчить про необхідність подальшого вдосконалення алгоритмів для кращого розрізнення тонких відмінностей між подібними емоційними станами.

Крім того, була вказана обрана метрика для оцінки якості (accuracy). Результати останніх моделей також були проаналізовані, і найкращий із них показав результат 67% .

Розроблена система здатна розпізнавати емоції в режимі реального часу, що робить її придатною для інтеграції в різні застосунки, включаючи психологічні дослідження, медицину, рекламу, розваги та безпеку. Для підвищення ефективності системи розпізнавання емоцій необхідні подальші дослідження та експерименти з різними архітектурами нейронних мереж і методами передобробки даних. Особливо перспективними є дослідження, спрямовані на покращення розпізнавання складних та рідкісних емоцій.

ВИСНОВКИ

Метою дипломного проєкту було підвищити ефективність розпізнавання емоцій шляхом розробки системи, що її реалізує на основі згорткової нейронної мережі. Програма реалізує виявлення наступних емоцій за допомогою аналізу міміки обличчя: злість, відразу, страх, щастя, спокій (нейтральний вираз), смуток та здивування у режимі реального часу на відео з вебкамери.

Підсумовуючи, можна сказати, що емоційний ШІ дозволяє менеджерам підприємств бути більш емпатичними та зрозумілими для кожного споживача в потрібний момент. Це дає змогу компаніям пропонувати відповідні товари та послуги правильним клієнтам у правильний час, що підвищує залученість клієнтів. Також ШІ значно полегшує роботу маркетологам, допомагаючи визначити, які рекламні ролики найбільше цікавлять цільову аудиторію та що саме слід включити для досягнення найкращих результатів. В інших сферах життя емоційний ШІ може сприяти вирішенню різноманітних соціальних завдань.

В ході роботи над дипломним проєктом, я дослідила високорівневу об'єктно-орієнтовану мову програмування Python. Виявилось, що саме на Python, написання такого алгоритму було дуже зручним. Кількість бібліотек, які він пропонує середньому користувачеві, величезна, а сама мова дуже проста порівняно з іншими мовами високого рівня. Використовуючи методи та засоби досліджених джерел, було спроектовано систему розпізнавання людських емоцій за допомогою архітектури CNN.

Результатом реалізації є готовий програмний продукт, який виконує поставлену задачу. Після глибокого аналізу різноманітних матеріалів про штучний інтелект, можу з впевненістю сказати, що сьогоденні технології розвиваються дуже швидко. Хоча існують проблеми у програмних продуктах, з часом, знаходячи для них рішення, штучний інтелект стане складовою нашого повсякденного життя. Точний час цього переходу важко передбачити, але величезний потенціал цієї сфери вже очевидний

ПЕРЕЛІК ПОСИЛАНЬ

1. Arhiv, Emotion Recognition from Multiple Modalities: Fundamentals and Methodologies. Режим доступу: <https://arxiv.org/abs/2108.10152>, Дата звернення: 17.05.2024 р.
2. BMC Psychology, Development and application of emotion recognition technology — a systematic literature review, Режим доступу: <https://bmcpsychology.biomedcentral.com/articles/10.1186/s40359-021-00593-3>, Дата звернення: 20.05.2024 р.
3. Fastbots, SpaCy vs. TensorFlow: Selecting the Ideal NLP Framework. Режим доступу: <https://fastbots.ai/blog/spacy-vs.-tensorflow-selecting-the-ideal-nlp-framework-for-your-project>, Дата звернення: 21.05.2024 р.
4. IEEE Xplore, A Comprehensive Review of Speech Emotion Recognition Systems, Режим доступу: <https://ieeexplore.ieee.org/document/9637545>, Дата звернення: 21.05.2024 р.
5. IEEE Xplore, A Survey of AI-Based Facial Emotion Recognition: Features, ML & DL Approaches, Режим доступу: <https://ieeexplore.ieee.org/document/9458961>, Дата звернення: 20.05.2024 р.
6. IBM, Bag of Words, Режим доступу: <https://www.ibm.com/topics/bag-of-words>, Дата звернення: 18.05.2024 р.
7. Pratibha Kumari J., AI in Psychology - Transformative Horizons: The Integration of AI in Psychology Services for Enhanced Mental Well-being, Режим доступу: <https://www.linkedin.com/pulse/ai-psychology-transformative-horizons-integration-services-jha-u19vf/>, Дата звернення: 18.05.2024 р.
8. ScienceDirect, Emotion recognition and artificial intelligence: A systematic review, Режим доступу: <https://www.sciencedirect.com/science/article/pii/S1566253521001832>, Дата звернення: 19.05.2024 р.

9. Sensors, New Trends in Emotion Recognition Using Image Analysis by Neural Networks, A Systematic Review, Режим доступу: <https://www.mdpi.com/1424-8220/21/5/1592>, Дата звернення: 20.05.2024 р.
10. TensorFlow, Keras: The high-level API for TensorFlow, Режим доступу: <https://www.tensorflow.org/guide/keras>, Дата звернення: 18.05.2024 р.
11. Саломео А., Автоматизований контент-аналіз: Використання багатства текстових даних, Режим доступу: <https://mindthegraph.com/blog/uk/automated-content-analysis/#:~=Аналіз%20настроїв%3A%20Визначення%20настрою%20або%20настрої%20в%20соціальних%20мережах>, Дата звернення: 21.05.2024 р.
12. Сидоренко І., AI Emotion Recognition, Режим доступу: <https://labelyourdata.com/articles/ai-emotion-recognition>, Дата звернення: 21.05.2024 р.
13. Сліпченко О. О., Клименко С. В., Аналіз сучасного стану інформаційних технологій оцінки психоемоційного стану людини. Дніпровський національний університет імені Олеся Гончара. Режим доступу: <https://tit.dut.edu.ua/index.php/telecommunication/article/view/2417/2298>, Дата звернення: 21.05.2024 р.
14. Чуйко Г., Актуальність застосування систем аналізу емоцій. Тернопільський національний технічний університет імені Івана Пулюя, Режим доступу: https://elartu.tntu.edu.ua/bitstream/lib/35110/2/MCTD_2021_Chuiko_H-Actuality_of_application_135-137.pdf, Дата звернення: 19.05.2024 р.
15. Дерев'янюк С., Емоційний штучний інтелект у професійній підготовці майбутніх психологів, Режим доступу: https://www.researchgate.net/publication/349655171_EMOCIJNIJ_STUCNIJ_IN_TELEKT_U_PROFESIJNIJ_PIDGOTOVCI_MAJBUTNIH_PSIHOLOGIV, Дата звернення: 19.05.2024 р.

16.Кемпбелл С., Підручник SciPy у Python: що таке, бібліотека, функції та приклади, Режим доступу: <https://www.guru99.com/uk/scipy-tutorial.html>, Дата звернення: 18.05.2024 р.

17.Кисіль Т.М., Зінченко В.В., Хотиненко А.В., АНАЛІЗ ВИКОРИСТАННЯ ЧАТ-БОТІВ У СФЕРІ ОХОРОНИ ЗДОРОВ'Я / Науково-практична конференція «Сучасні досягнення компанії HEWLETT PACKARD ENTERPRISE в галузі ІТ та нові можливості їх вивчення і застосування». Збірник тез. – К.: ДУІКТ, 2023р.,с.14-16,Режим

доступу:https://drive.google.com/file/d/1EZMzK2iXUuyTMy1BY58n_PZPJ-6fvk8m/view?usp=sharing, Дата звернення: 19.05.2024 р.

18.Костенко Н., Батаєва К., Іванова В., Сучасні методики контент-аналізу, Режим доступу: https://www.researchgate.net/publication/326736532_Sucasni_metodiki_kontent-analizu, Дата звернення: 17.05.2024 р.

19.Хотиненко А.В., Кисіль Т.М., Аналіз застосувань комп'ютерних систем штучного інтелекту в медицині / Науково-практична конференція «Проблеми комп'ютерної інженерії». К.: ДУІКТ, 2023 р. с.64-66, Режим доступу: https://duikt.edu.ua/uploads/p_2626_33497568.pdf, Дата звернення: 21.05.2024 р.

20.Evergreens, Emotion AI. Режим доступу:<https://evergreens.com.ua/ua/articles/emotion-ai.html>, Дата звернення: 19.05.2024 р.

ДОДАТОК А. ПРОГРАМНИЙ КОД ПРОЄКТОВАНОЇ СИСТЕМИ

`video_emotion_work.py` (Головний файл для розпізнавання емоцій на відео-потоці)

```
from statistics import mode

import cv2

from keras.models import load_model

import numpy as np

from utils.datasets import get_labels
from utils.inference import detect_faces
from utils.inference import draw_text
from utils.inference import draw_bounding_box
from utils.inference import apply_offsets
from utils.inference import load_detection_model
from utils.preprocessor import preprocess_input

# параметри для завантаження моделей та назв емоцій
detection_model_path =
'../trained_models/detection_models/haarcascade_frontalface_defa
ult.xml'
emotion_model_path =
'../trained_models/emotion_models/fer2013_mini_XCEPTION102-
0.66.hdf5'
emotion_labels = get_labels( ' fer2013 ' )

# гіперпараметри для рамки
frame_window = 10
```

```
emotion_offsets = (20, 40)
```

```
# завантаження моделей
```

```
face_detection = load_detection_model(detection_model_path)
```

```
emotion_classifier = load_model(emotion_model_path,  
compile=False)
```

```
emotion_target_size = emotion_classifier.input_shape[1:3]
```

```
emotion_window = []
```

```
# запуск відео потоку
```

```
cv2.namedWindow('window_frame')
```

```
video_capture = cv2.VideoCapture(0)
```

```
while True:
```

```
    bgr_image = video_capture.read()[1]
```

```
    gray_image = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2GRAY)
```

```
    rgb_image = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2RGB)
```

```
    faces = detect_faces(face_detection, gray_image)
```

```
    for face_coordinates in faces:
```

```
        x1, x2, y1, y2 = apply_offsets(face_coordinates,  
emotion_offsets)
```

```
        gray_face = gray_image[y1:y2, x1:x2]
```

```
        try:
```

```
            gray_face = cv2.resize(gray_face,  
(emotion_target_size))
```

```
        except :
```

```
            continue
```



```

        gray_face = preprocess_input(gray_face, True)
        gray_face = np. expand_dims(gray_face, 0)
        gray_face = np. expand_dims(gray_face, -1)
        emotion_prediction = emotion_classifier.
predict(gray_face)
        emotion_probability = np.
max(emotion_prediction)
        emotion_label_arg =
np.argmax(emotion_prediction)
        emotion_text =
emotion_labels[emotion_label_arg]
        emotion_window.append(emotion_text)

        if len(emotion_window) > frame_window:
            emotion_window.pop(0)
        try :
            emotion_mode = mode(emotion_window)
        except :
            continue

        if emotion_text == 'angry' :
            color = emotion_probability * np.
asarray((255, 0, 0))
        elif emotion_text == 'sad' :
            color = emotion_probability * np.
asarray((0, 0, 255))
        elif emotion_text == 'happy' :

```

```

        color = emotion_probability * np.
asarray((0, 255, 0))
        elif emotion_text == 'surprise' :
            color = emotion_probability * np.
asarray((255, 255, 0))
        else :
            color = emotion_probability * np.
asarray((255, 255, 255))

        color = color. astype( int )
        color = color. tolist()

        draw_bounding_box(face_coordinates,
rgb_image, color)
        draw_text(face_coordinates, rgb_image,
emotion_mode,
                    color, 0, -45, 2, 2)

        bgr_image = cv2.cvtColor(rgb_image, cv2.COLOR_RGB2BGR)
cv2.imshow('window_frame', bgr_image)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
video_capture.release()
cv2.destroyAllWindows()

```

train_emotion_classifier.py (Файл для проведення тренування нейронної мережі)

```
from keras.callbacks import CSVLogger, ModelCheckpoint,
EarlyStopping

from keras.callbacks import ReduceLROnPlateau

from keras.preprocessing.image import ImageDataGenerator

from models.cnn import mini_XCEPTION
from utils.datasets import DataManager
from utils.datasets import split_data
from utils.preprocessor import preprocess_input

# параметри
batch_size = 32
num_epochs = 10000
input_shape = (64, 64, 1)
validation_split = .2
verbose = 1
num_classes = 7
patience = 50
base_path = '../trained_models/emotion_models/'
data_generator = ImageDataGenerator(
    featurewise_center=False,
    featurewise_std_normalization=False,
    rotation_range=10,width_shift_range=0.1,
    height_shift_range=0.1,zoom_range=.1,
    horizontal_flip=True)

# параметри моделі/компіляція
```

```

model = mini_XCEPTION(input_shape, num_classes)
model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()

datasets = ['fer2013']
for dataset_name in datasets:
    print ( ' Training dataset : ', dataset_name)

    # зворотні дані
    log_file_path = base_path + dataset_name +
'_emotion_training.log'
    csv_logger = CSVLogger(log_file_path, append=False)
    early_stop = EarlyStopping('val_loss', patience=patience)
    reduce_lr = ReduceLRonPlateau('val_loss', factor=0.1,

patience=int (patience/4), verbose=1)
    trained_models_path = base_path + dataset_name +
'_mini_XCEPTION'
    model_names = trained_models_path + '. {epoch:02d} -
{val_acc:.2f} . hdf5'
    model_checkpoint = ModelCheckpoint(model_names, 'val_loss',
verbose=1,

save_best_only=True)
    callbacks = [model_checkpoint, csv_logger, early_stop,
reduce_lr]

```

```
# завантаження бази даних
```

```
data_loader = DataManager(dataset_name,  
image_size=input_shape[:2])  
faces, emotions = data_loader.get_data()  
faces = preprocess_input(faces)  
num_samples, num_classes = emotions.shape  
train_data, val_data = split_data(faces, emotions,  
validation_split)  
train_faces, train_emotions = train_data  
model.fit_generator(data_generator.flow(train_faces,  
train_emotions,  
batch_size), steps_per_epoch=len(train_faces) / batch_size,  
epochs=num_epochs, verbose=1, callbacks=callbacks,  
validation_data=val_data)
```

спн.py (Файл із архітектурою нейромережі)

```
from keras.layers import Activation, Convolution2D, Dropout,  
Conv2D  
from keras.layers import AveragePooling2D, BatchNormalization  
from keras.layers import GlobalAveragePooling2D  
from keras.models import Sequential  
from keras.layers import Flatten  
from keras.models import Model  
from keras.layers import Input  
from keras.layers import MaxPooling2D  
from keras.layers import SeparableConv2D  
from keras import layers  
from keras.regularizers import l2
```

```

def mini_XCEPTION(input_shape, num_classes,
l2_regularization=0.01):
    regularization = l2(l2_regularization)

    # початок
    img_input = Input(input_shape)
    x = Conv2D(8, (3, 3), strides=(1, 1),
kernel_regularizer=regularization,
                use_bias=False)(img_input)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Conv2D(8, (3, 3), strides=(1, 1),
kernel_regularizer=regularization,
                use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    # 1 коло
    residual = Conv2D(16, (1, 1), strides=(2, 2),
padding='same',
                use_bias=False)(x)
    residual = BatchNormalization()(residual)
    x = SeparableConv2D(16, (3, 3), padding='same',
kernel_regularizer=regularization,
                use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = SeparableConv2D(16, (3, 3), padding='same',

```

```
        kernel_regularizer=regularization,  
use_bias=False)(x)
```

```
    x = BatchNormalization()(x)  
    x = MaxPooling2D((3, 3), strides=(2, 2),  
padding='same')(x)  
    x = layers.add([x, residual])
```

2 коло

```
    residual = Conv2D(32, (1, 1), strides=(2, 2),  
padding='same',  
use_bias=False)(x)
```

```
    residual = BatchNormalization()(residual)  
    x = SeparableConv2D(32, (3, 3), padding='same',  
        kernel_regularizer=regularization,  
use_bias=False)(x)
```

```
    x = BatchNormalization()(x)
```

```
    x = Activation('relu')(x)
```

```
    x = SeparableConv2D(32, (3, 3), padding='same',
```

```
        kernel_regularizer=regularization, use_bias=False)(x)
```

```
    x = BatchNormalization()(x)
```

```
    x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
```

```
    x = layers.add([x, residual])
```

3 коло

```
    residual = Conv2D(64, (1, 1), strides=(2, 2),  
padding='same',  
use_bias=False)(x)
```

```
    residual = BatchNormalization()(residual)
```

```

x = SeparableConv2D(64, (3, 3), padding='same',
                    kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(64, (3, 3), padding='same',
                    kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])

# 4 коло
residual = Conv2D(128, (1, 1), strides=(2, 2),
                 padding='same',
                 use_bias=False)(x)
residual = BatchNormalization()(residual)
x = SeparableConv2D(128, (3, 3), padding='same',
                    kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(128, (3, 3), padding='same',
                    kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
x = Conv2D(num_classes, (3, 3), padding='same')(x)
# kernel_regularizer=regularization,
x = GlobalAveragePooling2D()(x)

```



```
output = Activation('softmax', name='predictions')(x)
model = Model(img_input, output)
return model
```

inference.py (Файл із функціями підтримки)

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
from keras.preprocessing import image

def load_image(image_path, grayscale=False, color_mode='rgb',
target_size=None):
    pil_image = image.load_img(image_path, grayscale,
color_mode,target_size)
    return image.img_to_array(pil_image)

def load_detection_model(model_path):
    detection_model = cv2.CascadeClassifier(model_path)
    return detection_model

def detect_faces(detection_model, gray_image_array):
    return detection_model.detectMultiScale(gray_image_array,
1.3, 5)

def draw_bounding_box(face_coordinates, image_array, color):
    x, y, w, h = face_coordinates
    cv2.rectangle(image_array, (x, y), (x + w, y + h), color,
2)
```

```

def apply_offsets(face_coordinates, offsets):
    x, y, width, height = face_coordinates
    x_off, y_off = offsets

    return (x - x_off, x + width + x_off, y - y_off, y + height +
            y_off)

def draw_text(coordinates, image_array, text, color, x_offset=0,
              y_offset=0,

              font_scale=2, thickness=2):
    x, y = coordinates[:2]
    cv2.putText(image_array, text, (x + x_offset, y + y_offset),
                cv2.FONT_HERSHEY_SIMPLEX,
                font_scale, color, thickness,
                cv2.LINE_AA)

def get_colors(num_classes):
    colors = plt.cm.hsv(np.linspace(0, 1,
    num_classes)).tolist()
    colors = np.asarray(colors) * 255
    return colors

```

datasets.py (Файл для завантаження бази даних фото)

```

from scipy.io import loadmat
import pandas as pd
import numpy as np
from random import shuffle

```

```
import os
import cv2

class DataManager(object):
    def __init__(self, dataset_name='imdb',
                 dataset_path=None, image_size=(48,
48));

        self.dataset_name = dataset_name
        self.dataset_path = dataset_path
        self.image_size = image_size
        if self.dataset_path is not None:
            self.dataset_path = dataset_path
        elif self.dataset_name == 'fer2013':
            self.dataset_path = '../datasets/fer2013/fer2013.csv'
        else:
            raise Exception(
                'Incorrect dataset name, please input
fer2013')

    def get_data(self):
        if self.dataset_name == 'fer2013':
            ground_truth_data = self._load_fer2013()

    def _load_fer2013(self):
        data = pd.read_csv(self.dataset_path)
        pixels = data['pixels'].tolist()
        width, height = 48, 48
```

```

    faces = []
    for pixel_sequence in pixels:
        face = [int(pixel) for pixel in
pixel_sequence.split(' ')]
        face = np.asarray(face).reshape(width, height)
        face = cv2.resize(face.astype('uint8'),
self.image_size)
        faces.append(face.astype('float32'))
    faces = np.asarray(faces)
    faces = np.expand_dims(faces, -1)
    emotions = pd.get_dummies(data['emotion']).as_matrix()
    return faces, emotions

def get_labels(dataset_name):
    if dataset_name == 'fer2013':
        return {0: 'angry', 1: 'disgust', 2: 'fear', 3: 'happy',
                4: 'sad', 5: 'surprise', 6: 'neutral'}
    else :
        raise Exception('Invalid dataset name')
def get_class_to_arg(dataset_name='fer2013'):
    if dataset_name == 'fer2013':
        return {'angry': 0, 'disgust': 1, 'fear': 2, 'happy':
3, 'sad': 4,
                'surprise': 5, 'neutral': 6}
    else :
        raise Exception('Invalid dataset name')

def split_data(x, y, validation_split=.2):

```

```

    num_samples = len(x)
    num_train_samples = int((1 -
validation_split)*num_samples)
    train_x = x[:num_train_samples]
    train_y = y[:num_train_samples]
    val_x = x[num_train_samples:]
    val_y = y[num_train_samples:]
    train_data = (train_x, train_y)
    val_data = (val_x, val_y)
    return train_data, val_data

```

preprocessor.py (Файл препроцесор вхідних даних)

```

import numpy as np
from scipy.misc import imread, imresize

def preprocess_input(x, v2=True):
    x = x.astype('float32')
    x = x / 255.0
    if v2:
        x = x - 0.5
        x = x * 2.0
    return x

def _imread(image_name):
    return imread(image_name)

def _imresize(image_array, size):
    return imresize(image_array, size)

```

```
def to_categorical(integer_classes, num_classes=2):  
    integer_classes = np.asarray(integer_classes, dtype='int')  
    num_samples = integer_classes.shape[0]  
    categorical = np.zeros((num_samples, num_classes))  
    categorical[np.arange(num_samples), integer_classes] = 1  
    return categorical
```

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО КОНТЕНТ -АНАЛІЗУ ЕМОЦІЙ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ »

Виконала: здобувачка вищої освіти гр. ШІД-41
Аліна ХОТИНЕНКО
Керівник: ст.викладач, Тетяна КИСІЛЬ

2024 рік

Мета роботи: підвищення ефективності роботи системи розпізнавання емоцій. Аналіз існуючих програм та виявлення наявних недоліків, щоб в подальшому удосконалити свою систему. За допомогою отриманих знань створити систему, яка розпізнає емоції.

Об'єкт дослідження: процес роботи системи розпізнавання емоцій осіб та підвищення їх якості.

Предмет дослідження: система розпізнавання емоцій осіб з використанням штучного інтелекту.

Основні завдання кваліфікаційної роботи:

- Структура штучної нейронної мережі
- Особливості розпізнавання емоцій осіб
- Бібліотеки для реалізації проєкту
- Реалізація проєкту

Структура штучної нейронної мережі

Нейронні мережі містять щонайменше три типи прошарків: вхідний, прихований та вихідний. Прошарок вхідних нейронів отримує дані або з вхідних файлів, або безпосередньо з електронних датчиків.

Вихідний прошарок пересилає інформацію безпосередньо у зовнішнє середовище, до вторинного комп'ютерного процесу, або до іншого пристрою.

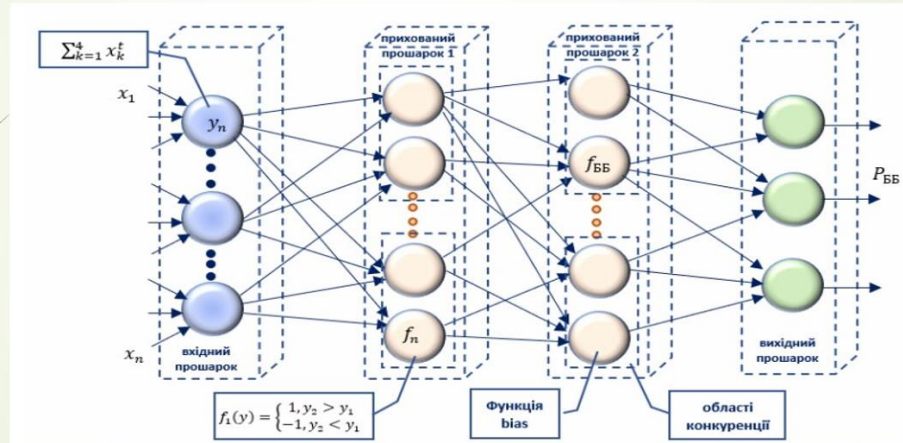


Рис. 1 – Структура багатошарової штучної НМ

Згорткова нейронна мережа складається з трьох основних типів шарів:

- згорткових;
- підсумовуючих;
- повністю зв'язаних.

Згорткові шари використовуються для виявлення різних ознак у вхідних даних, лінійні краї зображень чи текстури.

Підсумовуючі шари зменшують розмір зображення, об'єднуючи інформацію у певних областях.

Повністю зв'язані шари обробляють ці ознаки та генерують кінцевий результат, такий як класифікація зображення. Для проекту розроблена згорткова нейронна мережа на основі архітектури LeNet.

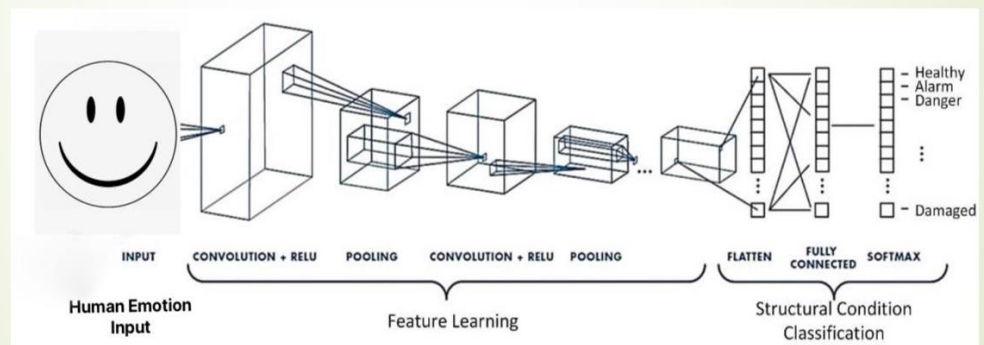


Рис. 2 - Загальна архітектура згорткової нейронної мережі

Особливості розпізнавання емоцій осіб

Розглянемо характеристики емоцій у поєднанні брів та рота:

- ✓ **Смуток.** Характерні ознаки смутку – кінчики брів та рота знижуються.
- ✓ **Щастя.** При цій емоції брови та кінчики рота підіймаються.
- ✓ **Гнів.** Для емоції гніву характерні зниження та зморщення брів, а рот при цьому відкривається та знижуються кінчики.
- ✓ **Образа.** У разі відчуття образи брови знижуються, а рот піднімається і знижуються прикінцеві значення лінії.
- ✓ **Здивування.** При здивуванні брови підіймаються, а рот відкривається.

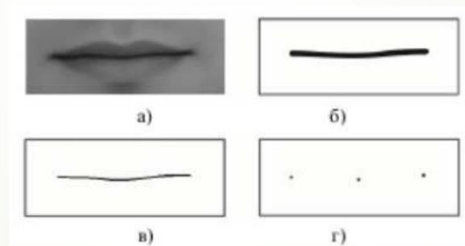


Рис. 3 – Процес виділення ключових точок (а – напівтонове зображення; б – бінарне зображення; в – застосування градієнтної маски; г – локалізація ключових точок)

Використані бібліотеки для реалізації проєкту

Для роботи зі CNN в програмі були використані такі бібліотеки :

Pandas, OpenCV, Numpy, Pillow, Matplotlib, H5py, Keras, Tensorflow, Scipy.

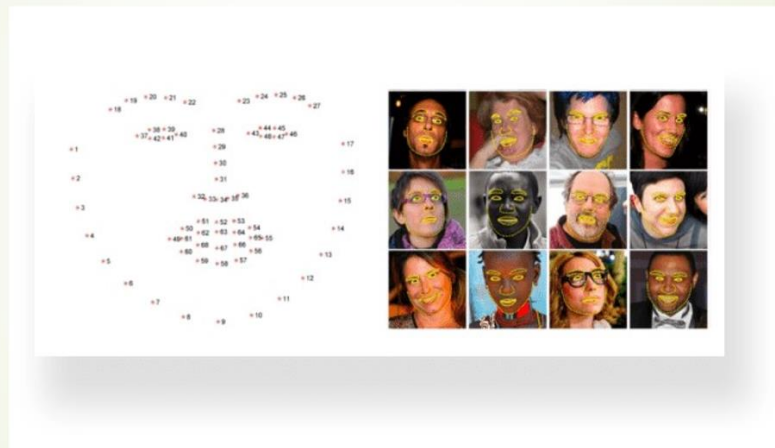


Рис. 4 - Розпізнавання обличчя на базі OpenCV

Тестування роботи програми

Для розробки застосунку, я обрала мову програмування Python. Переваги використання Python для комп'ютерного зору:

- ✓ проста у написанні коду,
- ✓ обширні бібліотеки для машинного навчання,
- ✓ відкритий вихідний код,
- ✓ можливість прямої інтеграції з веб-фреймворками.

Результатом запуску є вікно програми яке транслює зображення з камери комп'ютера. Система розпізнає обличчя та емоцію особи.

Результат відразу відображається на екрані і може змінюватись, паралельно зміні виразу обличчя.

- Перевіriamo як система розпізнає емоції:
1. Злість;
 2. Сум;
 3. Щастя (Рис. 5);
 4. Спокій (Нейтральний вираз);
 5. Здивування.

Результат розпізнавання обличчя

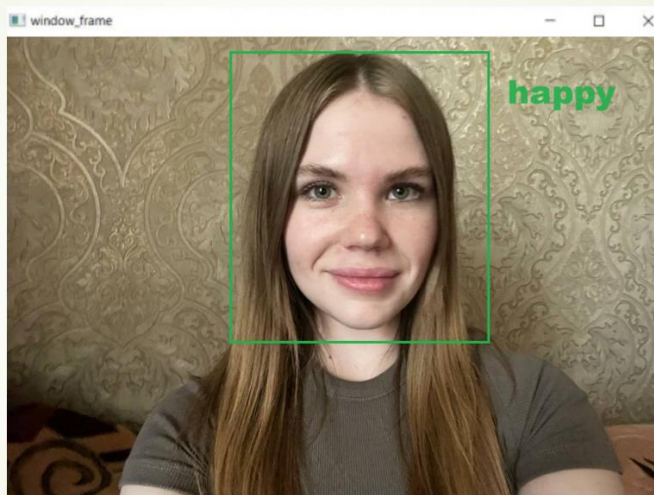


Рис.5 - Результат емоції «Щастя»

№	Емоція	Кількість фото для тестування	Оцінка ефективності розпізнавання
1.	Злість	10	67%
2.	Страх	10	52%
3.	Щастя	10	67%
4.	Спокій(Нейтральний вираз обличчя)	10	65,7%
5.	Сум	10	64%
6.	Здивування	10	52%

Система розпізнає емоції, але через погану якість зображення, часто допускається помилок. Для точнішого розпізнавання емоцій, приміщення має бути добре освітленим і обличчя повинно розташовуватися в анфас якомога ближче до камери.

Результати останніх моделей також були проаналізовані, і найкращий із них показав результат **67%**. Програма потребує подальшої роботи над нею та вдосконалення.

ВИСНОВКИ

1. Проведено аналіз методів та особливостей функціонування системи розпізнавання емоцій людини з використанням штучного інтелекту.

2. У ході виконання роботи було розглянуто:

- методи вирішення задачі розпізнавання емоцій,
- існуючі аналоги систем,
- архітектури глибоких нейронних мереж

3. Застосовано бібліотеку OpenCV, яка слугує для аналізу і обробки зображення обличчя у режимі реального часу на відео з вебкамери ноутбуку.

Результат можна покращити шляхом збільшення вхідних зображень, використання методів аугментації даних, таких як зміщення, обертання, масштабування, що допоможе розширити набір даних.

Створена система розпізнавання емоцій, може бути застосована в різних галузях, таких як безпека, реклама, розваги, психологія, медицина.



Дякую за увагу!