МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Кафедра Комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

Аліна САВЧЕНКО.

«_____» _____ 2022р.

КВАЛІФІКАЦІЙНА РОБОТА

(ДИПЛОМНА РОБОТА, ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ **"МАГІСТР"**

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ

"ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ"

Тема: " Використання нейронної мережі для розпізнавання емоцій"

Виконавець: студентка групи УС-212М Шимчук Яна Вячеславівна

Керівник: к.т.н., доцент Моденов Юрій Борисович

Нормоконтролер: Ігор РАЙЧЕВ

Київ – 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки, комп'ютерної та програмної інженерії

Кафедра: Комп'ютерних інформаційних технологій

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

ЗАТВЕРДЖУЮ
Завідувач випускової кафедри
_____Аліна САВЧЕНКО
«_____»_____ 2022 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Шимчук Яни Вячеславівни
(прізвище, ім'я, по батькові)

1. Тема роботи: «Використання нейронної мережі для розпізнавання емоцій» затверджена наказом ректора № 1774/ст від 28.09.2022р.

2. Термін виконання роботи: з 26 вересня 2022 року по 27 листопада 2022 року.

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

4. Зміст пояснювальної записки: вступ, огляд предметної області, нейронна мережа, реалізація системи розпізнавання емоцій, висновки.

5. Перелік обов'язкового ілюстративного матеріалу: архітектура нейронних мереж, архітектура згорткових нейронних мереж, демонстрація роботи системи.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання на дипломну роботу, створення плану дипломної роботи та побудова плану-графіку виконання робіт.	26.09.2022 – 28.09.2022	
2.	Огляд та аналіз наукової літератури по темі дипломної роботи та написання Розділу 1.	29.09.2022 – 09.10.2022	
3.	Написання Розділу 2 дипломної роботи.	10.10.2022 –20.10.2022	
4.	Написання Розділу 3 і Розділу 4 дипломної роботи. Завершення створення пояснювальної записки дипломної роботи.	21.10.2022 – 31.10.2022	
5.	Оформлення та друк пояснювальної записки.	01.11.2022 – 07.11.2022	
6.	Створення презентації, доповіді та підготовка до захисту дипломної роботи.	08.11.2022 – 15.11.2022	
7.	Підготовка матеріалів дипломної роботи для передачі секретарю ДЕК (папка, конверт, диск із файлом диплому, рецензія, відгук).	16.11.2022 – 18.11.2022	

7. Дата видачі завдання 26 вересня 2022р.

Керівник дипломної роботи _____ Юрій МОДЕНОВ
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Яна ШИМЧУК
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Використання нейронної мережі для розпізнавання емоцій» складається зі вступу, трьох розділів, висновку та списку бібліографічних посилань і містить 72 сторінки та 19 рисунків. Список бібліографічних посилань складається з 20 найменувань.

Ключові слова: РОЗПІЗНАВАННЯ ЕМОЦІЙ, НЕЙРОННА МЕРЕЖА, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА.

Актуальність роботи: технології розпізнавання емоцій людини стрімко розвиваються, нині вони стали корисними в багатьох сферах нашого життя. Ці системи оптимізували роботу багатьох процесів.

Мета дипломної роботи: дослідити як працюють системи розпізнавання емоцій, сфери в яких вони можуть бути використані. Також проаналізувати існуючі програми та знайти наявні недоліки, щоб в подальшому удосконалити свою систему. За допомогою отриманих знань створити систему, яка розпізнає емоції.

Об'єкт дослідження: системи розпізнавання емоцій людини за допомогою нейронних мереж.

Предмет дослідження: розробка системи розпізнавання емоцій людини за допомогою нейронних мереж.

Теоретична основа: україномовні та зарубіжні джерела пов’язані з обраною темою, збірки статей наукових конференцій, тематичні публікації на сайтах.

Теоретична і практична значимість: на основі отриманих знань можна безкоштовно побудувати свою програмну реалізацію системи розпізнавання емоцій з відкритим кодом та без ліцензійних обмежень за досить короткий проміжок часу.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

ШІ – Штучний інтелект

ML (Machine learning) – Машинне навчання

FER (Facial emotion recognition) – Розпізнавання емоцій на обличчі

SaaS (Software as a service) – Програмне забезпечення для розпізнавання облич як послуга

API (Application Programming Interface) – Програмний інтерфейс програми

LFW (Labeled Faces in the Wild) – це база даних фотографій облич, призначена для вивчення проблеми необмеженого розпізнавання облич

SDK (software development kit) – набір із засобів розробки, утиліт і документації, який дає програмістам змогу створювати прикладні програми за визначеною технологією або для певної платформи (програмної або програмно-апаратної)

SLA (Service-level agreement) – угода між постачальником послуг і користувачем про рівень послуг

БД – База даних

НМ – Нейгронна мережа

ШНМ – Штучна нейронна мережа

ГНМ / DNN – Глибинна нейронна мережа

CNN (Convolutional neural network) – Згорткова нейронна мережа

FC (Fully Connected Layer) – Повнозв'язний шар

CV (Computer vision) – Комп’ютерний зір

PyPI (Python Package Index) – каталог програмного забезпечення, написаного на мові програмування Python

Pip – система керування пакунками, яка використовується для встановлення та управління програмними пакетами, які написані на Python

ВСТУП

Розпізнавання емоцій - технологія розпізнавання осіб, котра з роками розвивається та розширюється, є однією з різноманітних технологій розпізнавання осіб. Наразі певна програма може аналізувати та оцінювати емоції обличчя людини завдяки використанню програмного забезпечення для розпізнавання емоцій обличчя. Ця програма імітує роботу людського мозку за допомогою складного розподілу зображень, що дозволяє їй також розпізнавати емоції.

Штучний інтелект (ШІ) - це те, що розпізнає і вивчає різні вирази обличчя, щоб використовувати їх у поєднанні з інформацією, яка йому надається. Такий підхід допомагає органам влади ідентифікувати емоції людини, використовуючи лише технологію, що має важливе значення для низки завдань, включаючи розслідування та інтерв'ю.

Використовуючи аналіз виразу обличчя, розпізнавання емоцій може ідентифікувати різні емоції обличчя. Наведу перелік найпоширеніших виразів обличчя, які відповідають різним емоціям : гнів - підняте підборіддя, зосереджений погляд, опущені та насуплені брови; радість переважно виділяється піднятими куточками рота; здивування: великі очі, вигнуті брови, відвисла щелепа; великі очі, роззявлений рот, зморщений лоб - це зазвичай ознаки страху; смуток характеризується насупленими бровами і надутим чолом; кусання губ може символізувати тривогу.

Глибоке навчання - це функція штучного інтелекту для розпізнавання облич, яка імітує те, як людський мозок розробляє шаблони для виявлення об'єктів і навіть прийняття рішень. Це підмножина штучного інтелекту і техніки для навчання машин.

Основою глибокого навчання є нейромережі. Нейронна мережа - це алгоритм, який імітує роботу мозку і був натхненний структурою кори головного мозку. З часом він набув популярності, особливо останнім часом завдяки своїм перевагам в ідентифікації емоцій.

Нейронна мережа містить три шари: вхідний шар, прихований шар і вихідний шар, як і кора головного мозку. У нейронну мережу можуть бути додані бажані дані, і всі вони проходять через ці шари. Кожен шар змінює кожне вхідне значення, намагаючись отримати бажаний і цільовий вихід.

Щоб бути впевненим, що вихідні дані є точними і придатними, програма розпізнавання емоцій проходить навчання. Для того, щоб алгоритми працювали, необхідно розуміти входи і виходи. Таким чином, алгоритми повинні бути здатні виявляти людські емоції. Для цього використовуються два методи:

1. Категоріальний

Згідно з цією методикою, існує обмежений набір емоцій, які потрапляють у різні задані класи. Презирство, горе, шок, відраза, гнів, радість і страх є одними з емоцій, які згадуються.

2. Вимірний

Виходячи з цього методу, розмірні емоції існують у вигляді спектру і не можуть бути конкретно визначені. Циркумплексна модель афекту використовує два виміри, але емоційний стан PAD має три.

При використанні машинного навчання для їх моделювання обрана методика має деякі суттєві наслідки. Ймовірно, буде побудований класифікатор, і тексти та зображення будуть позначені людськими емоціями, такими як "щасливий", "сумний" і т.д., коли буде обрана категоріальна модель людських емоцій. Однак, якщо буде прийнята вимірна модель, результати повинні бути на ковзній шкалі.

Розпізнавання емоцій сьогодні використовується для багатьох речей, про деякі з яких люди навіть не здогадуються. Ось список деяких сфер, де вже застосовують дану технологію: заходи з безпеки, підтримка для персоналу, сервіс підтримки клієнтів, для допомогти дітям зі спеціальними потребами, оцінювання залученності аудиторії, оцінка відеоігор по реакціям людей , охорона здоров'я та ще багато інших.

Звісно алгоритми ідентифікації емоцій ще далекі від досконалості. Навіть якщо вони дійсно можуть сприймати емоції, то все одно стикаються з проблемами чи навіть створюють їх. Наприклад, система може інтерпретувати невербальні

сигнали як більш тривожні, ніж вони є насправді. Крім того, оскільки ШІ природно асоціює певні вирази обличчя з певними емоціями, він не здатний розрізняти справжні і фальшиві емоції і може бути легко обманутий.

Штучному інтелекту також важко виносити точні судження, оскільки він не здатний зрозуміти, як змінюються емоції в різних культурах.

Загалом, такі упередження в розпізнаванні емоцій можуть призвести до серйозних помилкових припущень і серйозних непорозумінь, якщо їх не виправити. Подальший прогрес буде досягнутий з урахуванням потенціалу подібних помилок.

Проте згідно з дослідженням, проведеним у 2015 році, результати ідентифікації покращуються, а саме, за рахунок видалення з вхідних фотографій змінних, що збивають з пантелику, та використання достатнього освітлення.Технологія розпізнавання емоцій ШІ регулярно досліджується і вдосконалюється з метою виявлення та усунення нових небезпек і викликів, а також запобігання будь-яким суспільним етичним і культурним труднощам.

Мета роботи - реалізувати алгоритм розпізнавання емоцій людини у реальному часі за допомогою відеокамери. Для того щоб досягти ідентифікації емоцій було використано згорткову нейронну мережу та матеріал з бази даних Kaggle, що я використала для випробування класифікатора, а також навчання алгоритму. Щоб визначати емоції в режимі реального часу, знімаючи вхідні дані з комп'ютерної веб-камери я використала бібліотеку OpenC.

РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Сфери застосування

Використання технології розпізнавання облич увійшло в кожен аспект нашого повсякденного життя, від розблокування телефонів до доступу до приватних файлів на ноутбуках. Останнім часом ведеться багато дискусій про використання технології ідентифікації емоцій за допомогою міміки. ШІ створює програмне забезпечення для виявлення емоцій на обличчі, яке в поєднанні з машинним навчанням (ML) і глибоким навчанням змінить багато секторів з точки зору безпеки, зручності і більш розумних результатів на основі ШІ. FER на основі ШІ може бути застосований в декількох секторах.

1.1.1. Освіта

Одним із наслідків пандемії Covid 19 було те, що будь – яка навчальна структура мала змінити режим навчання на оффлайн або як його ще називають - дистанційне. Звісно в такій формі навчання є і свої переваги, але незважаючи на них, навчальна система була змушена зіткнутись з багатьма недоліками такого формату. Платформи для навчання, які були розроблені на базі ШІ сканують м’язи лиця студентів чи учнів і за допомогою цього розпізнають 7 основних емоцій людини. Розробники вважають, що саме завдяки цьому принципу викладачам, вчителям, а також репетиторам буде набагато легше найти особистий підхід до кожного учня чи студента, тому що з таким програмним забезпеченням педагог здатний оцінити рівень зацікавленості кожного.

Такий додаток до освітнього процесу має помітно знизити відсоток незнання матеріалу, адже він не тільки відслідковує емоцій, а також постійно надає різні не типові завдання, опираючись на емоційний стан, воно генерує тести в стилі гри, для того щоб процес отримання знань був більш

різноманітним. В своєму потенціалі, такий ШІ, здатний на те, щоб його віртуальний клас був кращим, аніж у будь – якого викладача.

1.1.2. Медицина

З тих пір, як життя і здоров'я стали ключовими цінностями в більшості сучасних цивілізацій, бізнес в сфері HealthTech значно виріс. Ідентифікація емоцій сьогодні надає низку можливостей у таких різноманітних галузях, як психологія, неврологія і навіть біомедична інженерія.

При правильному використанні таке програмне забезпечення може мати значний вплив на автоматизацію діагностики різноманітних захворювань (зокрема, мозкових та психологічних). Це має полегшити роботу лікарів та надати пріоритет пацієнтам з серйозними захворюваннями, які потребують швидкого лікування.

Однак, враховуючи складність анотування та безпеки, пов'язану з конфіденційним характером даних, пов'язаних зі здоров'ям, кількість неточностей та упереджень в алгоритмах виявлення емоцій, розроблених для HealthTech, може бути вражаючою. Щоб уникнути таких помилок, дизайн моделі виявлення даних і емоцій повинен бути максимально якісним.

Технологія розпізнавання емоцій може допомогти медичним працівникам зрозуміти пацієнтів, які мають труднощі з нормальним вираженням своїх емоцій. Це дасть змогу лікарям надавати кращу допомогу своїм пацієнтам.

1.1.3. Тестування для відеоігор

Відеоігри створюються з розрахунку на певну аудиторію. Кожна відеогра спрямована на те, щоб викликати у споживачів певну поведінку та набір почуттів. На етапі тестування людей просять пограти в гру протягом певного часу, а їхні відгуки використовуються для покращення кінцевого продукту. Використання розпізнавання виразу обличчя може допомогти визначити почуття, які відчуває користувач в режимі реального часу під час гри, без необхідності вручну аналізувати відео повністю.

Такий зворотній зв'язок можна отримати, оцінюючи пряму трансляцію користувача та розпізнаючи його вираз обличчя. Хоча роздратування і лють є переважаючими емоціями в складних відеоіграх, використання розпізнавання виразу обличчя може допомогти визначити, які емоції відчуваються в певні моменти гри. Під час гри також ймовірно, що будуть відзначені деякі несподівані або негативні почуття. Отримання інформації від користувачів, які грали в гру, може бути марнотратним. Це пов'язано з тим, що досвід важко описати словами. Крім того, гравці можуть бути не в змозі згадати, що вони переживали в емоційному плані протягом гри. Розпізнавання емоцій обличчя є корисною технікою для виходу за рамки усного або письмового введення і розпізнавання того, що відчуває користувач. Коли вхідні дані отримуються в такому стилі, вони стають дійсно не нав'язливими з точки зору користувацького досвіду. У той же час, цей тип зворотного зв'язку є більш надійним, ніж інші.

1.1.4. Автомобільна промисловість

За даними Міністерства транспорту США, помилки водія є причиною близько 95% ДТП зі смертельними наслідками, саме тому автовиробники в усьому світі все більше зосереджуються на тому, щоб зробити транспортні засоби більш персоналізованими та безпечними для водіння. Для виробників має сенс використовувати ШІ, оскільки система розпізнавання емоцій може виявляти незначні зміни міміки обличчя, які передують сонливості, і надавати водієві персоналізовані повідомлення з проханням зробити перерву на каву, змінити музику або температуру в салоні.

1.1.5. Забезпечення правопорядку

Інша проблема з алгоритмами розпізнавання емоцій - це системи спостереження, які дозволяють автоматизувати роботу поліції та прикордонного контролю. Як виявляти небезпечних людей? Більш фундаментально, як охарактеризувати небезпечну особу? Чи є розпізнавання облич достатньо міцним фундаментом для такої діяльності?

Розпізнавання облич є найбільш поширеною технологією спостереження у світі. До міст, за якими найбільше ведеться спостереження, належать Лондон, Велика Британія, а також Тайюань і Вусі в Китаї. У США, однак, спостерігається тенденція відходу від використання технології розпізнавання облич та емоцій для забезпечення правопорядку. У деяких містах США (Сан-Франциско, Каліфорнія, Бостон, штат Массачусетс, Спрінгфілд, штат Массачусетс, Портленд, штат Орегон, і Портленд, штат Мен) взагалі заборонили поліції використовувати таке програмне забезпечення.[1]

І небезпідставно. Побачивши розшукувану особу на вулиці, ви можете допомогти зловити та затримати небезпечного злочинця. З іншого боку, додавання ідентифікації за емоціями може бути надзвичайно упередженим для багатьох культурних, етнічних та релігійних меншин.

1.1.6. Робототехніка

Найбільш яскравим прикладом того, наскільки просунутим може бути AI, є "робот-громадянин" Софія від HansonRobotics. Людиноподібні роботи задіяні у сфері обслуговування клієнтів: в одному з готелів Токіо є роботи-адміністратори, що зустрічають гостей. Роботизовані Аватарки, такі, як T-HR3 від Toyota, можуть повторювати рухи операторів-людей. Освітні роботи вміють читати емоції, дозволяють проводити навчання за індивідуальними програмами і можуть стати ефективним рішенням для інклюзивної освіти, як, наприклад, роботи Pepper і NAO від SoftBankRobotics. І таких прикладів справді безліч. [2]

Заключне слово

Незважаючи на свої спірні проблеми, технологія ідентифікації емоцій є багатомільярдною сферою, яка має багаторічну історію. Дедалі більше питань етичного характеру, пов'язаних з особистим спостереженням, з'являється в міру того, як ми прагнемо усунути всі упередження і вирішити

проблеми з ефективністю. Яким саме шляхом ми будемо йти з цією швидко прогресуючою технологією, стане зрозуміло лише з часом.

1.2. Аналіз існуючих програм

Доклавши максимум зусиль, я склала детальний перелік кожного доступного на сьогоднішній день програмного рішення для розпізнавання облич. На превеликий подив, активний розвиток більшості перспективних рішень з вільним доступом розпочався лише у 2020 році. Зважаючи на те, що платні продукти для підприємств не надають великої кількості відомостей для звичайних користувачів, провести їх ретельне дослідження мені було доволі проблематично.

Типи наявних на ринку рішень для розпізнавання облич

Передусім слід звернути увагу на великий вибір програмних продуктів для розпізнавання облич. Для роботи з деякими з них вам зовсім необов'язково мати навички роботи з алгоритмами машинного навчання, проте деякі з них потребують значно більше знань і зусиль.

На мою думку, я б виділила наступні три основні типи сервісів розпізнавання облич, кожен з них має переваги та недоліки:

1. Програмне забезпечення для розпізнавання облич як послуга (software as a service - SaaS). Постачальник послуг з розпізнавання облич бере на себе всі турботи по керуванню та технічній підтримці високонавантажених серверів, крім того, він не відстає за технологіями машинного навчання. Від вас вимагається лише підключити програму до своїх IT-систем за допомогою API-інтерфейсу. Такі рішення надають чимало можливостей, проте мають і чимало проблем. Насамперед, оскільки провайдер вирішує всі питання, даний сервіс є найдорожчим. Також потрібно мати стабільне з'єднання з Інтернетом, адже доведеться надсилати великі зображення на інтернет-сервер. Так як ви передаєте свої зображення

сторонній організації та ніяк не контролюєте, що саме відбувається з ними, то існують і певні побоювання з приводу безпеки.

2. Варіанти REST API, розміщені на власному хостингу. Ці технології мають як локальне, так і хмарне розміщення. Їм не загрожують ті ж проблемні моменти, притаманні SaaS-продуктам. Можна керувати тим, куди потрапляють дані, зберігаючи їх на власних серверах (або в приватній хмарі), і навіть можете спроектувати офлайн-рішення. Натомість, вам доведеться самостійно керувати серверним адмініструванням. Проте, сервери, як правило, працюють у вигляді контейнерів Docker, завдяки чому їх легко організувати. Незважаючи на те, що вони коштують не так багато, як їхні конкуренти SaaS, саморозміщені рішення, все-таки, досить дорогі. На щастя, зараз активно розробляються альтернативні варіанти саморозміщених REST API з відкритим вихідним кодом. Вони є доволі перспективними, не зважаючи на те, що не так розвинуті, як інші варіанти.

3. Фреймворки та бібліотеки з відкритим вихідним кодом. Часто вони є вільно доступними, адже чимало дослідників прагнуть поділитися вихідним кодом для розробки своїх передових методологій. Для використання цього типу програмного забезпечення вам, безсумнівно, знадобляться принаймні деякі знання з машинного навчання. Якщо ви хочете поєднати ці рішення з вашим власним програмним забезпеченням, вам також потрібно буде вкласти певні зусилля в його RESTification. Перевага полягає в тому, що ви отримаєте передове рішення, з яким ви добре знайомі.

Зрозуміло, наскільки важливим є розуміння своїх можливостей. Раніше функція ідентифікації облич для пересічних громадян коштувала надзвичайно дорого. Вартість послуг з розпізнавання облич для потокового відео може починатися від $86,40 в день і перевищувати $30,000 на камеру в рік. Швидше за все, саме тому технологія розпізнавання облич стає популярною лише тоді, коли її починають впроваджувати державні органи або провідні компанії.

Але оскільки з'являється все більше і більше вільного вибору, це вже не є такою великою проблемою, як це було раніше. Хоча вони знаходяться на різних стадіях розвитку, малі та середні підприємства вже можуть їх використовувати. Вони не тільки для нішевих любителів.

Я хочу пояснити, що точність була основним критерієм при виборі програмного забезпечення для розпізнавання облич для порівняння. Рішення можуть використовувати різноманітні бенчмарки, щоб продемонструвати свої високі стандарти. Щороку на ринку з'являються нові технології розпізнавання облич, а разом з ними і стандарти продуктивності. Тому порівнювати рішення навіть дворічної давності зі свіжими може бути складно. Проте існує один надзвичайно давній, але все ще широко використовуваний бенчмарк: Labeled Faces in the Wild (LFW). Я мала змогу оцінити правильність кожної відповіді з нашого списку, оскільки, на щастя, всі вони надали результати цього тесту.

Найкращі безкоштовні програмні рішення для розпізнавання облич у 2022 році:

1.Ageitgey/face_recognition (GitHub Repository)

Отримавши 45 тис. схвальних відгуків на GitHub, це, ймовірно, найбільш поширена бібліотека для безкоштовного розпізнавання облич. У вас є вибір між використанням їх API на Python або їх бінарної програми командного рядка для її використання. Всі основні платформи мають посібники з встановлення, і є навіть образ докера для більш швидкого встановлення. Незважаючи на те, що є поважні підстави вважати його популярним, ви повинні знати про деякі дуже суттєві мінуси, якщо бажаєте ним користуватися. Передусім, репозиторій все ще містить коміти, проте, судячи з усього, з моменту попереднього оновлення у 2018 році не відбулося жодних суттєвих змін (три роки у сфері ШІ - це великий термін). Крім того, алгоритм розпізнавання облич є дещо застарілим і дає лише 99,38% точності

на LFW. Зрештою, з цим додатком не так легко інтегруватися, адже у нього відсутній інтерфейс REST API.

2. CompreFace

Рішення опублікували тільки в липні 2020 року на GitHub, воно налічує всього лише приблизно 2000 зірок, однак здається доволі багатообіцяючим. У цьому переліку CompreFace - одне з декількох саморозміщених рішень для розпізнавання облич з REST API - його можливо запустити однією командою docker-compose. Завдяки тому, що програма використовує REST API, для її впровадження не вимагається кваліфікація інженера з машинного навчання - інтегрувати її у свою програму взагалі нескладно. Крім того, рішення є гнучким у масштабуванні, завдяки чому можна паралельно обробляти відразу кілька графічних потоків, будь це відео чи просто зображення на екрані. Також CompreFace пропонує зрозумілий користувацький інтерфейс для керування ролями учасників та коллекціями облич. Програма пропонує можливість вибору одного з двох методів розпізнавання облич: FaceNet (точність LFW 99,65%) та InsightFace (точність LFW 99,86%). Остання актуальна версія системи на момент написання диплому - 1.0.0.

3. DeepFace

Завдяки своїй схожій назві з алгоритмом розпізнавання обличчя DeepFace від Facebook, цей бібліотечний фреймворк, який був завантажений на GitHub у лютому 2020 року, набрав близько 4 100 оцінок. Крім того, ця платформа містить інші розпізнавальні алгоритми, такі як FaceNet та InsightFace. Додатково доступний REST API, але він дозволяє лише перевіряти методи, що унеможливлює формування колекцій облич та пошук серед них вашого лиця. Якщо ви володієте мовою Python, то почати роботу доволі просто, але для всіх інших - вкрай проблематично. Остання актуальна версія програми на момент написання диплому - 0.0.75.

4. FaceNet

Відкрита бібліотека Python, яка використовує FaceNet, була розроблена спеціалістами Google як технологія розпізнавання облич. Численні посібники використовують репозиторій, який налічує 12 600 зірок, як основну бібліотеку. Хоча ця методика є відносно застарілою, сучасні дослідники продовжують її застосовувати (востаннє - для розпізнавання облич у масках). Цей алгоритм демонструє доволі непогану ефективність (99,65% на базі даних LFW, тобто це, звісно, не жахливо, проте недостатньо для найкращого результату). До недоліків можна віднести нестачу REST API а також той факт, що сховище вже давно не оновлюється (востаннє його оновлювали аж у середині весни 2018 року).

5. InsightFace

Ще одним пакетом з публічним доступом на мові Python, який набрав 12,1 тис. оцінок, є InsightFace. Він працює на основі RetinaFace - однієї з найсучасніших та найнадійніших систем розпізнавання та ідентифікації осіб (SubCenter-ArcFace). Цей репозиторій є досить актуальним наразі. На наборі даних LFW даний алгоритм є правильним на 99,86%. Складність у застосуванні є його поки що єдиною вадою. Зверніть увагу на CompreFace і InsightFace-REST, якщо вам потрібні рішення, які працюють з InsightFace, мають практичніший REST API і здатні працювати з докерного контейнера.

6. InsightFace-REST

Хоча цей багатообіцяючий репозиторій був сформований ще у 2019 році, але розвиватися на постійній основі він почав лише у жовтні 2020 року. Це докерне програмне середовище, яке пропонує практичний REST API, подібний до CompreFace. Те, що творці втричі збільшили швидкість розпізнавання InsightFace, є його головною особливістю! Недоліком такого підходу є те, що ви повинні розробити свій особистий класифікатор, оскільки вони пропонують тільки вбудовування облич і не мають API для безпосереднього розпізнавання осіб. Також репозиторій досі не має власної ліцензії, тож потрібно буде звертатися до творця за дозволом на його

використання. Остання актуальна версія на момент написання диплому - v0.7.0.0.

Найкраще платне програмне забезпечення для розпізнавання облич у 2022 році (в алфавітному порядку)

1. Amazon Rekognition

У своїй версії SaaS Amazon Rekognition надає значний безкоштовний тестовий період тривалістю 12 місяців і 5 000 вільних розпізнавань щомісяця. Невизначеність оточує існування самостійної версії. Якщо ви виберете якусь із трьох запропонованих мов програмування, для яких вони пропонують SDK, то розпочати створення програми досить просто (Java, .Net та Python). Вартість стартує від 1 долара на кожні 1,000 розпізнавань і залежить від щомісячної кількості запитів. Крім того, Amazon пропонує безліч інших функцій, включаючи визначення статі та віку, виявлення орієнтирів та розпізнавання емоцій.

2. Deep Vision AI

Існують як SaaS-версії Deep Vision AI, так і версії, що розміщуються на власному хостингу. Однак, оскільки на веб-сайті немає даних про їх вартість, можна припустити, що вони співпрацюють виключно з бізнесом на корпоративному рівні. Також відсутня будь-яка інформація про мови програмування, для яких пропонуються SDK. На додаток до розпізнавання облич є сервіси для визначення статі та віку.

3. FaceFirst

Незважаючи на те, що ця система часто з'являється у переліках найліпших систем розпізнавання облич, даних про FaceFirst дуже мало. Імовірно, вони мають справу лише з бізнесом і пропонують лише самостійну хостингову версію. Також надається функція оцінки віку.

4. Face++

Нестандартно обмежена безкоштовна програма Face++ означає, що неможливо гарантувати, того що програма буде нормально працювати

протягом певного періоду часу. Компанія надає безліміт запитів, але їх можна робити лише три рази на секунду, і всі вони розподіляються порівну серед усіх користувачів, з безкоштовним тарифом. Крім того, ви повинні усвідомлювати, що платний тариф вдвічі дорожчий, ніж у інших лідерів - Amazon і Microsoft, які беруть по 1 долару за кожні 1 000 розпізнавань. Face++, однак, надає широкий спектр додаткових послуг, таких як - розпізнавання емоцій та статі, визначення орієнтирів та встановлення віку, а також підтримує більшість SDK, включаючи Python, PHP, Java, Javascript, C++, Ruby, iOS та Matlab. Крім того, вони надають як власні, так і SaaS-версії. Проте слід пам'ятати, що компанія Megvii, дочеринською компанією якої являється компанія Face++, з середини 2022 року знаходиться під дією санкцій уряду США за те, що її технологія була використана для порушення прав людини уйгурів в Сіньцзяні.

5. FaceX

FaceX - це нещодавній індійський бізнес, котрий пропонує SaaS-версії та версії для самостійного розміщення. Сформована у 2018 році,компанія немає безкоштовного тарифу, а єдиною ціновою опцією є щомісячне членство, яке коштує 15 доларів США на місяць і починається з 300 запитів щодня. SDK для мов програмування ніде не згадуються, хоча FaceX пропонує ряд інших послуг, включаючи ідентифікацію орієнтирів, визначення віку та визначення статі.

6. Kairos

Kairos має 14-денну безкоштовну ознайомчу версію, однак вона має обмеження на 10 000 запитів. Далі доведеться оплачувати обидві підписки, які починаються від $19 на місяць і $20 за кожні 1 000 розпізнавань. Якщо брати до уваги їхніх конкурентів, то це занадто дорого. З іншого боку, сервіс забезпечує SDK для мов PHP, JS, .Net і Python і крім цього інші послуги, як-от ідентифікація орієнтирів, визначення віку та гендерного розпізнавання. Також доступні версії як для самостійного розміщення, так і для SaaS.

7. Machine Box

Ліцензія на безоплатне використання від Machine Box надзвичайно інтригуючи - компанія пропонує самостійне рішення, яке є безоплатним, доки ви будете використовувати колекцію з не більше ніж 100 зображень. SDK для мови Go є ще однією перевагою, але ви можете отримати вигоду від цього, тільки якщо ви користуєтеся Go. Лише коли у вас є бізнес-ліцензія, він може бути масштабований. Будучи єдиним самостійним варіантом, який я змогла випробувати, я хочу відзначити, що почати роботу та інтегрувати його доволі легко. Здається, що безкоштовна версія є навіть більш базовою, ніж кілька масштабованих програм розпізнавання облич з вільним доступом, які можна використовувати в комерційних цілях.

8. Microsoft Azure Cognitive Services Face API

При ліміті в 20 запитів на хвилину, великодушний вільний план Microsoft дає змогу безкоштовно надсилати 30 000 запитів щомісяця. Але слід пам'ятати, що в цій ситуації немає SLA. Вартість платних запитів (з SLA) становить від 1 долара за 1 000 розпізнавань і базується на щомісячному обсязі розпізнавання. Підтримувані SDK включають for.Net, Python, Java, Node.js та Go. На додаток до багатьох інших послуг, включаючи ідентифікацію емоцій, розпізнавання орієнтирів, визначення віку та встановлення гендерної приналежності, а також Microsoft надає SaaS-версії та версії, розміщені на власному хостингу.

9. Paravision

Фірма з розпізнавання облич, націлена на бізнес і носить назву Paravision. Вони не вказують, чи пропонують вони додаткові послуги, і пропонують виключно варіанти, що розміщуються на власному хостингу. Paravision пропонує Python та C++ SDK. Крім того, немає інформації про тарифи, як це прийнято для подібних підприємств.

10. Trueface

Ще один стартап, що пропонує розпізнавання облич для бізнесу - Trueface. Компанія пропонує лише розміщені на власному хостингу системи розпізнавання облич з додатковими функціями, такими як визначення орієнтирів, встановлення віку та гендерної диференціації. Підтримуються Python та C++ SDK. Вартість Trueface аналогічно невідома.

Висновок

На сьогоднішній день сфера і надалі стрімко розвивається. Немає сенсу використовувати застарілі сервіси, які більше не підтримуються, оскільки алгоритми розпізнавання облич стають все більш точними. Нові технології з відкритим вихідним кодом зробили використання програмного забезпечення для розпізнавання облич більш дешевим. Існує багато варіантів використання цієї інноваційної технології, і навіть якщо порівнювати їх може бути складно, серед них обов'язково знайдеться той, який ідеально підійде саме вам.

1.3. Недоліки

1.3.1. Проблеми розпізнавання емоцій на зображеннях та відео

Нам часто буває складно ідентифікувати емоції людини, судячи тільки по виразу обличчя. Наукові праці стверджують, що різні люди визначають різні емоції дивлячись на один і той самий вираз обличчя. А для програм на основі штучного інтелекту це завдання навіть більш складне. Вчені і розробники з усього світу ведуть запеклі дискусії про те чи наявні на даний момент алгоритми розпізнавання емоцій є точними.

Є фактори, які ускладнюють ідентифікацію емоцій, їх можна поділити на дві категорії - технічного характеру і психологічного характеру.

Технічні:

Виявлення емоцій зіштовхується із низкою тих самих перешкод, що й розпізнавання предметів, що рухаються на відео, а саме : розпізнавання предметів, наявність факторів непрогнозованої поведінки і так далі. Зупинимося на основних технічних труднощах побудови ER-рішення та визначимо основні способи вирішення цих труднощів.

- Збільшення обсягу даних

ER-рішення, подібно всім алгоритмам машинного та глибинного навчання, передбачають наявність великого обсягу даних для тренувань. Серед них мають бути відеоматеріали різної кадрової частоти, відзняті з різноманітних точок зору, на різному тлі, з особами будь-якої національності, статі та етнічної приналежності, тощо.

Втім, більша частина доступних у відкритому доступі наборів даних є неповними. Зокрема, їм бракує розмаїття за кольором та статтю, вони маю

Є три варіанти вирішення даної проблеми:

1. Створити ваш особистий датасет. Такий спосіб є найбільш дорогим і трудомістким, проте внаслідок цього можна мати ідеальний датасет, необхідний для вирішення поставленої задачі.

2. Об'єднати різні датасети. Таким чином, можна випробувати ефективність розробленого вами програмного алгоритму на різних додаткових масивах даних.

3. Модифікуйте дані по ходу виконання завдання. Наприклад, дехто з науковців пропонує модифікувати фільми використані раніше, шляхом їх скорочення, внесення змін до умов освітленості, зменшення швидкості, збільшення швидкості, додаткового шуму, і т.д..

- Проблеми з візуальними перешкодами та освітленням обличчя

Перекриття внаслідок зміни положення тіла - це поширена перешкода для розпізнавання обличчя людини на кадрах відео, надто якщо мова йде про необроблені відеодані. Зазвичай тактикою боротьби з цією проблемою є фронталізація, завдяки якій відбувається розпізнавання ознак обличчя та вироблення опорних точок для тривимірного зображення.

Також досить типовими є змінені умови висвітлення та різниця в контрастності в умовах відсутності обмежень. Для покращення якості розпізнавання розробники, як правило, користуються методами корекції зовнішнього світла.

Є й деякі неортодоксальні методи. Так, частина дослідників радять вводити в інформаційний трафік інфрачервоний або близькоінфрачервоний прошарок. Такий прошарок малочутливий щодо впливу зміни рівня освітленості та дозволяє отримувати дані про коливання температури шкіри, які відображають емоції людини.

- Ідентифікація рис обличчя

Лиця скануються на елементи рис зовнішності за допомогою програми ідентифікації емоцій. Подібне сканування буває проблематичним з наступних причин:

- Діапазон розташування ознак зовнішності. Система "запам'ятовує" усереднену дистанцію між ознаками обличчя та здійснює їх пошук лише в межах цього діапазону. Приміром, у неї виникатимуть труднощі з розрізненням очей, розташованих на великій відстані один від одного.

- Розміри предмета. ER-рішення стикаються з труднощами при ідентифікації атипових ознак, наприклад, нетиповий розмір або колір губ, носа і тд.

- Тон шкіри. Через відтінок шкіряного покриву алгоритм розпізнавання в деяких ситуаціях неправильно визначає або невірно розпізнає певну рису обличчя.

Деякі розробники використовують підхід, заснований на компонентах, який розбиває ознаки зовнішності на багато секцій в залежності від анатомії лиця, задля покращення ефективності розпізнавання рис зовнішності. Після цього програма самостійно вносить компоненти до системи з належним маркуванням.

- Розпізнавання неповних емоцій

Переважна частина програм зосереджується саме на детекції піків експресії підвищеної емоційності активності, не звертаючи уваги на експресії меншої активності. Такий підхід часто спричиняє хибне розпізнавання

емоцій при дослідженні замкнутих осіб чи вихідців з держав із традиціями приховування емоцій.

Додаючи більше ознак у бази, програму можливо натренувати на розпізнавання первинних сигналів прояву емоцій. Задля цього, окрім міток емоцій та ознак зовнішності, необхідно додати до наборів ознак ознаки міміки, що характеризують глибину емоцій.

Запровадження поглибленої пікової нейронної мережі також вирішує цю проблему. Вона проводить аналіз пікових та позапікових виражень однакових емоцій і зводить до мінімуму частоту оновлення кадрів.

- Вловлювання контексту емоції

Фон та положення корпусу забезпечують стільки само відомостей стосовно реакції, скільки й лице. Однак, вилучення контексту є трудомісткою операцією, адже потребує вивчення цілого кадру, окрім лиця. Емоції, що передаються за допомогою рухів, також менш досліджені, порівняно з мімікою.

Контекстно-орієнтовані нейронні мережі (такі як CACA-RNN і CAER-Net) розбивають потокове відео на два прошарки: лице (чи кілька лиць) і його навколишнє середовище (поза людини, рухи, предмети на зображенні і т.д.). Власне сама технологія теж поділяється на дві частини, одна з яких відповідає за аналіз міміки лиця, а інша - за аналіз оточення. Остаточний підсумок включає в себе два виміри моделі.

- Розбіжності за расовою ознакою

Нам буває важко розпізнати почуття оточуючих, зокрема, представників відмінної від нас расової приналежності чи відмінних від нас традицій та звичаїв. Навіть найсучасніші технології штучного інтелекту мають суттєву перешкоду в цій сфері. Приміром, ще в 2015 році рішення по розпізнаванню GooglePhoto не могли відрізняти темношкірих осіб.

Проблематика криється насамперед як в розмаїтті інформації, так і в кадровому наповненні науково-дослідницьких колективів, що займаються розробкою. Більшість публічних баз даних надають багато інформації про білих чоловіків, але не про жінок або людей з іншим кольором шкіри. Тому що колективи, які проводять дослідження, зазвичай формуються білими чоловіками, які часто перевіряють результати досліджень на власних прикладах.

Варіантом вирішення даної ситуації могла б стати більш ретельна перевірка масивів інформації напередодні тестування. Або ж формувати різнорідніші колективи.

Психологічні:

Ще в 1950-х роках вчені-психологи вивчали питання взаємодії вираження обличчя та прояву емоцій. Однак, є низка недоопрацювань. Оскільки ці дослідження слугують основою для алгоритмів розпізнавання емоцій, дуже необхідно знати про існуючі проблеми.

- Культурні відмінності в вираженні емоцій

Незважаючи на те, що в всі ми відчуваємо одні й ті ж самі почуття, проявляють їх всі неоднаково. Представники західних народів висловлюють свою емоцію окремим набором міміки та жестів, які відповідають кожній із семи базових емоцій. Японці та китайці, навпаки, схильні бути стриманішими, виражаючи свої почуття лише за допомогою унікальних рухів очей.

Такий фактор не можна не враховувати перед відбором вибірки для тестування. Слід користуватися базою даних з людьми, що належать до такої ж культури, як і особи, емоції яких ви хочете ідентифікувати, або тестувати алгоритм на багатьох БД.

- Виявлення емоцій дітей

Грудні малюки та дошкільнята виражають почуття не так, як це роблять повнолітні дорослі. Діти більше усвідомлюють, аніж здатні передати голосом, та відповідають різноманітним виразом обличчя на те, що відбувається довкола них. Крім того, молоді люди не контролюють власні реакції.

Однак дослідникам не вдається встановити, чи ці мімічні жести відображають емоції у новонароджених і дітей дошкільного віку. Ще одна складність виникає через те, як дорослі інтерпретують емоції дітей. Перші досліди емоцій немовлят, наприклад, припускали, що вони здатні демонструвати злість вже у семимісячному віці. Фактично діти всього лише супилися, і достовірно асоціювати такий погляд з певною емоцією в настільки ранніх вікових періодах не можна.

- Невірні індикатори емоцій

Незважаючи на те, що ознаки семи основних емоцій були широко досліджені, далеко не усі наукові праці погоджуються з тим, про що саме свідчать ці ознаки. Чимало психологів, приміром, переконані, що насуплений лоб - це ознака злості. Згідно з попередніми дослідженнями, всього 30% осіб супляться, будучи розгніваними. Отже, можна вважати, що 70% проявляють свою злість інакше, і хмуриться для того, щоб передати інші емоції.

Проблемність процесу ідентифікації людських емоцій завдяки штучному інтелекту - це проблема виключно психологічна, а не програмістська. Використовуючи невелику кількість ознак, можна отримати помилкові спрацьовування і погано натренований штучний інтелект. Отже, слідкуйте за появою нових наукових розробок у сфері вираження емоцій та регулярно тренуйте своє рішення на основі аналізу отриманих матеріалів.

1.3.2. Противники систем розпізнавання обличчя

В сьогоднішній час технології ідентифікації лиця є дуже поширеними серед багатьох сфер нашого життя. Нам рекламують їх, як засоби захисту вашого мобільного девайсу чи просто вхідної двері від пересічних людей.

Користуються нею і держструктури, такі як департаменти поліції. За даними Джорджтаунського університету поліцейські бази налічують в собі лиця більшої частини повнолітніх жителів Америки. Технологію ідентифікації осіб застосовують державні органи по всьому світу для виявлення незгодних і стеження за ними, а поліцейські Гонконгу застосовують цю технологію щоб переслідувати учасників акцій протесту.

Як одна з форм опору, небайдужі до приватного життя дизайнери, вчені та активні діячі розробляють одежу і засоби, які можна носити, щоб обдурити технологію розпізнавання облич. Для збереження своєї анонімності учасники демонстрацій в різних країнах від Гонконгу до США використовували маски, а служба зашифрованих повідомлень Signal навіть почала безкоштовно роздавати маски проти розпізнавання облич учасникам акцій протесту Джорджа Флойда.

ШІ застосовується програмами розпізнавання облич для швидкого розпізнавання облич чи будь-яких людських фігур. Проте програму легко ввести в оману шляхом використання оманливих форм одягу, який перешкоджає ШІ розібратися в тому, на що він дивиться. Інші методи перешкоджають ШІ правильно ідентифікувати людину, заплутуючи його фотографіями підставних осіб.

Такі досі нішеві розробки в основному були використані лише в наукових або мистецьких інсталяціях. Проте в міру поширення технології розпізнавання облич вони можуть перетворитися на новітню розробку в галузі натільних технологій.

Слід визнати, що технології розпізнавання облич не є надійними, і вже створюються певні алгоритми для їх обходу.

Рис. 1.1. Винахід Джипа ван Ліуванштейна

Джип ван Ліуванштейн (студент Утрехтської школи мистецтв в Нідерландах) розробив маску в формі лінзи (Рис. 1.1), яка не дає алгоритмам розпізнавати обличчя, але дозволяє людям впізнавати співрозмовника і розуміти, які емоції він виражає. Вигнута форма маски блокує механізми розпізнавання з усіх кутів.[3]

Рис. 1.2. Винахід Цзин-кай Лю

Інший студент з Нідерландів, Цзин-кай Лю, зробив проєктор (Рис. 1.2), який накладає на вас зображення обличчя іншої людини. Пристрій постійно перемикається між обличчями, через що розпізнати ваше справжнє обличчя стає ще важче.[3]

(a) Near infrared LED not lit (detection successful)

(b) Near infrared LED lit (detection failed)

Рис. 1.3. «Візор конфіденційності» ІсаоЕчізен

ІсаоЕчізен, професор Національного інституту інформатики в Токіо, розробив «візор конфіденційності» (Рис. 1.3), пристрій, що захищає від камер спостереження, які можуть зчитувати обличчя людей без дозволу. [3]

На фотографіях з лабораторії Ечізена видно, як візор заважає штучному інтелекту розпізнати обличчя. У пристрої є лампа, що випускає майже інфрачервоне випромінювання, яке створює візуальний шум на зображеннях камер, але при цьому не заважає звичайним людям сприймати обличчя.[3]

Рис. 1.4. CV Dazzle Адам Харві

Художник Адам Харві вирішив використовувати макіяж для захисту від алгоритмів, які вміють розпізнавати обличчя. Він розробив особливу техніку макіяжу під назвою CV Dazzle (Рис. 1.4) – він являє собою комбінацію макіяжу, шиньйонів, аксесуарів і каменів, які дозволяють трансформувати обличчя.

Техніка бере початок від прийому, який використовувався під час Першої світової війни — тоді кораблі фарбували в чорно-білі смуги, через що здалеку було важче зрозуміти їх розмір і в який бік вони спрямовані.[3]

Рис. 1.5. Шарф Санні Уікерс

Санні Уікерс, студентка дизайну з Нідерландів, придумала шарф на якому зображені обличчя (Рис. 1.5), через що збиває з пантелику алгоритми.

«Якщо дати програмі занадто багато інформації, вона заплутається і ви станете для неї невидимими», — вважає Санні Уікерс.[3]

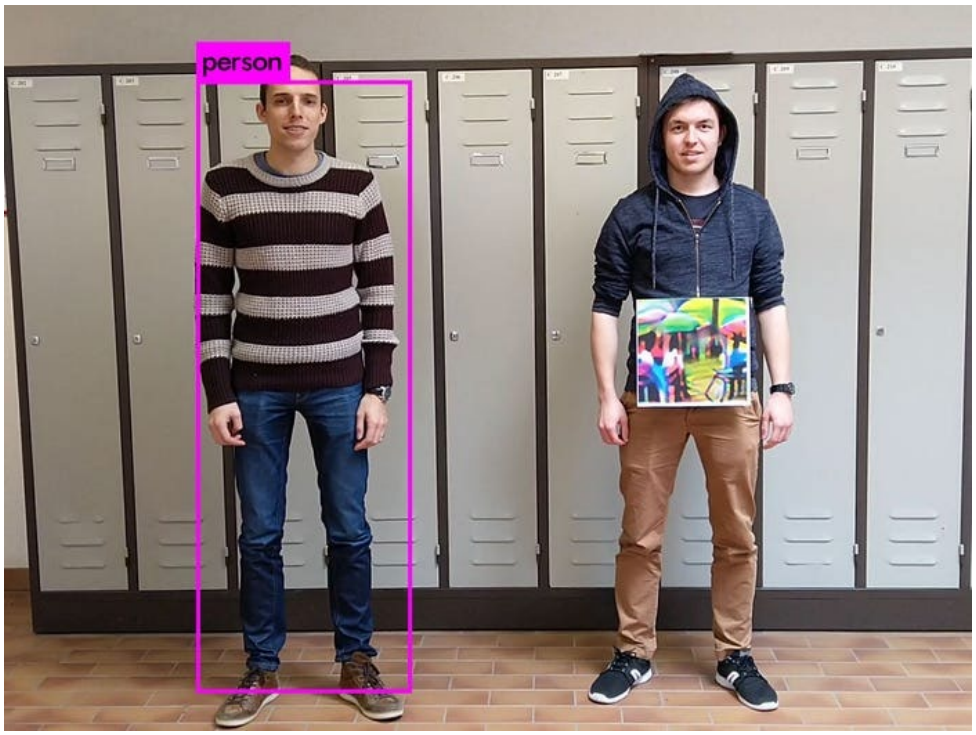

Рис. 1.6. Розробка дослідників Сімен Тіс, Віб ван Ранст і Тун Гедеме

Бельгійські дослідники Сімен Тіс, Віб ван Ранст і Тун Гедеме розробили спеціальний принт на одяг (Рис.1.6), який заважає алгоритмам виявляти обличчя. [3]

Рис. 1.7. Маска Зака Бласа

Художник Зак Блас зробив маску, зібрану з облич різних людей і тому має невизначену форму (Рис. 1.7). Своїм проєктом автор хоче звернути увагу публіки на проблему збору біометричних даних і забобонів, які діють в цій сфері.[3]

РОЗДІЛ 2. НЕЙРОННА МЕРЕЖА

2.1. Поняття нейронної мережі

Під нейронними мережами маються на увазі обчислювальні структури, які моделюють прості біологічні процеси, зазвичай асоційовані з процесами людського мозку. Вони являють собою розподілені та паралельні системи, здатні до адаптивного навчання шляхом аналізу позитивних і негативних впливів. Елементарним перетворювачем у цих мережах є штучний нейрон або просто нейрон, названий так за аналогією з біологічним прототипом. [4]

2.2. Структура штучної нейронної мережі

Штучні НМ є групуванням штучних нейронів, у вигляді з'єднаних між собою шарів. Більшість реалізацій використовують мережі, що містять щонайменше три типи прошарків – вхідний, прихований та вихідний (Рис. 2.1).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. [5]

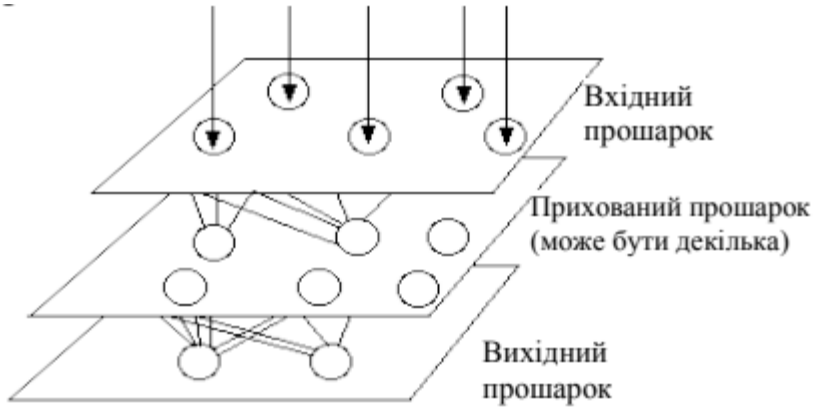

Рис. 2.1. Схема штучної НМ

Між двома попередніми прошарками може бути декілька прихованих прошарків, що містять багато різноманітно зв’язаних нейронів. Входи та виходи кожного з прихованих нейронів з’єднані з іншими нейронами. Напрямок зв’язку від одного нейрона до іншого є важливим аспектом нейромереж. У більшості мереж кожен нейрон прихованого прошарку отримує сигнали від всіх нейронів попереднього прошарку і зазвичай від нейронів вхідного прошарку.[5]

Після виконання операцій над сигналами нейрон передає свій вихід всім нейронам наступних прошарків, забезпечуючи передачу сигналу вперед (feedforward) на вихід. При зворотному зв’язку вихід нейронів прошарку скеровується до нейронів попереднього прошарку (рис. 2.2). [5]

Рис. 2.2. Схема штучної НМ зі зворотним зв’язком

2.3. Навчання штучної нейронної мережі

Нейромережа налаштовує ваги зв’язків за наявною навчальною множиною. Робота мережі розділяється на навчання та адаптацію. Під навчанням розуміється процес адаптації мережі до пропонованих еталонних зразків шляхом модифікації (відповідно до тих чи інших алгоритмів) вагових коефіцієнтів зв’язків між нейронами. Для процесу навчання необхідно мати модель зовнішнього середовища, у якій функціонує НМ, потрібну для

вирішення задачі інформацію. По-друге, необхідно визначити, як модифікувати вагові параметри мережі.[5]

Існують три види навчання НМ [5]:

− «із учителем», коли НМ має в своєму розпорядженні правильні відповіді (виходи мережі) на кожен вхідний приклад. Ваги налаштовуються так, щоб мережа виробляла відповіді близькі до відомих правильних відповідей;

− «без учителя» (самонавчання) не вимагає знання правильних відповідей на кожний приклад навчальної вибірки. Використовується внутрішня структура даних та кореляція між зразками в навчальній множині для розподілу зразків за категоріями;

− при змішаному навчанні частина вагів визначається за допомогою навчання з учителем, тоді як інша визначається за допомогою самонавчання.

Розрізняють також контрольоване та неконтрольоване навчання НМ. При контрольованому навчанні НМ змінний вихід постійно порівнюється з бажаним виходом. Ваги на початку встановлюються випадково, але під час наступних ітерацій коректуються для досягнення близької відповідності між бажаним та змінним виходом.[5]

Перед використанням НМ з контрольованим навчанням повинна бути навченою. Навчальні множини повинні бути досить великими, щоб містити всю необхідну інформацію для виявлення важливих особливостей і зв’язків. Але і навчальні приклади повинні містити широке розмаїття даних. Фаза навчання може тривати досить довго. Навчання вважається закінченим при досягненні НМ визначеного користувачем рівня ефективності. Цей рівень означає, що мережа досягла бажаної статистичної точності, оскільки вона видає бажані виходи для заданої послідовності входів.Після навчання ваги з’єднань фіксуються для подальшого застосування. [5]

Якщо після контрольованого навчання НМ ефективно опрацьовує дані навчальної множини, важливим стає її ефективність при роботі з даними, які

не використовувалися для навчання. У випадку отримання незадовільних результатів для тестової множини навчання продовжується. Тестування використовується для забезпечення запам’ятовування не лише даних заданої навчальної множини, але і створення загальних образів, що можуть міститися в даних.[5]

Неконтрольоване навчання передбачає, що комп’ютери можуть самонавчатись у справжньому роботизованому сенсі. На сьогодні неконтрольоване навчання використовується в мережах відомих, як самоорганізовані карти (self organizing maps), що перебувають у досить обмеженому користуванні, але доводять перспективність даного виду навчання.[5]

Алгоритм неконтрольованого навчання скерований на визначення близькості між групами нейронів, які працюють разом. Якщо зовнішній сигнал активує будь-який вузол у групі нейронів, дія всієї групи в цілому збільшується. Аналогічно, якщо зовнішній сигнал у групі зменшується, це приводить до гальмуючого ефекту на всю групу.[5]

Конкуренція між нейронами формує основу для навчання. Навчання конкуруючих нейронів підсилює відгуки певних груп на певні сигнали. Це пов’язує групи між собою та відгуком. При конкуренції змінюються ваги лише нейрона-переможця.[5]

2.4. Глибинне навчання

Глибинне навчання (deep learning) – це розділ машинного навчання, спрямований на побудову ієрархічних моделей шляхом використання високорівневих масових абстракцій даних на основі глибинного графу з багатьма обробними шарами, які здійснюють лінійні або нелінійні перетворення, тобто це – прогресуючий високорівневий витяг ознак з сирих (необроблених) первісних вхідних даних. У центрі глибинного навчання є глибинні нейронні мережі та методи їхньої побудови. [6]

Глибинна нейронна мережа (ГНМ, deep neural network, DNN) – це різновид ШНМ, що має багато шарів обробки даних, яка перетворює вхідні дані у вихідні, ієрархічно виділяючи та агрегуючи ознаки, підвищуючи рівень абстракції даних в напрямку від входів до виходів. [6]

Порівняно з мілкими НМ, ГНМ за рахунок збільшення кількості нейроелементів та зв'язків отримують більшу обчислювальну потужність і здатність моделювати більш складні залежності, а за рахунок спеціалізації шарів та високої ієрархічності обробки даних стають більш зручними для сприйняття та аналізу людиною. При цьому спеціалізація шарів обробки даних у ГНМ робить їх більш пристосованими до інтеграцію в мережеву модель апріорної інформації про предметну область. Проте, як правило, конкретні парадигми ГНМ мають більш обмежене застосування у конкретних задачах (наприклад, деякі архітектури можуть застосовуватися лише для розпізнавання зображень і не придатні для інших задач). [6]

2.5. Згорткова нейронна мережа

Згорткова нейронна мережа (CNN) - це алгоритм глибокого навчання, який може приймати вхідне зображення, призначати важливість (навчальні ваги та упередження) різним аспектам/об'єктам на зображенні та бути здатним диференціювати один від іншого. Попередня обробка, необхідна в ConvNet, набагато нижча порівняно з іншими алгоритмами класифікації. В той час як в примітивних методах фільтри створюються вручну, при достатньому навчанні ConvNet має можливість вивчати ці фільтри/характеристики. [7]

Архітектура CNN була натхненна структурою зорової кори головного мозку людини, вона аналогічна структурі з'єднання нейронів у людському мозку. Деякі нейрони реагують на сигнали виключно в обмеженій ділянці поля зору, котру називають "рецептивним полем" (Receptive Field). Скупчення таких полів накладаються один на одного, щоб охопити всю зорову область.

2.6. Архітектура згорткової нейронної мережі

2.6.1. Базова архітектура

Архітектура CNN складається з двох основних частин (Рис. 2.3)

- Інструмент згортки, призначений для відокремлення та ідентифікації різних особливостей зображень для проведення аналізу в процесі, який має назву "Виділення особливостей" (Feature Extraction).
- Система виділення характеристик складається з багатьох пар шарів згортки чи об'єднання шарів.
- Повністю з'єднаний шар, який обробляє вихідні дані процесу згортки та за допомогою ознак, витягнутих на попередніх етапах, прогнозує клас зображення.
- Така модель виділення ознак CNN націлена на зменшення числа характеристик, представлених у наборі даних. Створюються нові ознаки, які узагальнюють існуючі, зібрані у вихідному наборі ознак. Існує велика кількість рівнів CNN, як це показано на архітектурній схемі CNN.

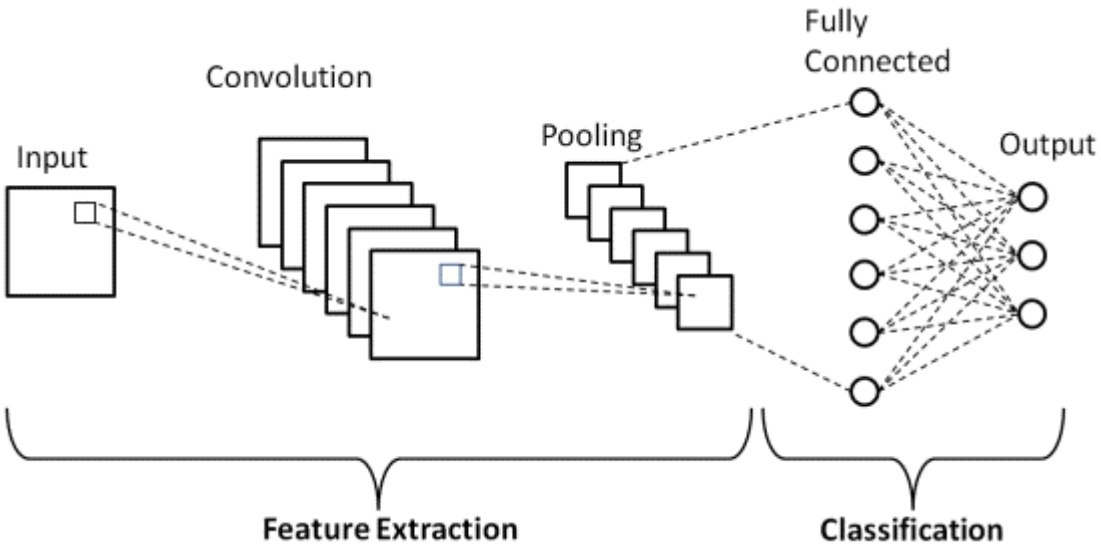

Рис. 2.3. Базова архітектура згорткової нейронної мережі

Існує три типи шарів, які складають CNN: шари згортки, об'єднані шари та повнозв'язні шари (FC). Коли ці шари складаються, формується

архітектура CNN. На додаток до цих трьох шарів, є ще два важливих параметри, які є шаром відсіву та функцією активації, які визначені нижче. [8]

1. Згортковий шар (Convolutional Layer)

Даний шар є першим шаром, який використовується для виділення різних ознак з вхідних зображень. У цьому шарі виконується математична операція згортки між вхідним зображенням і фільтром певного розміру MxM. Шляхом переміщення фільтра по вхідному зображенню береться точковий добуток між фільтром і частинами вхідного зображення з урахуванням розміру фільтра (MxM).

Результат роботи системи є мапою особливостей. Ця мапа містить відомості про зображення, такі як кути та ребра. Після цього карту подають на наступні шари, з метою вивчення ще ряду інших ознак вхідного зображення.

Після застосування операції згортки на вхідному зображенні згортковий шар у CNN передає отриманий результат до наступного шару. Переваги шарів згортки в CNN полягають у тому, що вони забезпечують цілісність просторового зв'язку між пікселями.

2. Об'єднувальний шар (Pooling Layer)

Зазвичай після шару згортки слідує шар об'єднання. Основною метою цього шару є зменшення розміру згорнутої карти особливостей для зменшення обчислювальних витрат. Це виконується шляхом зменшення зв'язків між шарами і незалежно працює над кожною картою особливостей. Залежно від методу, що використовується, існує декілька типів операцій об'єднання. В основному вона підсумовує особливості, згенеровані шаром згортки. [8]

У полі максимального об'єднання з карти об'єктів береться найбільший елемент. У режимі Average Pooling здійснюється розрахунок середнього значення елементів для заданого розміру секції зображення. У режимі Sum

Pooling розраховується загальна сума всіх елементів у заданій ділянці. Шар об'єднання зазвичай виконує роль сполучної ланки між шаром згортки та повнозв’язним шаром.

Дана модель CNN підсумовує характеристики, отримані в шарі згортки, допомагаючи мережам самостійно ідентифікувати ці характеристики. Крім того, завдяки цьому обчислення в мережі також зменшуються.

3. Повнозв'язний шар (Fully Connected Layer)

Повнозв'язний шар (FC) складається з ваг і зсувів разом з нейронами і використовується для з'єднання нейронів між двома різними шарами. Ці шари зазвичай розміщуються перед вихідним шаром і утворюють останні кілька шарів архітектури ШНМ. [8]

Вхідні зображення, при цьому, з попередніх шарів стають згладженими і подаються на шар FC. Після цього згладжений вектор проходить через ще декілька шарів FC, на яких, як правило, виконуються операції з використанням математичних функцій. Саме на цьому етапі розпочинається процес класифікації. Через те, що два пов'язаних шари працюють краще, ніж один пов'язаний шар, вони з'єднуються між собою. Завдяки цим шарам в CNN зменшується людський нагляд

4. Виключення (Dropout)

Коли всі ознаки підключаються до шару FC, зазвичай, це призводить до перенавчання на навчальній вибірці даних. Це трапляється тоді, коли окрема модель працює настільки добре на навчальних даних, що погіршує продуктивність моделі при застосуванні її на нових даних.

Саме для подолання такої проблеми застосовують шар відсіву, тобто кілька нейронів вилучаються з нейронної мережі в процесі навчання, що призводить до зменшення розміру моделі. Після проходження рівня відсіву 0,3, 30% вузлів випадковим чином вилучаються з нейронної мережі.

Відсівання дозволяє підвищити ефективність моделі машинного навчання, перешкоджаючи надмірному пристосуванню, роблячи мережу більш простою. Під час навчання з нейронної мережі випадають нейрони.

5. Функції активації

Останнім, але не менш важливим параметром моделі CNN є активаційні функції. Вони використовуються для вивчення та апроксимації будь-якого виду безперервних і складних зв'язків між змінними мережі. Простими словами, вона вирішує, яка інформація моделі повинна вистрілити в прямому напрямку, а яка ні в кінці мережі. [8]

Це додає мережі нелінійності. Існує декілька загальновживаних функцій активації, таких як ReLU, Softmax, tanH та Sigmoid. Кожна з цих функцій має специфічне застосування. Для двійкової класифікації за моделлю CNN перевага надається сигмоїдний та софтмаксній функціям, а для багатокласової класифікації, як правило, використовується софтмакс. Простіше кажучи, функції активації в моделі CNN визначають, чи повинен нейрон бути активований чи ні. Він вирішує, чи важливий вхідний сигнал для роботи, чи ні, щоб передбачити його за допомогою математичних операцій. [8]

2.6.2. Архітектура LeNet

Для проекту розроблена згорткова нейронна мережа на основі архітектури LeNet, з внесенням деяких змін.

У 1998 році архітектура LeNet-5 була представлена в дослідницькій роботі під назвою "Навчання на основі градієнта, застосоване до розпізнавання документів" Яна Лекуна, Леона Ботту, Йошуа Бенгіо та Патріка Хаффнера. Це одна з найбільш ранніх і базових архітектур CNN. [8]

Вона складається з 7 шарів (Рис.2.4). Перший шар складається з вхідного зображення розміром 32×32. Воно згортається за допомогою 6 фільтрів розміром 5x5, що призводить до розміру 28x28x6. Другий шар - це

операція об'єднання, яка фільтрує зображення розміром 2x2 і кроком 2. Таким чином, розмір результуючого зображення буде 14x14x6. [8]

Аналогічно, третій шар також включає в себе операцію згортки з 16 фільтрами розміром 5x5, а потім четвертий шар об'єднання з аналогічним розміром фільтрів 2x2 і другим кроком. Таким чином, результуючий розмір зображення буде зменшено до 5x5x16. [8]

Після того, як розмір зображення зменшено, п'ятий шар є повністю з'єднаним згортковим шаром з 120 фільтрами, кожен з яких має розмір 5×5. У цьому шарі кожна з 120 одиниць цього шару буде з'єднана з 400 (5x5x16) одиницями з попередніх шарів. Шостий шар також є повністю з'єднаним шаром з 84 блоками. [8]

Останній сьомий шар буде вихідним шаром softmax з "n" можливими класами залежно від кількості класів у наборі даних. [8]

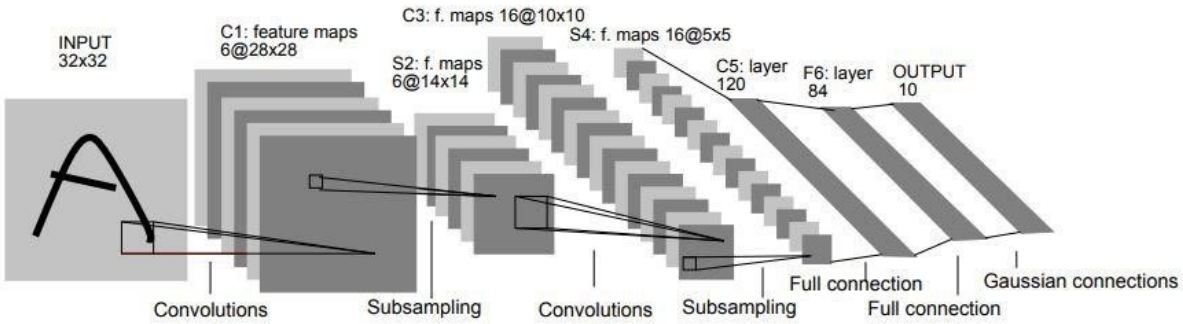

Рис. 2.4. LeNet-5 CNN архітектура

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ

3.1. Мова програмування

Для розробки застосунку, я обрала мову програмування Python, оскільки вона має велику кількість переваг.

Python - це мова програмування, яка спрямована на те, щоб як початківці, так і досвідчені програмісти могли легко перетворювати ідеї в код. Серед сучасних мов програмування в галузі машинного навчання вона є найбільш поширеною, зрілою та добре підтримуваною, тому чимало розробників зазначають у резюме саме Python.

Реалізація комп'ютерного зору мовою Python дозволяє розробникам автоматизувати процеси, які пов'язані з візуалізацією. Незважаючи на те, що інші мови програмування також підтримують комп'ютерний зір, Python впевнено перемагає у конкурентній боротьбі.

Переваги використання Python для комп'ютерного зору:

- Проста у написанні коду

Основна мета Python зробити програмування максимально простим для користувача. Завдяки цьому програмісти мають змогу більше приділяти уваги дизайну, а не кодуванню. Це ідеально для тих, хто тільки починає працювати з машинним навчанням та основами програмування.

- Обширні бібліотеки для машинного навчання

Python активно використовується для машинного навчання. Аналітики даних віддають багато часу цій мові, тому що нею легко писати, і вона безкоштовна. CV-розробникам можна не хвилюватися про проекти, над якими вони працюють, тому що більшість кейсів вже включені в бібліотеки Python.

- Відкритий вихідний код

На відміну від MATLAB (який теж орієнтований на аналіз даних, візуалізацію тощо), Python є безкоштовним. Не варто говорити, про те, що для Python все, що вам потрібно - це лише наявність комп'ютера, і можна починати роботу.

- Можливість прямої інтеграції з веб-фреймворками

Мова Python має готові веб-фреймворки, наприклад, Django. Його метою є швидка розробка, охайний та реалістичний дизайн. Окрім того, Python включає в себе мікро-фреймворки, які є такими ж функціональними, як і більш масштабні аналоги.

- Широко використовуваний

Завдяки цьому вона має більшу спільноту. На тему Python + OpenCV є безліч публікацій в блогах і ресурсах в Інтернеті, тому ви завжди можете знайти допомогу у вирішенні будь-якої проблеми.

3.2. Робота з пакетами Python

3.2.1. Індексатор пакетів Python

Python має досить обширну базову бібліотеку, яку можна відразу ж використовувати. Але у випадку, якщо вам потрібен пакет, якого немає в основній бібліотеці, ви можете знайти його в каталозі пакетів Python Package Index.

За бажанням автора пакунка, Python Package Index (PyPI) також включає дані пакунка, такі як дистрибутивні файли для дистрибутивів, упакованих за допомогою distutils. Команди register та upload з Distutils використовуються для розміщення файлів дистрибутивів та метаданих до PyPI відповідно. [9]

3.2.2. Система управління пакетами

Pip - це менеджер пакетів Python за замовчуванням. За допомогою нього здійснюється встановлення та керування пакетами, що не належать до складу стандартної бібліотеки Python.

Pip є одним із основних інсталяторів Python, починаючи з версій 3.4 та 2.7.9 для Python 3 та Python 2, відповідно. Pip - це інструмент, який повинен мати кожен пітоніст, оскільки він використовується у багатьох проектах на Python.

Для тих, хто працює з іншими мовами програмування, вам повинна бути близька концепція пакетного менеджеру. Ruby використовує gem, JavaScript використовує npm, а the.NET використовує NuGet. Pip зараз є фактичним менеджером пакетів для Python.

Мова Python розглядається як така, що має батарейки в комплекті. Це пояснюється тим, що в стандартну бібліотеку Python включено широкий вибір пакетів і модулів, призначених для підтримки розробників у їхніх зусиллях з програмування.

Крім того, Python має активну спільноту, що доповнює бібліотеку пакетами, здатними відповідати вимогам вашої розробки. Доступ до них здійснюється завдяки PyPI, індексу пакетів Python.

Багато пакетів, зокрема фреймворки для розробки, інструменти та бібліотеки, розміщені на PyPI. Більшість з таких пакетів надають зручні інтерфейси для реалізації функціональних можливостей стандартної бібліотеки Python. Звісно ж ви можете інсталювати або ж, навпаки, видалати їх всіх скориставшись системой управління пакетами pip, за допомогою наступної команди: pip install назва пакета/бібліотеки/модуля або pip unstall.

3.3. Бібліотеки використані для реалізації проєкту

3.3.1. OpenCV

Найпопулярніший і, ймовірно, найпростіший спосіб виявлення облич за допомогою Python — це використання пакету OpenCV.

OpenCV (Open Source Computer Vision Library) - бібліотека програмного забезпечення з відкритим вихідним кодом для комп'ютерного зору та машинного навчання. OpenCV була створена для забезпечення загальної інфраструктури для додатків комп'ютерного зору та прискорення використання машинного сприйняття в

комерційних продуктах. Будучи ліцензованим продуктом Apache 2, OpenCV дозволяє компаніям легко використовувати та модифікувати код. [10]

Вона містить близько 2500 алгоритмів, які містять широкий спектр як традиційних, так і новітніх методів комп'ютерного зору та машинного навчання. За допомогою цих алгоритмів здійснюється виявлення аналогічних знімків серед наявних у БД візуальних даних, усунення ефекту почервоніння очей на фотографіях, виконаних зі спалахом, моніторинг переміщення погляду, зчитування ландшафту, а також накладання міток для встановлення на нього доповненої реальності. Вони дозволяють ідентифікувати обличчя, виявляти об'єкти, проводити аналіз відеозаписів, відслідковувати переміщення відеокамери, відслідковувати об'єкти, що рухаються (Рис. 3.1), витягувати тривимірні моделі предметів, добувати тривимірні точкові хмари зі стереокамер, склеювати знімки щоб створити знімок високої якості. Понад 47 тис. осіб використовують OpenCV, а завантажили його, за оцінками, понад 18 млн разів. Пакет інтенсивно використовується.

Рис. 3.1. Система ідентифікує пішеходів

Функціональність OpenCV [11]:

- Введення/виведення зображення/відео, обробка, відображення (core, imgproc, highgui)
- Виявлення об'єктів/особливостей (objdetect, features2d, nonfree)
- Монокулярний або стерео комп'ютерний зір на основі геометрії (calib3d, stitching, videostab)
- Обчислювальна фотографія (фото, відео, superres)
- Машинне навчання та кластеризація (ml, flann)
- CUDA прискорення (gpu)

Поряд з такими відомими компаніями, як Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, які використовують бібліотеку, є багато стартапів, таких як Applied Minds, VideoSurf та Zeitera, які широко використовують OpenCV. Застосування OpenCV охоплює діапазон від зшивання зображень вуличного огляду разом, виявлення вторгнень у відеоспостереження в Ізраїлі, моніторингу шахтного обладнання в Китаї, допомоги роботам у навігації і підборі об'єктів у Willow Garage, виявлення нещасних випадків утоплення в басейні в Європі, запуску інтерактивного мистецтва в Іспанії і Нью-Йорку, перевірки злітно-посадочних смуг на наявність сміття в Туреччині, перевірки етикеток на продуктах на фабриках по всьому світу до швидкого розпізнавання облич в Японії. [10]

Він має інтерфейси C++, Python, Java та MATLAB і підтримує Windows, Linux, Android та Mac OS. OpenCV здебільшого орієнтований на додатки технічного зору в реальному часі та використовує переваги інструкцій MMX та SSE, коли вони доступні. Зараз активно розробляються повнофункціональні інтерфейси CUDA та OpenCL. Існує понад 500 алгоритмів і приблизно в 10 разів більше функцій, які складають або підтримують ці алгоритми. OpenCV написаний нативно на C++ і має шаблонний інтерфейс, який без проблем працює з контейнерами STL. [10]

3.3.2. NumPy

NumPy (Numerical Python) є фундаментальним пакетом для наукових обчислень на мові Python. Це бібліотека мови Python, яка надає об'єктам багатовимірного масиву, різні похідні об'єкти (такі як замасковані масиви і матриці),

а також набір процедур для швидких операцій над масивами, включаючи математичні, логічні, маніпуляції з фігурами, сортування, вибірку, введення/виведення, дискретні перетворення Фур'є, основи лінійної алгебри, базові статистичні операції, випадкове моделювання і багато іншого. [12] NumPy була створена в 2005 році Тревісом Оліфантом. Це проект з відкритим вихідним кодом, який знаходиться у вільному доступі. Серед користувачів NumPy є багато різних людей, від початківців кодувальників до досвідчених дослідників, які займаються сучасними науковими та промисловими дослідженнями та розробками.

У мові Python є списки, які виконують функцію масивів, але вони повільно обробляються. NumPy має на меті надати об'єкт масиву, який працює до 50 разів швидше, ніж традиційні списки Python. Об'єкт масиву в NumPy називається ndarray, він надає багато допоміжних функцій, які роблять роботу з ndarray дуже простою. [13]

Бібліотека NumPy містить багатовимірні масивні та матричні структури даних. Вона надає ndarray, однорідний об'єкт n-вимірного масиву, з методами для ефективної роботи з ним. NumPy можна використовувати для виконання широкого спектру математичних операцій над масивами. Він додає до мови Python потужні структури даних, які гарантують ефективні обчислення з масивами та матрицями, а також надає величезну бібліотеку високорівневих математичних функцій, які оперують з цими масивами та матрицями. [14]

Існує декілька важливих відмінностей між масивами NumPy та стандартними послідовностями Python [12] :

- Масиви NumPy мають фіксований розмір при створенні, на відміну від списків Python (які можуть динамічно зростати). Зміна розміру масиву призведе до створення нового масиву та видалення початкового.
- Елементи в масиві NumPy обов'язково повинні бути одного типу даних, а отже, мати однаковий розмір в пам'яті. Виняток: можна мати масиви об'єктів (Python, в тому числі NumPy), що дозволяє створювати масиви з елементів різного розміру.

- Масиви NumPy полегшують проведення складних математичних та інших видів операцій над великими масивами даних. Як правило, такі операції виконуються більш ефективно і з меншою кількістю коду, ніж це можливо за допомогою вбудованих послідовностей Python.
- Зростає кількість наукових та математичних пакетів на основі Python, які використовують масиви NumPy; хоча вони, як правило, підтримують вхідні дані у вигляді послідовностей Python, але перед обробкою перетворюють їх у масиви NumPy, а також часто виводять їх у вигляді масивів NumPy. Іншими словами, для ефективного використання значної частини (можливо, навіть більшості) сучасного наукового/математичного програмного забезпечення на основі Python недостатньо просто знати, як використовувати вбудовані типи послідовностей Python - потрібно також знати, як використовувати масиви NumPy.

3.3.3. SciPy

SciPy - це набір математичних алгоритмів та зручних функцій, побудованих на розширенні NumPy мови Python. Він додає значної потужності інтерактивному сеансу Python, надаючи користувачеві високорівневі команди та класи для маніпулювання та візуалізації даних. За допомогою SciPy інтерактивний сеанс Python стає середовищем обробки даних і прототипування систем, що конкурує з такими системами, як MATLAB, IDL, Octave, R-Lab і SciLab. [15]

Додатковою перевагою базування SciPy на Python є те, що це також робить потужну мову програмування доступною для використання при розробці складних програм та спеціалізованих додатків. Наукові додатки, що використовують SciPy, отримують вигоду від розробки додаткових модулів у численних нішах програмного ландшафту розробниками по всьому світу. Все, від паралельного програмування до підпрограм і класів для роботи в Інтернеті та базах даних, стало доступним для програміста Python. Вся ця потужність доступна на додаток до математичних бібліотек в SciPy. [15]

Переваги використання даної бібліотеки:

- SciPy складається з різноманітних підпакетів, які допомагають вирішувати найпоширеніші завдання, пов'язані з науковими розрахунками.
- Пакет SciPy на мові Python є однією з найпоширеніших наукових бібліотек, яка уступає тільки GNU Scientific Library для C/C++ або Matlab.
- Зрозуміла та проста у використанні, а також має швидку розрахункову потужність.
- Підтримує роботу з масивами бібліотеки NumPy.

3.3.4. Matplotlib

Matplotlib – програмний пакет на мові програмування Python для візуалізації даних з використанням двовимірної 2D і тривимірної 3D графіки (Рис. 3.2). Існує ряд програмних засобів для створення графіків для доповідей та презентацій. Наприклад, можна відкрити CSV-файл в LibreOffice або Google Docs і побудувати в них графіки. Але що, якщо графіки або діаграми потрібно створювати регулярно, то для цього найкраще підходить Python і його пакет Matplotlib. [16]

Пакет Matplotlib за структурою нагадує SciPy, NumPy та IPython, також він надає можливості, подібні до пакету MATLAB. На даний час пакет працює з декількома графічними бібліотеками, включаючи wxWindows і PyGTK.

Пакет підтримує наступні види графіків та діаграм [16]:

- графіки (line plot);
- діаграми розсіювання (scatter plot);
- стовпчасті діаграми (bar chart) і гістограми (histogram);
- секторні діаграми (pie chart);
- деревоподібні діаграми (stem plot);
- контурні графіки (contour plot);
- поля градієнтів (quiver);
- спектральні діаграми (spectrogram).

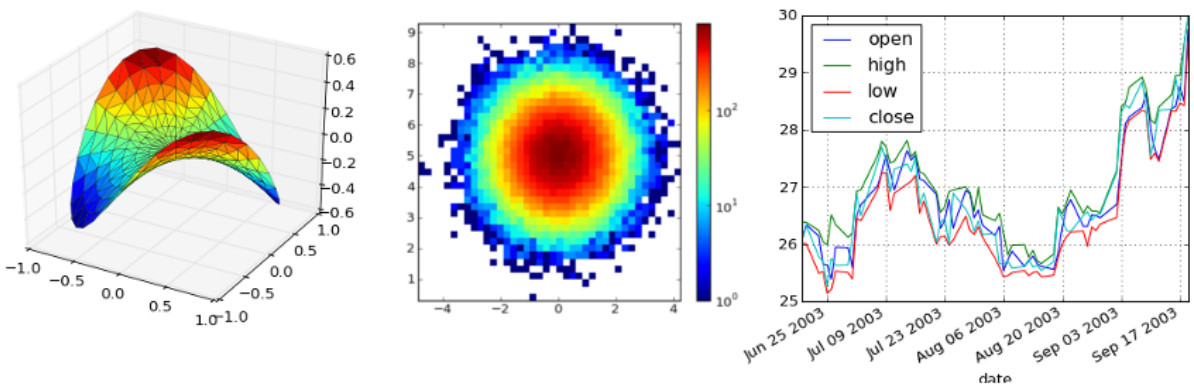

Рис. 3.2. Приклади графіків та діаграм побудованих за допомогою Matplotlib

Користувач має можливість самостійно побудувати сітку, задавати осі координат, додавати пояснення та підписи, користуватися логарифмічним масштабом чи полярними координатами.

За допомогою інструментарію mplot3d можна будувати прості тривимірні графіки. Також є і інші інструментальні засоби: для побудови картографічних зображень, утиліти для GTK, для роботи з Excel та інші. Завдяки Matplotlib ви можете створити анімацію або просто картинку.

3.3.5. TensorFlow

Машинне навчання досить складна галузь, але завдяки системам машинного навчання, таким як TensorFlow від компанії Google (які значно спрощують процеси збору даних, навчання моделей, прогнозування і уточнення майбутніх результатів), впровадження моделей машинного навчання стало значно простішим завданням, аніж воно здавалося раніше.

TensorFlow - це бібліотека з відкритим вихідним кодом, розроблена компанією Google в першу чергу для додатків глибокого навчання. Вона також підтримує традиційне машинне навчання. TensorFlow спочатку розроблявся для великих числових обчислень без урахування глибокого навчання. Однак вона виявилася дуже корисною і для розвитку глибокого навчання. [17]

Вона була створена командою Google Brain і вперше випущена для громадськості в 2015 році. TensorFlow об'єднує безліч моделей і алгоритмів машинного навчання і глибокого навчання (також відомих як нейронні мережі) і

робить їх корисними за допомогою загальних програмних метафор. Він використовує Python або JavaScript для надання зручного інтерфейсу API для побудови додатків, одночасно виконуючи ці додатки на високопродуктивній мові C++. TensorFlow також має широку бібліотеку попередньо навчених моделей, які можна використовувати у власних проектах. [18]

TensorFlow дозволяє розробникам створювати графіки потоків даних - структури, які описують, як дані рухаються через граф, або серію вузлів обробки. Кожен вузол в графі представляє математичну операцію, а кожен зв'язок або ребро між вузлами - багатовимірний масив даних, або тензор. [18]

TensorFlow працює на основі графів потоків даних, які мають вузли та ребра. Оскільки механізм виконання у вигляді графів, то набагато простіше виконувати код TensorFlow розподілено на кластері комп'ютерів при використанні графічних процесорів. [17]

За допомогою TensorFlow можна побудувати графіки потоків даних, що описуватимуть, як дані рухатимуться через граф. Граф - це сукупність вузлів, які являють собою математичну операцію. Зв'язок ("ребро") між вузлами - є багатовимірним масивом даних. Він приймає входи у формі багатовимірного масиву, за допомогою якого можна побудувати блок-схему операцій, котрі можуть бути виконані над цими входами.

Архітектура тензорного потоку працює в три важливих етапи [17]:

- Попередня обробка даних - структурування даних та приведення їх до одного граничного значення;
- Побудова моделі - побудова моделі для даних;
- Навчання та оцінка моделі - використання даних для навчання моделі та її тестування з невідомими даними.

3.3.6. Keras

Keras - це API для глибокого навчання, написаний на мові Python, що працює на платформі машинного навчання TensorFlow. Він був розроблений з акцентом на забезпечення швидкого експериментування (Рис. 3.3). Можливість якомога швидше

перейти від ідеї до результату є ключовим фактором для проведення якісних досліджень. [19]

Особливості Keras - саме такі [19]:

- Простий - але не спрощений. Keras зменшує когнітивне навантаження на розробника, дозволяючи зосередитися на тих частинах проблеми, які дійсно мають значення.
- Гнучкий - Keras приймає принцип поступового розкриття складності: прості робочі процеси повинні бути швидкими і легкими, в той час як довільно просунуті робочі процеси повинні бути можливими за допомогою чіткого шляху, який спирається на те, що ви вже вивчили.
- Потужний - Keras забезпечує найкращу в галузі продуктивність і масштабованість: його використовують такі організації та компанії, як NASA, YouTube або Waymo.

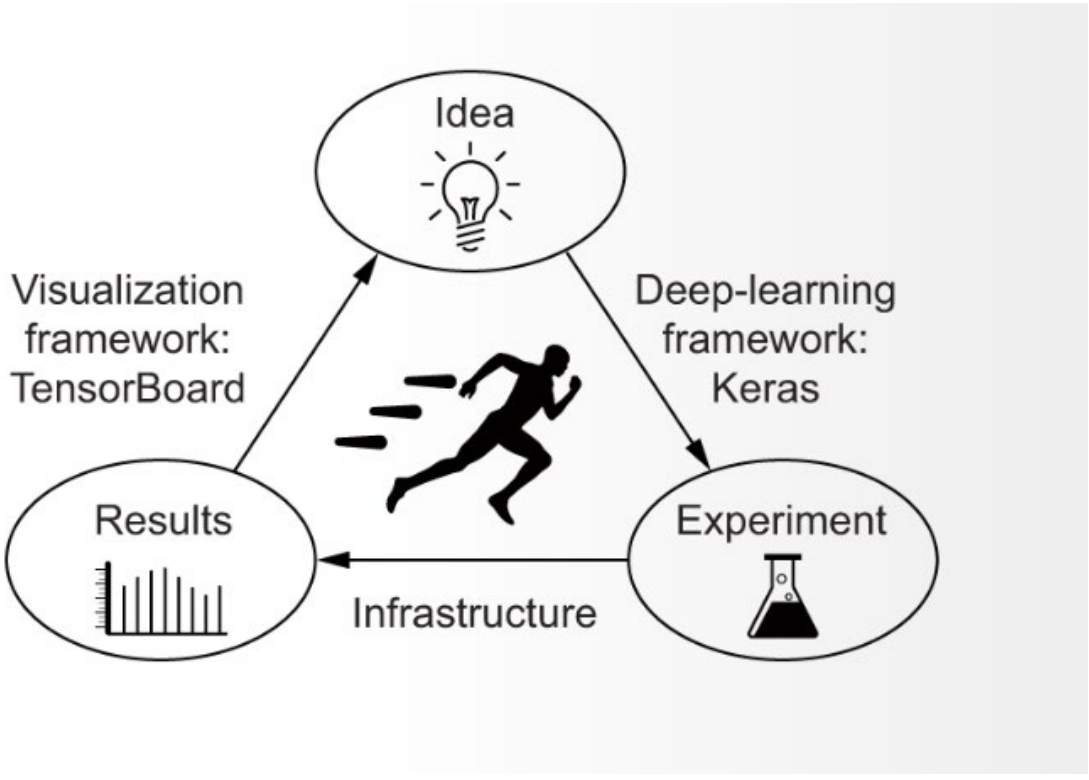

Рис. 3.3. Keras – допомагає пришвидшити процеси

Для чого потрібен Keras:

- Зручна побудова моделей, за якими відбуватиметься навчання.
- Налаштування прошарків у моделях - як правило, підбір необхідної кількості прошарків необхідний для точності.
- Опрацювання процесів введення і виведення інформації з моделі.
- Перетворення вхідних даних, які поступають у навчальну модель.
- Зручний підбір набору даних для навчання.
- Візуалізація моделі.
- Підготовка моделі до роботи, визначення її функцій похибки та оптимізаторів.
- Тренування і тестування моделі.
- Збирання та первинний запуск програми машинного навчання.

Усе вищезазначене можна робити і без Keras, займе значно більше часу і буде важче. Keras виконує роль програмного інтерфейсу, який спрощує дії.

3.3.7. Pandas

Pandas являє собою бібліотеку з відкритим вихідним кодом, яка забезпечує високопродуктивну маніпуляцію даними на мові Python. Назва Pandas походить від слова Panel Data, що означає економетрика з багатовимірних даних. Він використовується для аналізу даних на мові Python і був розроблений Уесом МакКінні у 2008 році. [20]

Pandas побудований на основі пакету Numpy, тобто для роботи Pandas потрібен Numpy.

До існування Pandas, Python мав змогу лише підготовлювати дані, але він мав змогу лише обмежено підтримувати аналіз даних. Отже, коли з'явився Pandas він розширив можливості аналізу даних. Мова може здійснювати п'ять основних кроків, необхідних для обробки та аналізу даних, незважаючи на походження даних, а саме: завантаження, обробка, підготовка, моделювання та аналіз.

Ключові особливості Pandas [20]:

- Має швидкий та ефективний об'єкт DataFrame з індексацією за замовчуванням та з можливістю налаштування.
- Використовується для реформування та повороту наборів даних.
- Групування за даними для агрегацій та перетворень.
- Використовується для вирівнювання даних та інтеграції відсутніх даних.
- Забезпечує функціональність часових рядів.
- Здатність обробляти різноманітні набори даних у різних форматах, таких як матричні дані, табличні гетерогенні дані, часові ряди.
- Здатність обробляти численні операції з наборами даних, такі як підстановка підмножин, нарізка, фільтрація, групування по групах, переупорядкування та переформування.
- Інтегрується з іншими бібліотеками, такими як SciPy та scikit-learn.
- Забезпечує швидку роботу, а якщо ви хочете прискорити її ще більше, ви можете використовувати Cython.

Перевагами Pandas над іншими мовами є:

- Представлення даних: Вона подає дані у формі, яка зручна для проведення аналізу даних за допомогою DataFrame та Series.
- Зрозумілий код: Зрозуміле API Pandas надає можливість сконцентруватися на головній частині коду. Таким чином, він надає чіткий і стислий код для користувача.

3.4. Реалізація програми

Щоб реалізувати проєкт була використана мова Python версія 3.7. Для роботи зі CNN були вибрані такі бібліотеки :

- Pandas (1.1.5)
- OpenCV(4.6.0.66)
- Numpy (1.16.0)
- Pillow(9.3.0)
- Matplotlib(3.4.0)
- H5py(2.10.0)

- Keras(2.4)
- Tensorflow(1.13.1)
- Scipy(1.2)

Може виникнути проблема зі сумісністю пакетів, якщо таке відбудеться, алгоритм не запуститься, для того щоб цього уникнути рекомендую використовувати версії пакетів, які встановлювала я, вони вказані у дужках.

Щоб алгоритм міг розпізнавати емоції йому треба якась база даних, на сьогоднішній день найти таку не потребує великих зусиль, тому що дана сфера активно розвивається. Я використала FER-2013, вона є можливо наймісткішою БД, у ній знаходиться близько 36 тисяч сірих зображень, кожне з яких містить в собі одну з 7 основних емоцій.

Використовуючи команду read_csv пакету даних pandas було збережено дані, які знаходяться в бібліотеці FER-2013 з форматом файлу *.csv.

Завдяки функціям з бібліотеки OpenCV, алгоритм здатний отримувати та обробляти відеодані з камери. За допомогою однієї з функції Haar Cascade Classifier, з цієї ж бібліотеки, програма визначає лиця людей. Після цього відбувається обробка даних і вивід результату у формі (Frame) на екран. Виводом тексту на екран займається функція draw_text з пакету даних Pillow.

Обробка даних відбувається згорткованою нейронною мережею, вона була досить легкою у написанні, завдяки Keras API, який у мене працює на TensorFlow.

3.5. Тестування роботи програми

Далі перейдемо до тестування розробленої програми. Запуск системи відбувається за допомогою командного рядка. Система розпізнає емоції в режимі реального часу. Результатом запуску є вікно програми яке транслює зображення з камери ком’ютера. Система розпізнає обличчя та емоцію яку людина показує. Результат відразу відображається на екрані і може змінюватись, паралельно зміні виразу обличчя.

Далі перевіримо як система розпізнає такі емоції:

1. Нейтральний вираз обличчя (Рис. 3.4).

2. Радість (Рис. 3.5).
3. Сум (Рис. 3.6).
4. Здивування (Рис. 3.7).
5. Злість (Рис. 3.8).

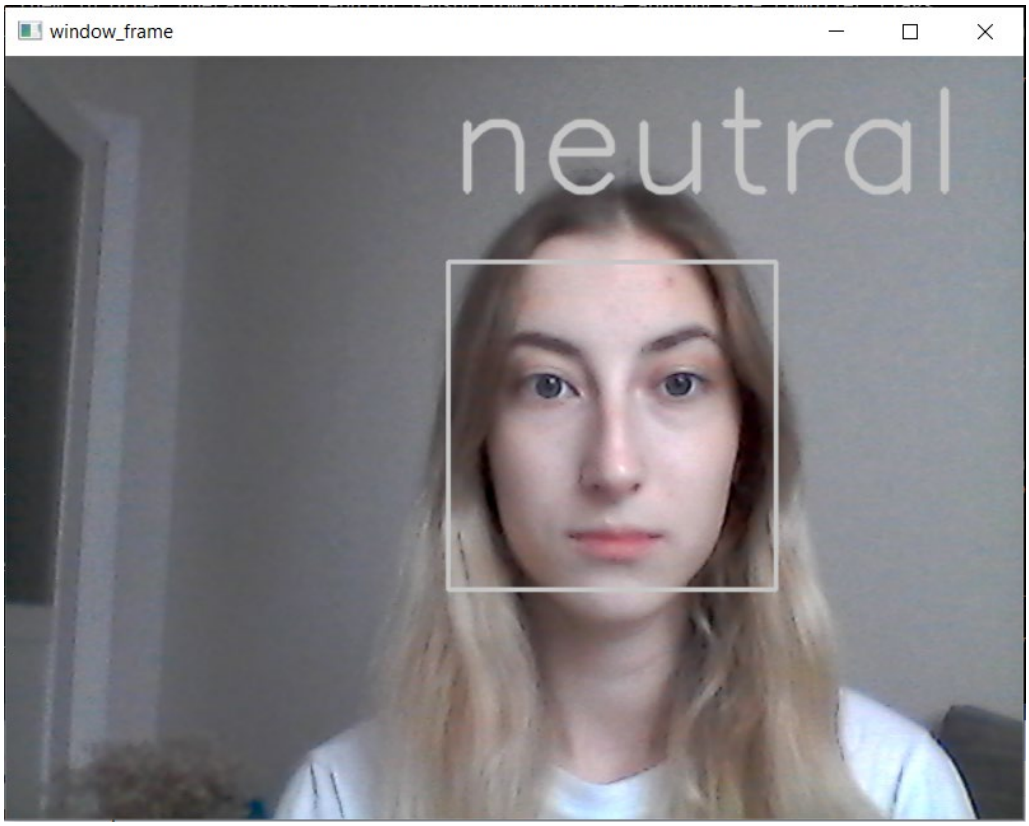

Рис. 3.4. Результат емоції «Нейтральної»

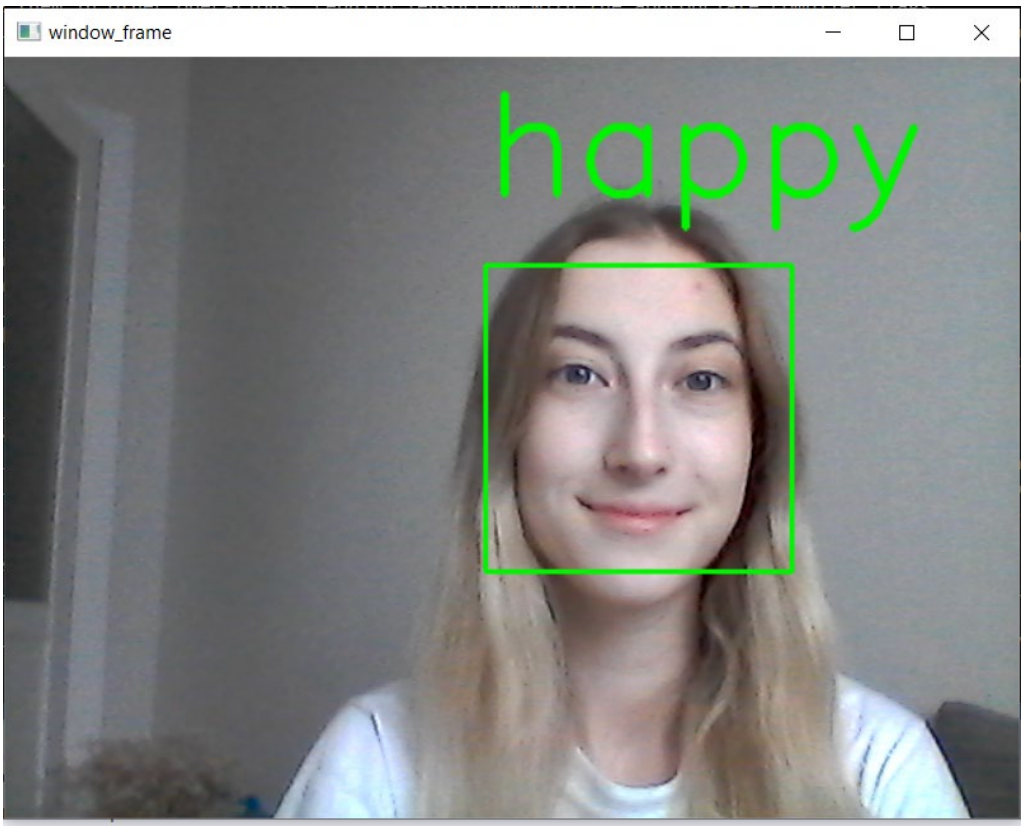

Рис. 3.5. Результат емоції «Радість»

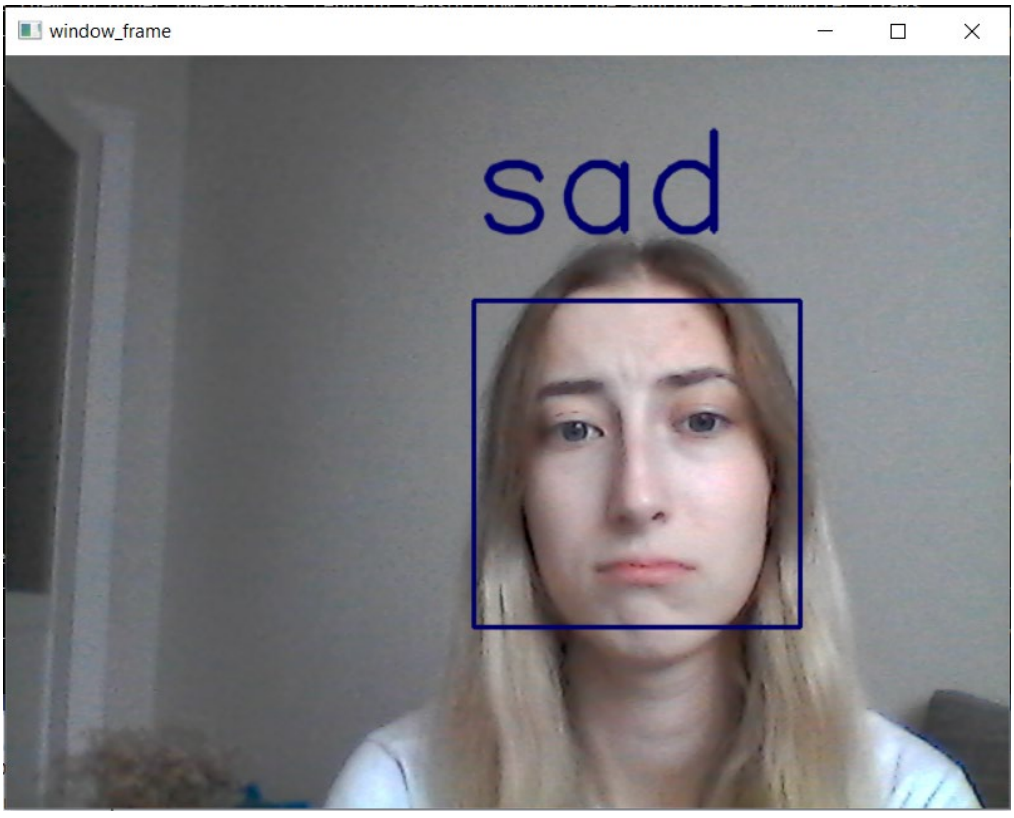

Рис. 3.6. Результат емоції «Сум»

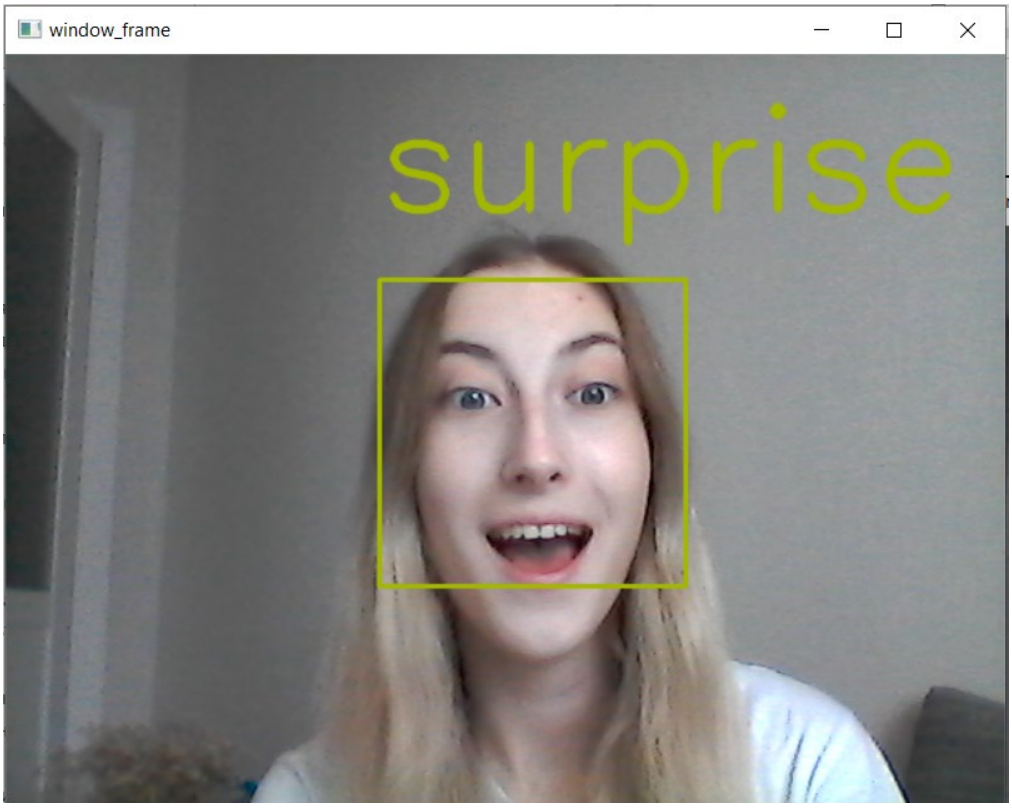

Рис. 3.7. Результат емоції «Здивування»

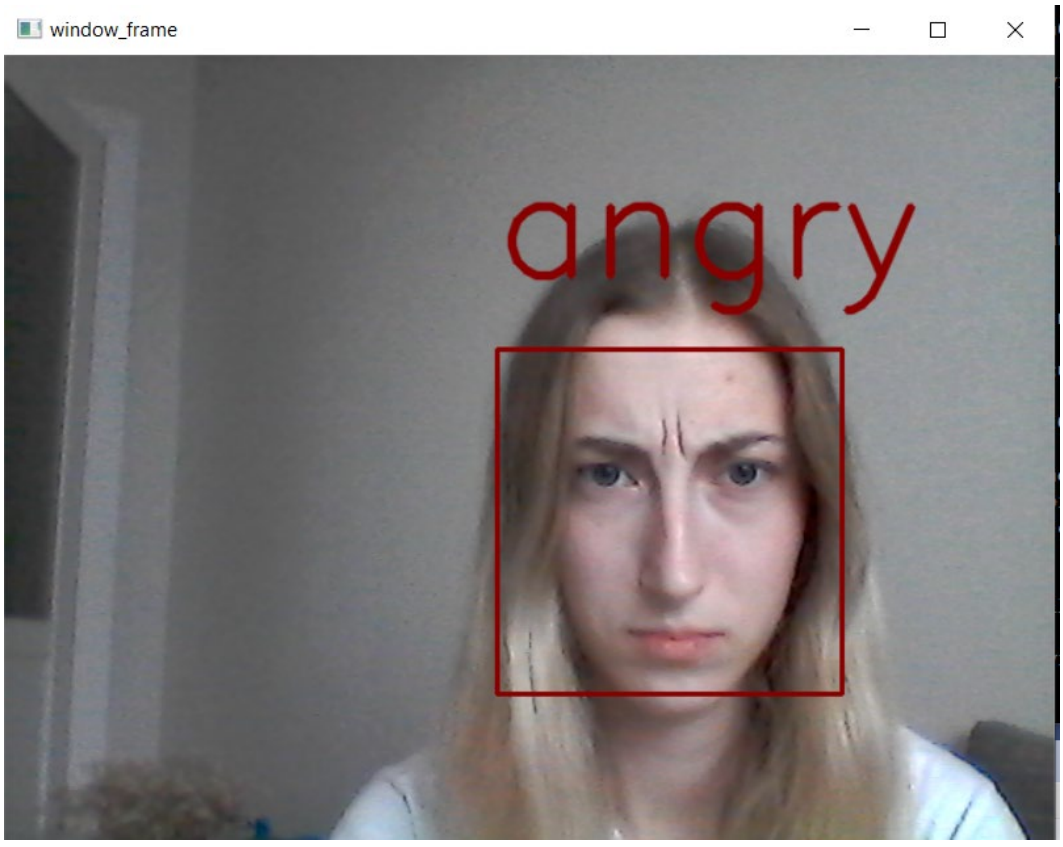

Рис. 3.8. Результат емоції «Злість»

Висновок: Система розпізнає емоції, але через погану якість зображення, часто допускається помилок. Програма потребує подальшої роботи над нею та вдосконалення.

ВИСНОВКИ

Якщо підбивати підсумки, то я зрозуміла, що емоційний ШІ дає можливість всім менеджерам підприємств бути емпатичними та зрозумілими для кожного споживача в потрібний для нього час. Це дозволяє компанії пропонувати відповідні товари та сервіси правильним клієнтам у правильний час, що призводить до більшої залученості клієнтів. Також ШІ може дуже сильно полегшити життя маркетологам і допомагати визначати, які рекламні ролики найбільше викликають інтерес у цільової аудиторії, і що саме потрібно включити в них, аби отримати найкращі результати. І так само в інших сферах життя, такий ШІ, може допомагати з різними завданнями соціального типу.

В ході роботи над дипломним проєктом, я дослідила високорівневу об’єктно-орієнтовану мову програмування Python. Вона виявилась дуже зручною саме для написання такого алгоритму. Ряд бібліотек, які вона може запропонувати звичайному користувачеві є дуже великим і сама мова є доволі простою, якщо порівнювати її з іншими високорівневими мовами. Використовуючи всі знання, які я отримала з досліджених джерел, я реалізувала алгоритм розпізнавання емоцій людини використовуючи архітектуру CNN у реальному часі.

Результатом реалізації є готовий програмний продукт, який виконує поставлену перед ним задачу.

Провівши ретельний аналіз різного роду матеріалів на тему штучного інтелекту з усією впевненістю можу сказати, що сьогодні темп розвитку технологій є дуже стрімким. Так, у програмних продуктах, які існують на даний час є дуже багато проблем, але згодом, коли ми найдем рішення до них, такі технології на базі ШІ не будуть здаватись нам чимось нереальним, а просто увійдуть у наше повсякденне життя. Звісно, я не можу навіть і близько зпрогнозувати, коли настануть такі реалії, але ми всі вже можемо побачити величезний потенціал у цій сфері.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Iryna Sydorenko. AI in Emotion Recognition: Does It Work? URL: https://labelyourdata.com/articles/ai-emotion-recognition (Дата звернення: 01.10.2022)
2. Емоційний AI: як технологія набуває людських рис. URL: https://evergreens.com.ua/ua/articles/emotion-ai.html (Дата звернення: 02.10.2022)
3. Олена Мусієнко. Як обійти технологію розпізнавання облич? Спроби дослідників. URL: https://www.imena.ua/blog/get-around-face-recognition-technology/ (Дата звернення: 05.10.2022)
4. Круглов В.В.. Искуственные нейронные сети. Теория и практика, 2-е изд. стериотип / В.В. Круглов, В.В. Борисов. – М.: Горячая линия – Телеком, 2002. – 8 с.
5. Савченко А. С., Синельніков О. О. Методи та системи штучного інтелекту: навч. посібник. К.: НАУ, 2017. 108-111 с.
6. С. О. Субботін НЕЙРОННІ МЕРЕЖІ: ТЕОРІЯ ТА ПРАКТИКА: навч.посібник. НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЗАПОРІЗЬКА ПОЛІТЕХНІКА", 2020. 88 с.
7. Sumit Saha A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Dec 15, 2018 URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (Дата звернення: 15.10.2022)
8. MK Gurucharan Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network URL: https://www.upgrad.com/blog/basic-cnn-architecture/ (Дата звернення: 17.10.2022)
9. The Python Package Index (PyPI) URL: http://omz-software.com/pythonista/docs/distutils/packageindex.html

(Дата звернення: 21.10.2022)

10. About URL: https://opencv.org/about/ (Дата звернення:)
11. OpenCV – Overview URL: https://www.geeksforgeeks.org/opencv-overview/ (Дата звернення: 22.10.2022)
12. What is NumPy? URL: https://numpy.org/doc/stable/user/whatisnumpy.html#whatisnumpy (Дата звернення: 22.10.2022)
13. NumPy Introduction URL: https://www.w3schools.com/python/numpy/numpy_intro.asp (Дата звернення: 22.10.2022)
14. NumPy: the absolute basics for beginners URL: https://numpy.org/doc/stable/user/absolute_beginners.html (Дата звернення: 22.10.2022)
15. Introduction https://docs.scipy.org/doc/scipy/tutorial/general.html (Дата звернення: 23.10.2022)
16. Matplotlib URL: https://kkite.pnu.edu.ua/wp-content/uploads/sites/50/2020/04/Matplotlib.pdf (Дата звернення: 23.10.2022)
17. What is TensorFlow? URL: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow#what_is_tensorflow (Дата звернення: 23.10.2022)
18. Serdar Yegulalp. What is TensorFlow? The machine learning library explained URL: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html (Дата звернення: 23.10.2022)
19. About Keras URL: https://keras.io/about/ (Дата звернення: 23.10.2022)
20. Pandas. URL: https://www.javatpoint.com/python-pandas (Дата звернення: 23.10.2022)

Додатки

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_default.xml'
emotion_model_path = '../trained_models/emotion_models/fer2013_mini_XCEPTION.102-
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)
```

cnn.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
```