

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ інфокомунікацій _____
(повна назва)

Кафедра _____ інформаційно-мережної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Дослідження методів розпізнавання статі та віку людини за її зображенням

(тема)

Виконав:
студент 2 курсу, групи ІМІМ-22-1
Лапенко Д.О.
(прізвище, ініціали)

Спеціальність 172 «Телекомунікації та
радіотехніка»
(код і повна назва спеціальності)

Освітня програма Інформаційно-мережна
інженерія
(повна назва освітньої програми)

Керівник доц. каф. ІМІ Омельченко С.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Безрук В.М. _____
(підпис) (прізвище, ініціали)

2024 р.

Не містить відомостей заборонених до відкритого публікування

Студент _____ / Лапенко Д.О./

Керівник _____ /Омельченко С.В./

Харківський національний університет радіоелектроніки

Факультет _____ інфокомунікацій _____
Кафедра _____ інформаційно-мережної інженерії _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 172 «Телекомунікації та радіотехніка» _____
Тип програми _____ освітньо-професійна _____
(код і повна назва)
Освітня програма _____ інформаційно-мережна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Лапенко Данилу Олександровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Дослідження методів розпізнавання статі та віку людини за її зображенням _____

затверджена наказом університету від 23 жовтня 2023 р. № 1233 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 31 січня 2024 р.

3. Вихідні дані до роботи _____

Дослідити методи розпізнавання статі та віку людини за її зображенням та проаналізувати мову програмування Python, бібліотеки Keras, TensorFlow, OpenCV, OpenVino, Pillow, PyTorch, Dlib, а також дослідити їх алгоритми.

4. Перелік питань, що потрібно опрацювати в роботі _____
Вступ _____

1 Огляд методів визначення статі та віку людини _____

2 Особливості оцінювання ознак та побудови вирішуючих правил в методах розпізнавання статі та віку людини _____

3 Засоби розробки програмного забезпечення для визначення статі та віку людини _____

4 Дослідження алгоритму для визначення статі та віку людини _____

Висновки _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 вкдючається до завдання з рішенням випускової кафедри) _____

Слайди у форматі Power Point (задачі та мета роботи, актуальність роботи, вступ,

поняття штучного інтелекту, машинного і глибокого навчання, нейронні мережі,

глибокі нейронні мережі, згорткова нейронна мережа, рекурентна нейронна мережа,

бібліотеки Keras та TensorFlow, реалізація проекту для розпізнавання статі та віку

людини, реалізація визначення атрибутів особи (вік, стать) на python, висновки)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Ознайомлення із завданням. Уточнення ТЗ.	23.10.2023	вик.
2	Підбір літератури за темою роботи	24.10-27.10.2023	вик.
3	Огляд методів визначення статі та віку людини	28.10.-11.11.2023	вик.
4	Особливості оцінювання ознак та побудови	12.11.-28.11.2023	вик.
	вирішуючих правил в методах розпізнавання		
	статі та віку людини	29.11.-07.12.2023	вик.
5	Засоби розробки програмного забезпечення для		
	визначення статі та віку людини		
6	Дослідження алгоритму для визначення статі та	08.12.-31.12.2023	вик.
	віку людини		
7	Оформлення презентаційного матеріалу,	01.01.-11.01.2024	вик.
	підготовка до захисту у ЕК		

Дата видачі завдання 23 жовтня 2023 р.

Студент _____

(підпис)

Керівник роботи _____ доц. каф. ІМІ Омельченко С.В.

(підпис)

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи по темі: “ Дослідження методів розпізнавання статі та віку людини за її зображенням ”

Робота містить 63 сторінок, 19 рисунків, 1 додаток та 19 бібліографічних найменувань.

Метою роботи є дослідження програмного забезпечення, яке за умови що обличчя людини потрапляє до зони видимості камери з використанням моделі машинного навчання приймає рішення про належність до статевої та вікової групи осіб.

Дослідити процеси визначення статі та віку людини за її зображенням та проаналізувати мову програмування Python, бібліотеки Keras, TensorFlow, OpenCV, OpenVino, Pillow, PyTorch, Dlib, а також дослідити їх характеристики.

Розглянути обробку розпізнавання облич, включаючи такі ключові компоненти, як розпізнавання обличчя, відстеження, вирівнювання та розділення функцій, а також висвітлити технічні проблеми створення системи розпізнавання облич. Зробити особливий акцент на важливості найуспішніших рішень, доступних на даний момент. Описати вибрані методи для застосування розпізнавання облич та їх можливе використання в різних сферах.

Ключові слова: дослідження методів розпізнавання статі та віку людини за її зображенням, нейронна мережа, комп'ютерний зір, Keras, TensorFlow, OpenCV, OpenVino, Pillow, CNN, PyTorch, Dlib, Python.

THE ABSTRACT

Explanatory note to the qualification work on the topic: " Research into methods for recognizing a person's gender and age using his image "

The work contains 63 pages, 19 figures, 1 appendix and 19 bibliographic titles.

The aim of the work is to study software that, provided that a person's face falls into the field of view of the camera using a machine learning model, decides whether to belong to the gender and age group of persons.

Investigate the processes of determining the sex and age of a person by his image and analyze the Python programming language, Keras, TensorFlow, OpenCV, OpenVino, Pillow, PyTorch, Dlib libraries, as well as explore their characteristics.

Consider face recognition processing, including key components such as face recognition, tracking, alignment, and separation of functions, and highlight the technical issues of creating a face recognition system. Emphasize the importance of the most successful solutions available at the moment. Describe selected methods for the application of face recognition and their possible use in various fields.

Key words: research into methods for recognizing a person's gender and age using his image, neural network, computer vision, Keras, TensorFlow, OpenCV, OpenVino, Pillow, PyTorch, Dlib, Python.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	9
ВСТУП.....	10
1 ОГЛЯД МЕТОДІВ ВИЗНАЧЕННЯ СТАТІ ТА ВІКУ ЛЮДИНИ	12
1.1 Ідентифікація статі та віку людини	12
1.2 Розгляд технології ідентифікації статі та віку людини	15
1.3 Аналіз комп'ютерного зору	17
1.4 Штучні нейронні мережі	20
2 ОСОБЛИВОСТІ ОЦІНЮВАННЯ ОЗНАК ТА ПОБУДОВИ ВИРІШУЮЧИХ ПРАВИЛ В МЕТОДАХ РОЗПІЗНАВАННЯ СТАТІ ТА ВІКУ ЛЮДИНИ.....	23
2.1 Методи та алгоритми розпізнавання статі та віку людини	23
2.2 Вейвлети Габора	26
2.3 Дескриптор функції	27
2.4 Тривимірне розпізнавання облич	29
2.5 Глибокі нейронні мережі.....	30
2.6 Згорткові нейронні мережі	33
2.7 Рекурентні нейронні мережі	35
2.8 Висновок до розділу	39
3 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ СТАТІ ТА ВІКУ ЛЮДИНИ	40
3.1 Огляд програмного забезпечення	40
3.2 Мова програмування Python	41
3.3 Бібліотеки Keras та TensorFlow	43
3.4 Бібліотека комп'ютерного зору OpenCV	45
3.5 Бібліотека комп'ютерного зору OpenVino	47
3.6 Бібліотека з відкритим кодом Dlib	50
3.7 Бібліотека PyTorch.....	52
3.8 Бібліотека Pillow	54
4 ДОСЛІДЖЕННЯ АЛГОРИТМУ ДЛЯ ВИЗНАЧЕННЯ СТАТІ ТА ВІКУ ЛЮДИНИ	55
4.1 Реалізація проекту для розпізнавання статі та віку людини за її зображенням	55
4.2 Аналіз проекту для розпізнавання статі та віку людини за її зображенням	59

ВИСНОВКИ	61
ПЕРЕЛІК ПОСИЛАНЬ	62
ДОДАТОК А СЛАЙДИ ПРЕЗЕНТАЦІЇ	64

ПЕРЕЛІК СКОРОЧЕНЬ

1. CNN — Convolutional Neural Network, згорткова нейронна мережа.
2. CV — Computer Vision, комп'ютерний зір.
3. ANN — Artificial Neural Network, штучна нейронна мережа.
4. DNN — Deep Neural Networks, глибокі нейронні мережі.
5. RNN — Recurrent Neural Network, рекурентна нейронна мережа.
6. BRNN — Двостороння рекурентна нейронна мережа.
7. LSTM — Long short-term memory, довготривала короткочасна пам'ять.
8. ReLU — rectified linear unit, функція активації для нейронних мереж.
9. ML — машинне навчання.
10. API - це набір готових класів, процедур, функцій, структур і констант, що надаються бібліотекою, сервісом для використання в зовнішніх програмних продуктах.
11. GPU — Graphics Proccesing Unit, графічний процесор комп'ютера.
12. CPU — Central processing unit, центральний процесор.

ВСТУП

Одне з перших рішень, які люди приймають після знайомства це класифікація людей включаючи стать та вік. Ці рішення ґрунтуються на низці різних характеристик, які надходять не лише від особистості, але й від середовища.

Відділи маркетингу або продажів компаній зазвичай працюють з цільовими клієнтами їхньої продукції. Тому важливо мати інформацію про свою цільову аудиторію в різних сферах. Подібним чином деякі послуги, ліцензії чи продукти доступні лише для аудиторії певної статі чи віку та потребують певного контролю.

Поєднання цих двох ідей робить необхідним автоматичне визначення статі та віку. Люди легко оцінюють ці обличчя, але це все одно важке завдання для комп'ютера. Метою цієї роботи є дослідження методів розпізнавання статі та віку людини за її зображенням.

Незважаючи, що прототип системи працюватимуть майже так само добре, як і люди, він має деякі обмеження. Існує багато інформації, як голос, одяг, поведінка, яку люди свідомо чи несвідомо використовують для визначення віку чи статі, що не використовуються штучною системою. Тому продуктивності людини важко досягти. Навіть коли люди працюють над зображеннями обличчя, вони використовують деяку інформацію високого рівня, таку як колір шкіри, волосся та волосся на обличчі. З іншого боку, система використовуватиме методи комп'ютерного зору, які люди не можуть використовувати. Оскільки є багато додатків і кожен має свої особливості, тому ми намагаємося знайти компроміс між помилкою та обчислювальними зусиллями при використанні наданих ресурсів.

Для навчання системи необхідна база даних зображень обличчя за віком і статтю. Ця база даних показує фактичні вхідні зображення системи. Етапи кожного процесу оцінки: попереднє дослідження, виділення ознак, оцінка. Етап попередньої обробки визначається для підготовки даних до етапу вилучення ознак. Ці методи працюють на рівні пікселя, рівня об'єкта (відстеження ключових точок) або на рівні зображення.

Для системи потрібні два типи специфікацій: специфікації високого рівня та специфікації низького рівня. Функції високого рівня – це функції, які безпосередньо пов'язані з віком/статтю, наприклад: волосся, борода, вуса або

зморшки, наприклад. З іншого боку, функції низького рівня не мають чіткого візуального зв'язку з прогнозом, і градієнт витягується за допомогою найпоширеніших методів комп'ютерного зору, таких як Гессе, різниця пікселів або градієнт, керований гистограмою (HOG).

Між віком і статтю існує важлива різниця: для статі є дві різні категорії (чоловіча та жіноча), тоді як для оцінки віку оцінювані значення є безперервними, тому метод оцінки має бути останнім. Тому соціальна стать визначається категоріальним методом, а вік – методом регресії.

1 ОГЛЯД МЕТОДІВ ВИЗНАЧЕННЯ СТАТІ ТА ВІКУ ЛЮДИНИ

1.1 Ідентифікація статі та віку людини

Розпізнавання віку та статі обличчя має такі ключові моменти, як прогнозування профілю споживача, реклама в соціальних мережах, взаємодія людини з комп'ютером, система пошуку зображень, демографічне профілювання, індивідуальні рекламні системи, безпека та спостереження. Представлено дослідження одноатрибутних (атрибут: стать або вік) та багатоатрибутних (стать і вік) моделей прогнозування. Огляд методів оцінки віку обличчя та гендерної класифікації на основі традиційних і розроблених на даний момент підходів глибокого навчання, аналізуючи їхні переваги та недоліки та ідеї для майбутніх досліджень.

Обличчя людини надає інформацію про різноманітні впізнавані характеристики, такі як стать, вік, емоції та етнічна приналежність. Прогнозування статі обличчя та віку є дуже важливим для цих ознак, оскільки вони широко використовуються та можуть бути використані. Розпізнавання статі обличчя визначається як класифікація статі людини на основі моделі обличчя в певний клас (чоловіча, жіноча). Прогнозування віку людини автоматично передбачає біологічний вік людини або її вікової групи: дитини, дорослого, літніх людей тощо.

Прогнозування статі обличчя та оцінка віку мають комерційні програми в режимі реального часу, такі як неінвазивне криміналістичне профілювання жертв/злочинців, відстеження за статтю та віком, взаємодія людини з комп'ютером, правоохоронні органи, контроль доступу та інтерактивні системи. Для моніторингу та контролю доступу можна використовувати для обмеження доступу осіб певної статі чи вікової групи до зон обмеженого доступу; дозволити доступ до сайту за певною віковою групою чи статтю; доступ до веб-додатку; контроль доступу до фізичних (місця для куріння, туалет) і небезпечних зон, таких як парк розваг тощо. Наприклад, в Японії торговельні автомати рекомендують напої (алкогольні напої, сигаретні пачки) за обличчям (віком) дорослих покупців.

У комерційних програмах відеоспостереження його можна використовувати для демографічного аналізу або порушення правил. Наприклад, купе, метро, автобуси, туалети та гуртожитки мають обмежений доступ для

людей певної статі, а пасажирів або гостей можна автоматично спрямовувати та контролювати за будь-якими порушеннями закону. Крім того, ці прогнози можна використовувати в стратегіях продажів і маркетингу та бізнес-плануванні, наприклад, щоб визначити кількість відвідувачів із профілями (чоловіки, жінки, підлітки, молодь, дорослі) у певних сферах, таких як громадські місця, торгові центри, банки. Можливо використовувати їх для адресної реклами на електронних білбордах для вікових груп, що динамічно змінюються. Системи прогнозування статі та обличчя також можна використовувати для розробки таких послуг, як автоматизовані системи охорони здоров'я у медичних установах. Визначення статі та віку нещодавно було додано до смартфонів як частину функції. Їх можна використовувати для автоматичного перепорядкування альбому та керування такими функціями, як упорядкування, вилучення та видалення зроблених фотографій залежно від вибраного віку та статі. Системи, засновані на взаємодії людини з комп'ютером, такі як автоматизована система інтелекту, використовуються для розпізнавання характеристик обличчя людини, таких як стать і вік, шляхом аналізу фізіологічної поведінки. Розпізнавання статі також можна використовувати для зменшення індексу пошуку в базі даних біометричних систем. Це також підвищує точність ідентифікації людини на основі віку та статі як рис обличчя. Ці результати також можуть бути використані в криміналістиці для прогнозування найкращої відповідності між зниклими особами та методами розпізнавання облич. Крім того, його можна використовувати для створення оновленого образу обличчя старого родича або зниклої дитини за допомогою синтезу віку обличчя.

Є два способи передбачити риси обличчя. Одноатрибутне навчання (SAL) або однозадачне навчання (STL), багатоатрибутне навчання (MAL) або багатозадачне навчання (MTL) у підході, який використовує SAL/STL для кожна функція/клас (стать або вік) навчаються або прогнозуються окремо без будь-якої кореляції. між різними функціями, тоді як підхід MAL (MTL) передбачає вивчення кількох функцій (для прогнозування статі та віку) за допомогою загальної паралельної моделі.

Стать можна передбачити за обличчям, голосом, фотографіями обличчя, відбитками пальців, шкірою рук і почерком, хоча вік також можна передбачити за антропологією кісток або обличчя. Обличчя є бажаною рисою, тому що його легко побачити. Сучасні сучасні методи розпізнавання обличчя та статі можна

розділити на дві категорії: традиційні ручні методи розробки ознак і методи на основі глибокого навчання.



Рисунок 1.1 — Ідентифікація людей

Загальні ознаки покладаються на дескриптори функцій, створені розробниками алгоритмів, щоб унікально відображати вік і стать на зображеннях обличчя. Ці ознаки є ефективними для демонстрації значного визнання віку та статі в контрольованих умовах. Але існують відмінності в продуктивності за неконтрольованих умов, які не враховуються при проектуванні. Різні методи традиційного проектування об'єктів включають методи на основі текстур, такі як локальне двійкове шаблонування (LBP).) Функції градієнта гістограми (HOG) на основі Хаара; Методи зменшення розмірності, такі як PCA, ICA; Такі методи виділення ознак, як дискретне косинусне перетворення (DCT), інваріантне перетворення ознак (SIFT), масштабування з орієнтирами граней. (відстань або статистика).

Останнім часом методи глибокого навчання привернули великий інтерес дослідників. Оскільки мультимедійні дані та передові обчислювальні системи, такі як графічні процесори, прості у використанні. Підхід глибокого навчання використовує згорточні нейронні мережі (CNN) для вилучення ознак віку/статі з великих наборів даних зображень обличчя за допомогою статистичного навчання з потужними можливостями нелінійного моделювання. Якщо мережа складна і глибока, можлива надмірна установка. І в простих моделях, якщо не вистачає зображень обличчя

Багатоатрибутне прогнозування (MAL або MTL), таке як оцінка віку та розпізнавання статі обличчя, розширює сферу використання та прийнятність програм. Крім того, MTL покращує точність системи за допомогою передбачення комбінації коваріатів, коли один атрибут не дає правильного прогнозу. Підхід глибокого навчання є хорошим MAL/MTL; отже, пропонується декілька методів вивчення кількох ознак (віку та статі) за допомогою CNN.

1.2 Розгляд технології ідентифікації статі та віку людини

Розпізнавання статі та віку людини за зображенням передбачає використання комп'ютерного зору та технологій обробки зображень. Загалом, систему розпізнавання людини можна визначити як процес розміщення людини в об'єктиві камери за допомогою бази даних попередньо збережених і визначених зображень.

Стандартна схема застосування: IP-камера відправляє відеопотік на сервер, спеціальне програмне забезпечення аналізує відеопотік і порівнює зображення людини у відеопотоці з оригінальним обличчям. Недоліками таких систем є велике навантаження на мережу та висока вартість серверів. Навіть найпотужніший сервер може підключити обмежену кількість IP-камер.

Ось деякі технології, які зазвичай використовуються для цієї мети:

Технологія розпізнавання обличчя. Алгоритми розпізнавання обличчя, які часто базуються на моделях глибокого навчання, як згорткові нейронні мережі (CNN), широко використовуються для аналізу рис обличчя. Ці алгоритми можуть ідентифікувати атрибути обличчя, які вказують на стать і вік, наприклад, контури обличчя, довжину волосся та інші відмінні риси.

Моделі керованого машинного навчання можна навчити на великих наборах даних, що містять зображення, позначені інформацією про стать і вік. Ці моделі вчать розпізнавати моделі та особливості, пов'язані з певною статтю та віковою групою.

Теплове інфрачервоне розпізнавання обличчя – це біометрична технологія, яка використовує інфрачервоне випромінювання, випромінюване людським обличчям, для ідентифікації та автентифікації осіб. Традиційні системи розпізнавання обличчя покладаються на видиме світло, але теплове інфрачервоне розпізнавання обличчя працює в інфрачервоному спектрі, виявляючи тепло, що випромінюється обличчям.

Існує 3D розпізнавання обличчя. Деякі системи використовують технології 3D-розпізнавання обличчя для захоплення та аналізу геометрії обличчя. Це може надати точнішу інформацію про структуру обличчя та вікові зміни.

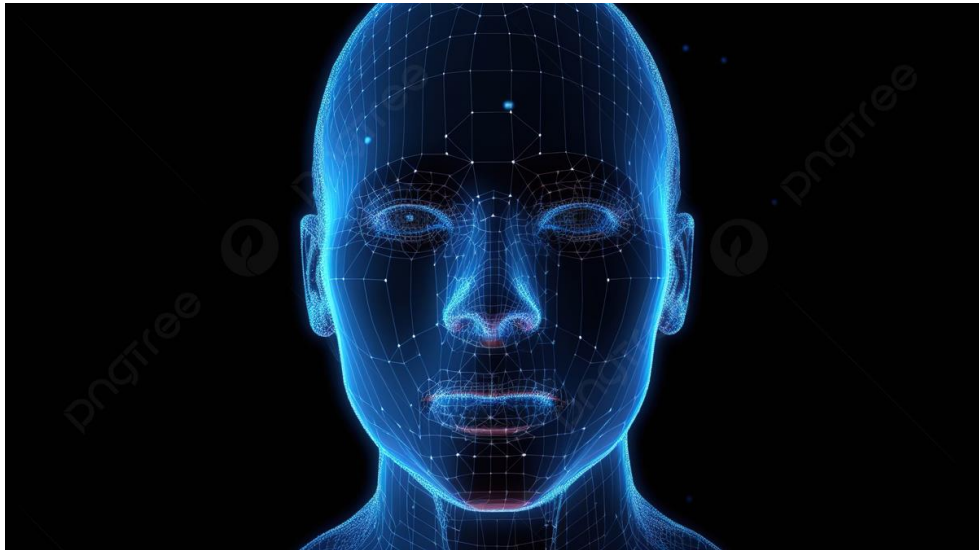


Рисунок 1.2 — 3D розпізнавання обличчя

Існує аналіз атрибутів. Методи обробки зображень застосовуються для аналізу певних атрибутів обличчя людини, таких як форма очей, носа та рота. Потім ці атрибути використовуються для прогнозування статі та оцінки віку.

Існує аналіз емоцій. Аналіз емоцій за виразом обличчя також можна використовувати як непрямий показник віку. Наприклад, певні вирази можуть бути пов'язані з певними віковими групами.

Існують підходи на основі даних. Підходи на основі даних передбачають збір і аналіз великих наборів даних зображень обличчя для виявлення статистичних моделей, пов'язаних із статтю та віком. Потім ці дані використовуються для навчання та вдосконалення моделей розпізнавання.

Існують програми що реалізуються в реальному часі. Деякі системи розроблено для додатків у режимі реального часу, що дозволяє аналізувати онлайн або записані відеопотоки відразу. Це особливо корисно в галузі безпеки та спостереження.

Існує етнічна належність і регіон. Деякі системи враховують регіональні та етнічні відмінності в рисах обличчя, оскільки ці характеристики можуть впливати на точність моделей прогнозування статі та віку.

Важливо зазначити, що розгортання таких технологій викликає етичні

міркування, пов'язані з конфіденційністю, згодою та потенційними упередженнями в даних, які використовуються для навчання. Встановлення балансу між технологічним прогресом і етичними міркуваннями є вирішальним у розробці та розгортанні систем розпізнавання статі та віку.

1.3 Аналіз комп'ютерного зору

Комп'ютерне зір — це галузь інформатики, яка зосереджена на моделюванні частин комп'ютера. Таке використання складної системи зору людини дозволяє комп'ютерам розпізнавати та обробляти об'єкти на фотографіях і відео так само, як це роблять люди. До недавнього часу комп'ютерний зір міг працювати лише в обмеженому діапазоні.

Завдяки прогресу в області штучного інтелекту та інноваціям у глибокому навчанні та нейронних мережах. За останні роки ця сфера досягла значного прогресу. Штучний інтелект краще за людей справляється з деякими завданнями, пов'язаними з розпізнаванням і маркуванням об'єктів. Одним із рушійних факторів у розвитку комп'ютерного зору є обсяг даних, який ми створюємо сьогодні. Його використовуватимуть для тренування та покращення комп'ютерного зору.

Одне з великих відкритих питань як у нейронауці, так і в машинному навчанні полягає в наступному: як насправді працює наш мозок і як ми можемо це передбачити за допомогою власних алгоритмів? Реальність така, що існує дуже мало працюючих і вичерпних теорій мозкових обчислень. Хоча нейронні мережі мають «імітувати те, як працює мозок», ніхто не впевнений, чи це насправді так.

Той самий парадокс стосується комп'ютерного зору: оскільки ми не з'ясували, як мозок і очі обробляють зображення, важко сказати, наскільки алгоритми, які використовуються у виробництві, наближаються до наших внутрішніх процесів.

На певному рівні комп'ютерний зір — це розпізнавання образів. Отже, один із способів навчити комп'ютер розуміти візуальні дані — це надати йому зображення, багато зображень, тисячі, якщо можливо, мільйони зображень, які були позначені, а потім об'єднати їх за допомогою різних програмних технік або алгоритмів. Піддаватися впливу міток, які дозволяють комп'ютеру бачити шаблони в усіх елементах, на які ці мітки посилаються.

Проста ілюстрація буфера зображення у градаціях сірого, який містить

зображення Авраама Лінкольна. Яскравість кожного пікселя представлена одним 8-бітним числом від 0 (чорний) до 255 (білий):

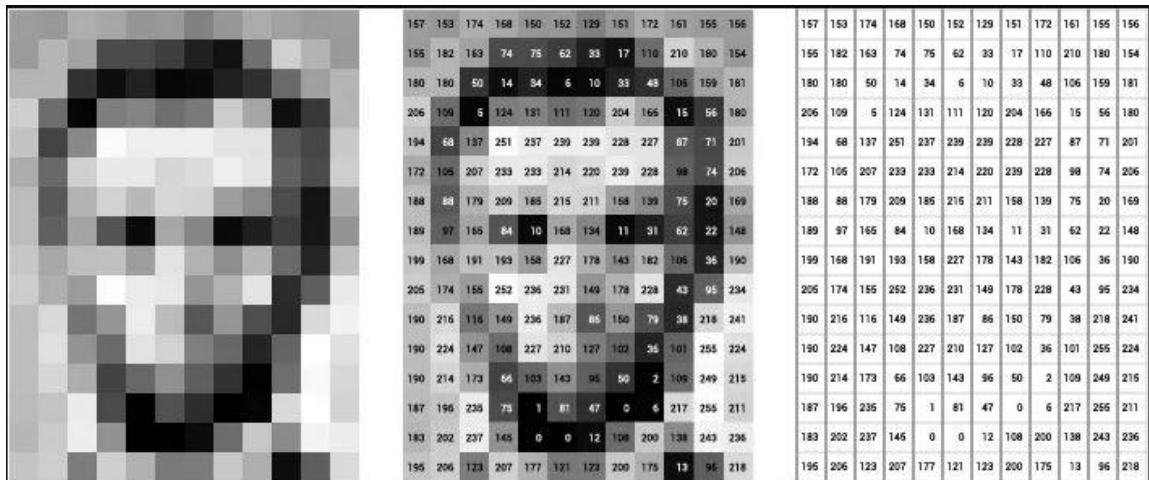


Рисунок 1.3 — Діаграма піксельних даних

Уявимо, що додаємо колір. Тепер усе ще складніше. Комп'ютери зазвичай зчитують колір як набір із трьох значень: червоний, зелений, синій (RGB) за шкалою від 0 до 255. Тепер кожен піксель фактично має три значення, які комп'ютер може зберігати, крім його положення. Якщо ви розфарбовуєте президента Лінкольна, ви отримаєте значення $12 \times 16 \times 3$, або 576.

До появи глибокого навчання комп'ютерний зір мав дуже обмежені завдання, які він міг виконувати, вимагаючи багато ручного кодування та зусиль з боку розробників і операторів. Наприклад, якщо хочете розпізнати обличчя, потрібно було виконати такі дії:

- Створити базу даних: потрібно зберігати окремі зображення кожного об'єкта, який я хочу відстежувати, у певному форматі.
- Описати зображення: Далі для кожного зображення потрібно ввести кілька важливих даних, таких як відстань між очима, розмір перенісся, відстань між верхньою частиною і носом.
- Створити новий образ. Далі потрібно зробити нові зображення, наприклад фотографії чи відео. Пройти процес уточнення, позначивши важливі точки на зображенні. Також слід враховувати ракурс зроблених знімків.

Після всієї цієї роботи вручну програма зможе порівняти розміри нового зображення з тими, що зберігаються в базі даних, і повідомити вам, чи відповідає воно будь-якому з профілів, які ви відстежуєте. Насправді практики дуже мало і більша частина роботи виконується вручну. А помилки все одно були великі.

Машинне навчання запропонувало інший підхід до вирішення проблем комп'ютерного зору. Завдяки машинному навчанню розробникам не потрібно вручну вводити кожне правило в свої програми для створення. Натомість вони програмують «об'єкти», невеликі об'єкти, які можуть виявляти певні візерунки на зображеннях. Алгоритми статистичного навчання, такі як лінійна регресія, логістична регресія, дерева рішень і опорні векторні машини, потім використовуються для виявлення шаблонів, класифікації зображень і розпізнавання об'єктів на зображеннях.

Машинне навчання допомогло вирішити багато проблем, які історично було важко вирішити за допомогою традиційних інструментів і методів розробки програмного забезпечення.

Глибоке навчання пропонує принципово інший підхід до машинного навчання. Глибоке навчання базується на нейронних мережах, які є функціями загального призначення, які можуть вирішити будь-яку проблему, яку можна представити на прикладах. Коли ви надаєте нейронній мережі багато символічних прикладів певного типу даних, вона виділяє загальні шаблони між прикладами та перетворює їх на математичні формули, які допомагають класифікувати інформацію в майбутньому.

Наприклад, щоб створити програму розпізнавання обличчя з глибоким навчанням, ви розробляєте або вибираєте попередньо написаний алгоритм для навчання алгоритму на прикладах людських облич, які потрібно розпізнати. Навівши достатньо багато прикладів, нейронна мережа може виявляти обличчя без будь-яких додаткових функцій чи налаштувань параметрів.

Глибоке навчання є дуже ефективним методом комп'ютерного зору. У більшості випадків для створення хороших алгоритмів глибокого навчання потрібен збір даних навчання з кількома мітками та оцінки, такі як рівень нейронної мережі, тип і кількість навчальних сеансів. Порівняно з попередніми типами машинного навчання, глибоке навчання легше та швидше розробляти та розгортати.

Більшість сучасних програм комп'ютерного зору використовують глибоке навчання. Глибоке навчання та глибокі нейронні мережі перемістилися з теоретичної сфери до практичного застосування завдяки доступності та розвитку апаратного забезпечення та ресурсів хмарних обчислень.

Комп'ютерний зір відіграє важливу роль у застосуванні розпізнавання облич, комп'ютерної технології, яка зіставляє обличчя та характеристики людей.

Алгоритми комп'ютерного зору виявляють риси обличчя на зображеннях і порівнюють їх із базою даних профілів обличчя. Пристрої користувача використовують розпізнавання обличчя для перевірки особи власника. Програми соціальних мереж використовують розпізнавання обличчя для пошуку та позначення користувачів. Правоохоронні органи використовують технологію розпізнавання обличчя, щоб ідентифікувати злочинців за відеопотоками.

1.4 Штучні нейронні мережі

ANN (Artificial Neural Network) Штучні нейронні мережі — це клас моделей машинного навчання, натхненний структурою та функціонуванням людського мозку. Вони складаються з взаємопов'язаних вузлів, тобто штучних нейронів, організованих у шари. Нейронні мережі є фундаментальною частиною глибокого навчання, підполя машинного навчання, яке зосереджується на використанні багаторівневих моделей (глибоких нейронних мереж) для вивчення ієрархічних представлень даних.

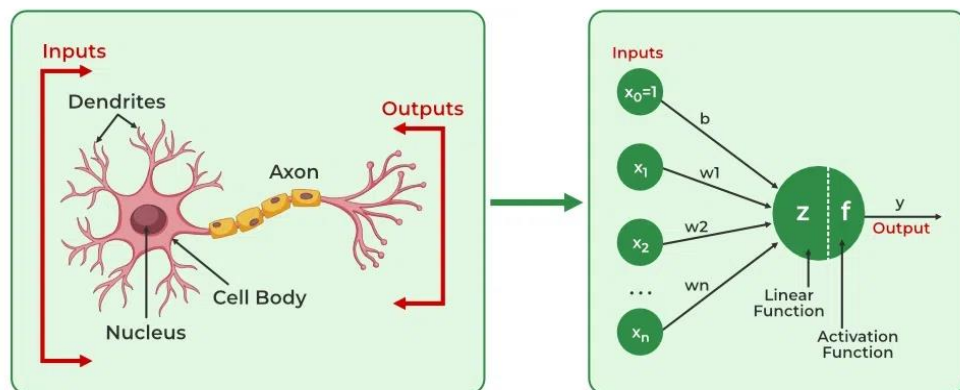


Рисунок 1.4 – Біологічні нейрони в штучні нейрони

Найпоширенішим завданням нейронних мереж є візуальне розпізнавання образів. Сьогодні існують мережі, які успішно розпізнають коди паперових і банківських карт, та ідентифікують об'єкти.

Розпізнавання зображень — це завдання ідентифікації зображення та класифікації його в одну з кількох визначених категорій. Таким чином, програмне забезпечення та програми для розпізнавання зображень можуть

використовувати візерунки, показані на зображенні, і відрізнити один об'єкт від іншого.

Розглянули ключові поняття, пов'язані з нейронними мережами.

Нейрони є основними одиницями нейронної мережі. Вони приймають вхідні дані, виконують обчислення та виробляють вихідні дані. У штучній нейронній мережі нейрони — це математичні функції, які обробляють вхідні дані. Нейронні мережі організовані в рівні: вхідний рівень, один або кілька прихованих шарів і вихідний рівень. Вхідний рівень отримує вихідні дані, а вихідний рівень виробляє остаточний прогноз або класифікацію. Приховані шари обробляють проміжні представлення.

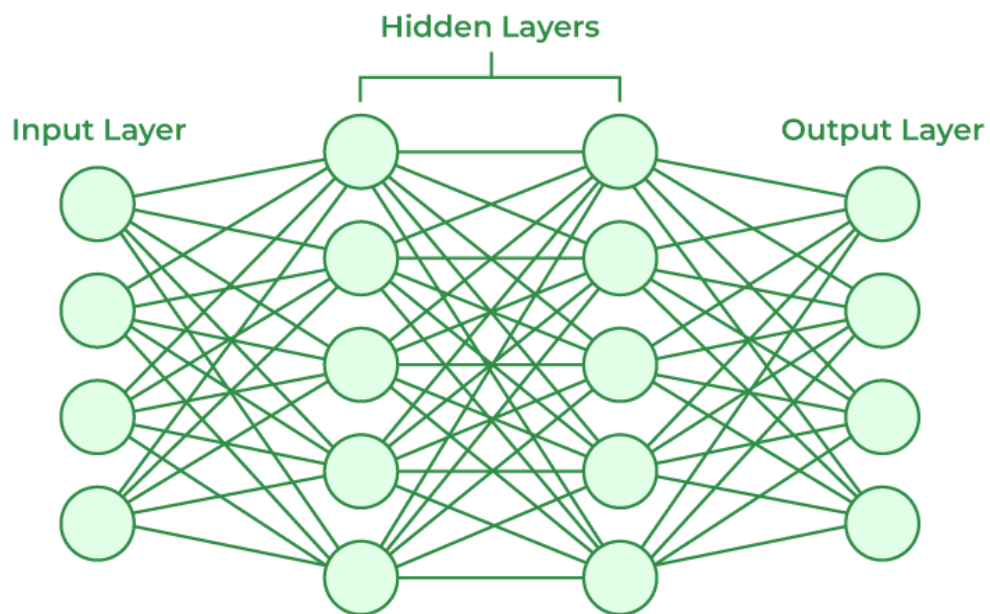


Рисунок 1.5 — Архітектура штучної нейронної мережі

Нейронні мережі прямого зв'язку (FNN). У FNN інформація поширюється в одному напрямку – від вхідного шару через приховані шари до вихідного. Це базова архітектура для таких завдань, як класифікація зображень і регресія.

Зворотне поширення. Зворотне розповсюдження – це контрольований алгоритм навчання, який використовується для навчання нейронних мереж. Це передбачає коригування ваг і зміщень на основі помилки (різниці між прогнозованим і фактичним виходом), щоб мінімізувати цю помилку.

Навчання нейронної мережі включає представлення її з позначеними даними і коригування параметрів моделі для мінімізації різниці між

прогнозованими та фактичними виходами. Навчання — це процес покращення продуктивності моделі з часом.

Глибокі нейронні мережі (DNN). DNN мають кілька прихованих рівнів, що дозволяє їм вивчати складні ієрархічні представлення даних. Глибоке навчання було особливо успішним у таких завданнях, як розпізнавання зображень і мови.

Згорткові нейронні мережі (CNN). CNN — це тип нейронної мережі, призначений для обробки сіткових даних, таких як зображення. Вони використовують згорткові шари для автоматичного вивчення просторових ієрархій об'єктів.

Рекурентна нейронна мережа (RNN). RNN розроблені для послідовних даних і мають з'єднання, які утворюють спрямовані петлі. Вони використовуються в таких завданнях, як обробка природної мови та аналіз часових рядів.

Нейронні мережі знайшли застосування в різних сферах, включаючи розпізнавання зображень і мови, обробку природної мови, охорону здоров'я, фінанси тощо. Успіх нейронних мереж пояснюється їхньою здатністю автоматично вивчати представлення з даних, що робить їх придатними для складних і багатовимірних завдань.

2 ОСОБЛИВОСТІ ОЦІНЮВАННЯ ОЗНАК ТА ПОБУДОВИ ВИРІШУЮЧИХ ПРАВИЛ В МЕТОДАХ РОЗПІЗНАВАННЯ СТАТІ ТА ВІКУ ЛЮДИНИ

2.1 Методи та алгоритми розпізнавання статі та віку людини

Алгоритми виявлення поверхні класифікуються за геометрією та текстурою. Методи на основі моделі можна розробити за допомогою статистичних засобів, таких як SVM [машина опорних векторів], LDA [лінійний дискримінантний аналіз], PCA [аналіз головних компонентів], метод ядра або перетворення спостереження. Методи геометричних елементів аналізують елементи внутрішньої поверхні та їх геометричне співвідношення.

Увага приділяється системі локальної диференціації. Ці методи представляють обличчя як лінійну підмножину його траєкторій зображення обличчя. Оскільки відстань від зони обличчя перпендикулярна до центру зображення, її можна легко перетворити на відстань за допомогою ймовірності.

Оскільки співвідношення елементів і швидкість обробки всієї поверхні не зменшуються, багато дослідників дотримувалися цього підходу і намагалися знайти найкращі властивості. Спробували комбінацію очей, рухів. Деякі неявні методи проектування Маркова також потрапляють у цю категорію та дуже популярні в обробці компонентів і виявленні поверхні.

Режим на основі обличчя відображає обличчя в кількох кадрах. Зображення — це вектор багатьох змінних. Цей метод часто використовується для пошуку об'єкта в частині зображення. Зразки зображень у порівнянні з навчальними наборами. Однак метод моделювання шукає модель вище. Нова модель відноситься до параметрів форми та структури, які використовуються для визначення зображення.

Підходи, засновані на зовнішньому вигляді, можуть бути лінійними або нелінійними. Статистично зразки виступають як компоненти. Когнітивна функція в дискримінаційній арифметичній ймовірності. Кожне зображення представлено об'єктом. Тому мета полягає у виборі та використанні відповідних статистичних інструментів для висновків та аналізу.

Для аналізу місцезнаходження доступні кілька статистичних інструментів. Ці інструменти аналізу можна застосовувати до двох або більше груп систем класифікації. До них належать аналіз основних характеристик [APC], дискретне

косинусне перетворення [DCT], лінійний дискримінантний аналіз (LDA), незалежний аналіз компонентів [ICA].

Одним із більш застосовуваних статистичних методів є аналіз основних характеристик. Математичний устрій допомагає зменшити розмірність, створюючи велику частину багатовимірних даних.

DCT являє собою набір точок даних про суму функцій косинусів для різних частот коливань. Дискретне косинусне перетворення базується на перетворенні Фур'є та може застосовуватися для ефективного перетворення зображень, стиснення контрасту та зменшення розміру.

LDA використовується для збору лінійних об'єктів, зберігаючи розділення між класами. На противагу від PCA, LDA старасться змоделювати варіацію рівня. LDA відрізняється для кожного шару з кількома векторами проєкції.

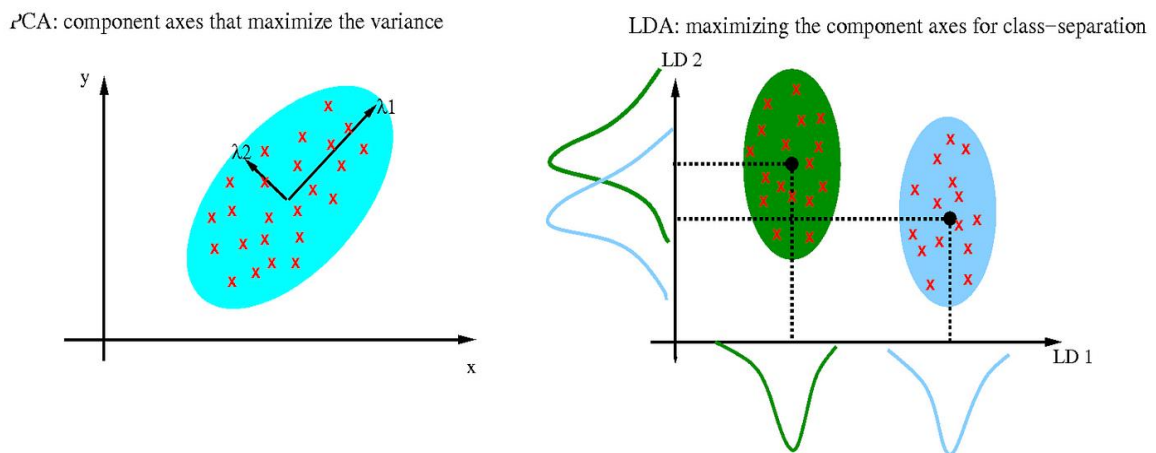


Рисунок 2.1 — Порівняння LDA та PCA

Використання базисних функцій для реалізації нелінійного PCA. Основна техніка полягає у використанні нелінійного відображення вхідних даних і подальшому розв'язанні відстані PCA у підпросторі.

LPP це хороший вибір для PCA для підтримки ландшафту та дизайну. Алгоритми розпізнавання шаблонів зазвичай шукають шаблон або його найближчих сусідів. Це може пришвидшити процес використання сайту, який підтримує стандарт LPP.

ICA намагається трансформувати дані в лінійну комбінацію статистично незалежних точок даних. Тому, метою є не дискретне представлення, а нейтральне представлення зображення. ICA є альтернативою PCA, яка

забезпечує кращу візуалізацію даних. Це міра дискримінантної достовірності, яку можна застосувати для покращення PCA.

Метод DCV збільшує подібність і усуває відмінності між об'єктами одного класу, тому всякий клас може фігурувати представленим нормальним вектором матриць просторового розподілу.

У випадку невідомого експерименту вектор кореляції обчислюється та налаштовується на клас із найближчим нормальним вектором. У розпізнаванні обличчя іноді використовуються нормалізовані різниці вектори або збагачені опорні векторні машини.

Подібно до LLE вводяться (NPP) і (ONPP). Цей метод зберігає внутрішню структуру шаблону. Вони використовують кероване даними масштабування для вирішення невеликих проблем, пов'язаних із представленням геометрії в просторі. ONPP робить карту стабільною та вирішує загальну проблему витрат. NPP потрібна для вирішення загальної проблеми запису характерних значень у прогнозних даних.

Багатолінійне розширення методу LDA. Він відрізняється від методу безперервної апроксимації тим, що він використовує прямий диференціальний аналіз даних фізичної деформації для збереження приблизної структури функціонального тензора. Іншим підходом є керована NPP (MNPP). Багаторядкові методи працюють безпосередньо з тензорними даними, а не з векторами чи масивами, і вирішують проблеми з тензорними представленнями для вилучення та візуалізації кількох функцій. Метод MNPP знаходить кілька підключених підполів, розгортаючи тензори уздовж шляхів різної довжини. Розширення часового інтервалу визначає кількість підблоків, які поступають до МАЕС.

Данні методи можуть неправильно відображати обличчя, якщо є суттєві відмінності в яскравості, виразі та інших факторах. Таким чином, застосування нелінійних методів не забезпечує значного покращення стосовно з лінійними методами LLP, LLE і LBP, які забезпечують простий і ефективний спосіб опису безперервної зміни даних на високому рівні. Методи DCV і SVM використовують методи підпростору. Збереження внутрішньої структури у зразках є областю методів ONPP і NPP. Проблема полягає в тому, що неясно, як встановити розмір діапазону або як визначити найкраще значення для нього.

2.2 Вейвлети Габора

Зі стрімким розвитком штучного інтелекту люди хочуть знати більше про риси обличчя, розпізнавати обличчя на фотографіях. Оцінка віку дуже важлива в реальному житті, наприклад у комп'ютерних мережах, зборі даних, бізнес-аналізі.

Вікові зміни проявляються в різних формах, таких як шкіра обличчя, м'які тканини та кістки. Використовуючи методи виділення обличчя, вік людського обличчя можна оцінити за допомогою створених людьми алгоритмів або методів глибокого навчання.

Вейвлети Габора мають такі важливі оптичні властивості, як вибірковість просторової частоти та просторова роздільна здатність. Ця технологія підходить для багатьох біометричних програм.

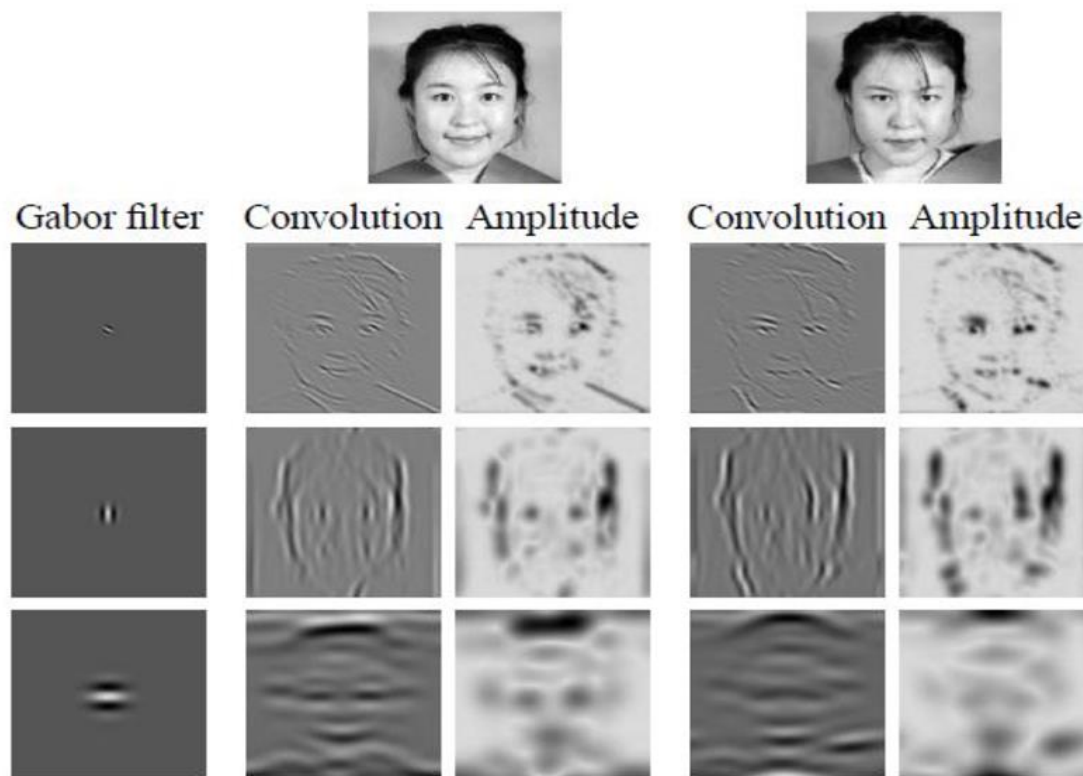


Рисунок 2.2 – Вейвлет Габора

Цей алгоритм забезпечує розпізнавання обличчя за допомогою багат шарового персептрона з використанням попереднього алгоритму. Перший – попередня обробка. Кожне зображення налаштовується на контрастність і чіткість. Кожне зображення використовує фільтр Габора. Фільтр Габора має п'ять карт і три просторові предиктори, отже, він має 15 Габорів.

Вейвлети Габора широко використовуються дослідниками розпізнавання облич, і риси Габора вважаються найкращим відображенням виразу обличчя на ранніх стадіях виявлення. Крім того, він чутливий до змін світла.

Порівняння методу GMTR з гамма-частотною моделлю Габора для класифікації форми. GPTR характеризується простою моделлю щільності Гауса розподілу часу Габора. Це дозволяє безперервно проводити статистичний аналіз моделі.

Вейвлет Габора використовується на основі зображення обличчя у фіксованому положенні. Кожна область сегментованого зображення обличчя визначається як квадратурна вихідна поверхня Габора, а потім відображається на підпростір за допомогою перетворення Кархунена-Ло. Результат кожного пошуку, вимірний генетичним алгоритмом (GA), використовувався для навчання вікна Пергана. Класифікатор використовує ваговий параметр для керування процесом класифікації.

Лапласівський підхід до вивчення облич на основі ядра для розпізнавання облич у кластеризації та класифікаторах Габора. Функція Габора кожного каналу використовується як нова модель класу для реалізації комбінованого методу класифікації. Цей метод важливий для вдосконалення використання набутих знань.

Основна перевага методу Габора полягає в тому, що оскільки зображення обличчя поєднується з набором фільтрів Габора, простір ознак Габора дуже великий.

Щоб вирішити цю проблему, найважливіші функції Gabor вибираються за допомогою алгоритму Adaboost і алгоритму генетичної ентропії (GA).

Однак вибір найефективнішого методу серед кількох функцій Габора займає багато часу. Крім того, продуктивність інструментів Gabor потребує обчислень, що робить ці функції непридатними для програм реального часу.

Перевага методу Габора полягає в тому, що розмір області ознаки Габора більший за розмір ознаки Габора, оскільки зображення обличчя створює фільтр Габора. Цей метод потребує обчислень і не може використовуватися в програмах реального часу.

2.3 Дескриптор функції

Під час обробки зображень дескриптор об'єкта представляє область зображення або ключову точку, яка збирає важливу інформацію про об'єкт зображення. Розглянемо один з алгоритмів обробки зображень, а саме дескриптор процесу.

Обробка зображень — це обробка комп'ютерного зору, яка стосується обробки та аналізу цифрових зображень. Він включає методи й алгоритми для покращення обробки, перетворення та вилучення інформації із зображень. Обробка зображень використовується в різних програмах, включаючи медичне зображення, комп'ютерне бачення, дистанційне зондування тощо.

Різноманітні алгоритми та методи використовуються для оптимізації, оптимізації та вилучення корисних функцій або даних, які автоматизують рішення та процеси. Це потужний і універсальний інструмент, який обробляє та аналізує цифрові зображення за допомогою бібліотеки комп'ютерного бачення з відкритим кодом (OpenCV). Для цього існують основні алгоритми кадрування, фільтрації, зміни розміру, а також розширені методи виявлення, виявлення об'єктів.

Дескриптор — це спосіб графічного представлення події з унікальними чи цікавими особливостями. Як правило, це набори чисел, які описують форму, форму або поведінку об'єкта в середовищі, яке ви хочете мати на зображенні. Дані про діяльність корисні для порівняння зображень і відображення контрастів, оскільки алгоритми можуть виявляти схожі області чи об'єкти на різних зображеннях. Деякі програми для цього включають ідентифікацію та зіставлення областей зображення, які можна використовувати для кластеризації зображень, відстеження об'єктів тощо. Необхідний для таких операцій.

Точка визначається як область або частина зображення, яка має певний колір або перетин, перетин кількох сегментів країв або як швидка зміна напрямку країв. Ці проблеми включають здатність залишатися стабільною, незважаючи на зміни в масштабі, обертанні та експозиції. Перш за все, важливо точно і надійно розрахувати цікаві місця.

Векторний об'єкт — це числове представлення опису об'єкта у векторній формі з одним або кількома вимірами. Це одновимірний вектор, який відображає інформацію про описи об'єктів у багатовимірний простір ознак. Збирає різноманітну інформацію, пов'язану з різними предметами. Комбінуючи ці часткові вектори форми, ми можемо побудувати простір фігури. Рівень точності, необхідний у векторі ознак, залежить від конкретного аспекту об'єкта, який ми

хочемо вивчити або уявити. Ця деталь визначається об'єктом, який ми вимірюємо.

Алгоритми прогнозування продуктивності включають SIFT, SURF, HOG, LBP та інші.

2.4 Тривимірне розпізнавання облич

Тривимірне розпізнавання облич (3D) — це новий спосіб розпізнавання облич. Він використовує 3D-обробку людських облич для виявлення рис у реальному часі. На відміну від традиційних 2D-методів, 3D-розпізнавання обличчя відтворює обличчя в 3D, забезпечуючи новий рівень точності.

На нашому обличчі є різні зморшки, перемички і западини. Хоча 2D-зображення можуть захоплювати риси обличчя, їм бракує глибинних деталей.

Одним із ключів до успіху системи 3D-розпізнавання обличчя є використання даних. Дослідники використовують 3D-сканери обличчя та великі бази даних для створення детальних 3D-моделей облич. Моделі можуть аналізувати шаблони та фіксувати складні вирази обличчя, такі як умови освітлення, жести та вирази обличчя.

Головною причиною, чому 3D-розпізнавання облич краще, ніж 2D, є його здатність розпізнавати різні візерунки. Він фіксує та досліджує глибину людського обличчя. 2D-методи покладаються лише на плоскі зображення, тоді як 3D-розпізнавання обличчя використовує 3D-модель обличчя, яка дозволяє використовувати тривимірну геометрію людського обличчя. Він забезпечує надійний захист від таких небезпек, як зміна освітлення, різні форми обличчя та різні кути нахилу обличчя.

Ключем до цього процесу є знання ШІ. Складні алгоритми виконують аналіз шаблонів на 3D-даних обличчя, забезпечуючи надійне розпізнавання, незважаючи на зміни місця розташування та умов освітлення. Професійний алгоритм підвищує надійність системи та робить її стійкою до факторів зовнішнього середовища.

Процес 3D-розпізнавання обличчя починається зі збору інформації про 3D-поверхню та аналізу таких властивостей, як глибина та кривизна. Ця техніка виявляє та витягує живі обличчя з кадрів фото чи відео. Йдеться не лише про розпізнавання очей і губ, та про розуміння глибини очних западин або губ.

Кожна деталь від скул до брів записується та перетворюється на електронний підпис. За допомогою штучного інтелекту дані, отримані з великих баз даних.

Перехід від 2D до 3D розпізнавання обличчя значно змінив зовнішній вигляд. На відміну від двовимірних моделей, які покладаються лише на візуальні функції, нова 3D-реконструкція обличчя використовує для розпізнавання складну геометрію обличчя. Цей метод має багато переваг:

Завдяки 3D аналізу розпізнавання обличчя не змінюється при зміні освітлення, форми обличчя та розміру голови. Це допомагає зменшити ризик шахрайства та зробити процес підтвердження особи більш безпечним і точним.

Через багатовимірну природу 3D-функцій системами зображень і відео важко маніпулювати.

Поєднання тривимірної геометрії покращує технічну стійкість навіть у складних умовах.

Розпізнавання облич є потужним інструментом, але він не надійний. Завдяки сучасним кібератакам із використанням високоякісних зображень і відео іноді можна обдурити систему. Виявлення подій дозволяє уникнути цього, забезпечуючи взаємодію системи з реальними людьми. Живе 3D-розпізнавання додає до облич геометрію. Він відстежує трафік у режимі реального часу та гарантує, що це не лише зображення чи відео.

2.5 Глибокі нейронні мережі

Глибинні нейронні мережі зробили революцію в сучасному штучному інтелекті. Останні наукові досягнення в області глибокого навчання та нейронних мереж значно покращили якість проектів ШІ.

Ці глибокі нейронні мережі допомагають розробникам досягати стабільних і ефективних результатів. Тому вони замінюють деякі традиційні методи машинного навчання.

Штучні нейронні мережі (ANN), також відомі як традиційні нейронні мережі, призначені для виконання невеликих завдань за допомогою простих мережевих систем. Штучні нейронні мережі в основному базуються на біологічних нейронних мережах. Це набір шарів, які виконують певне завдання. Кожен рівень має кілька вузлів, які працюють разом.

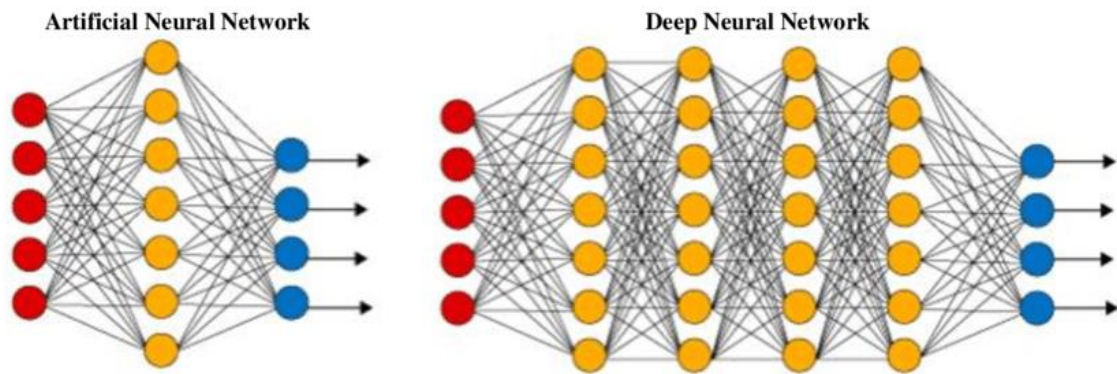


Рисунок 2.3 – Порівняння ANN та DNN

Ці методи зазвичай мають вхідний рівень, один або два прихованих шари та вихідний рівень. Хоча ці мережі можуть вирішувати прості математичні та обчислювальні завдання за допомогою базових структур портів і відповідних таблиць відповідності, вони мають труднощі з вирішенням складних завдань, таких як обробка зображень, комп'ютерне бачення та обробка природної мови.

Щоб вирішити цю проблему, ми використовуємо глибокі нейронні мережі, які зазвичай мають складну структуру прихованого шару, що складається з різних шарів, таких як згорткові шари, шари зберігання, товсті шари та інші унікальні шари. Ці додаткові рівні допомагають моделі краще зрозуміти проблему та забезпечують кращі рішення для складних проектів. Глибокі нейронні мережі мають більше рівнів (глибших), ніж ANN, і кожен рівень ускладнює модель, дозволяючи моделі ретельно обробляти вхідні дані для створення найкращого можливого рішення.

Глибокі нейронні мережі дуже популярні в багатьох проектах глибокого навчання завдяки своїй високій ефективності. Правильно розроблену глибоку нейронну мережу можна навчити отримувати бажані результати з високою точністю. Він ефективний у всіх сферах глибокого навчання, включаючи комп'ютерний зір, обробку природної мови та трансферне навчання.

Яскравим прикладом важливості глибоких нейронних мереж є розпізнавання об'єктів за допомогою таких моделей, як YOLO (You Only See One), BERT (Bilateral Representation Coding) і моделі передачі VGG. -19 RESNET.

Побудова нейронних мереж з нуля допомагає програмістам зрозуміти концепції та вирішити невеликі завдання під час програмування цих мереж. Однак створення цих мереж з нуля потребує багато часу та зусиль. Щоб спростити глибоке навчання, є багато інструментів і бібліотек для створення

глибоких моделей нейронних мереж, які можуть вирішувати складні проблеми лише за допомогою кількох рядків коду.

Поширені бібліотеки глибокого навчання та інструменти, які використовуються для створення глибоких нейронних мереж, включають TensorFlow, Keras і PyTorch. З випуском TensorFlow 2.0 бібліотеки Keras і TensorFlow були об'єднані в одну мову. Ця інтеграція дозволяє користувачам створювати складні нейронні мережі та вдосконалене кодування за допомогою Keras у мережах TensorFlow.

Бібліотека PyTorch — ще одне популярне середовище машинного навчання, яке дозволяє користувачам створювати цікаві дослідницькі проекти.

Хоча PyTorch не має компонентів візуалізації, він пропонує компактне та високопродуктивне рішення для моделювання глибокої нейронної мережі зі швидкою та простою конфігурацією GPU.

Платформа TensorFlow надає розробникам різноманітні потужні інструменти візуалізації для глибокого навчання. Панель Tensor є чудовим вибором для відображення, аналізу та інтерпретації даних проекту.

Для виконання всіх завдань машинного навчання потрібні спеціалізовані глибокі нейронні мережі. Двома поширеними підходами до глибокого навчання є згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN). Згорткові нейронні мережі дуже корисні для обробки зображень і комп'ютерних завдань.

Замість виконання звичайних матричних операцій над прихованими шарами цієї глибокої нейронної мережі ми виконуємо операції згортки. Це дозволяє мережі використовувати масштабований підхід для забезпечення максимальної ефективності та реальних результатів. Для завдань класифікації зображень і розпізнавання об'єктів необхідно розрахувати кількість даних на зображенні в моделі. Ця нейронна мережа допомагає ефективно вирішувати ці проблеми.

Нейронні мережі зі зворотним зв'язком часто використовуються для підвищення продуктивності обробки природної мови та семантичних завдань. Відома форма цього RNN, довготривала пам'ять (LSTM), часто використовується в різних завданнях, таких як машинний переклад, класифікація тексту та розпізнавання мовлення.

Ця мережа отримує відповідну інформацію з кожної початкової комірки та передає її до наступної комірки, зберігаючи відповідну інформацію для покращення продуктивності моделі.

Дані є однією з найважливіших вимог для глибокого навчання. Ключем до створення більш точних моделей є дані. У деяких випадках глибокі нейронні мережі часто вимагають великих обсягів даних для забезпечення оптимальної продуктивності та запобігання перегріву. Вимоги до даних для завдань розпізнавання об'єктів. Щоб модель з високою точністю ідентифікувала кожен об'єкт, може знадобитися більше даних. Хоча методи розширення даних корисні для швидкого вирішення деяких із цих проблем, проекти глибокого навчання повинні враховувати потреби в даних.

Графічні процесори потрібні для компіляції та навчання моделей складним завданням. Моделі часто краще навчаються на GPU або TPU, ніж на CPU.

2.6 Згорткові нейронні мережі

Від інших нейронних мереж згорткові нейронні мережі відрізняються від інших нейронних мереж високою продуктивністю при обробці сигналів зображення, звуку або тексту. Є три типи ступенів:

- Згортковий рівень
- Об'єднуючий шар
- Повнозв'язковий шар (FC).

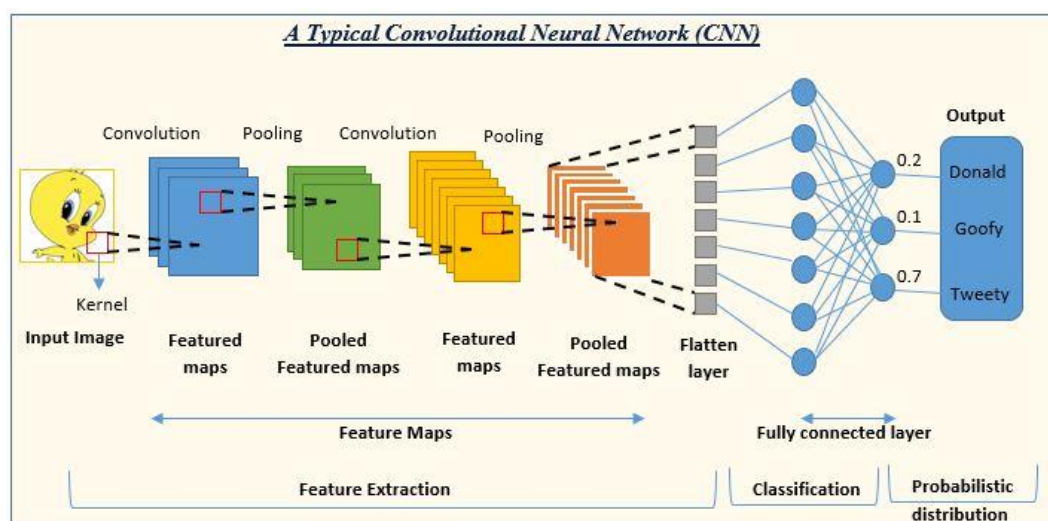


Рисунок 2.4 – Зображення згорткової нейронної мережі

Першим шаром згорткової мережі є згортковий рівень. Хоча шари можна додати або відняти пізніше, остаточний процес інтеграції є останнім рівнем. На кожному рівні складності датчик записує різні ділянки зображення. Верхні шари зосереджені на таких важливих елементах, як колір і рамка. Коли дані зображення проходять через датчик, датчик починає виявляти великі об'єкти або розміри об'єктів, поки не знайде ціль.

Процес обчислення є основною частиною генератора CNN і саме там виконується більшість обчислень. Для цього потрібно кілька речей: вхідні дані, фільтри та таблиця. Це означає, що вхідний файл має три виміри (висота, ширина та глибина), що відповідають зображенню RGB. У нас також є інструмент, який також називають ядром або фільтром, який проходить через датчик і щось перевіряє.

Активний символ — це двовимірна (2D) вага, яка відображає частину зображення. Хоча вони можуть відрізнитися за розміром, розмір фільтра зазвичай є матрицею 3x3. Це також визначає розмір приймача. Фільтр було застосовано до зображення та розраховано точку входу та розташування фільтра.

Вага залишається незмінною, коли ви рухаєтесь навколо зображення. Його також називають параметром продуктивності. Деякі параметри, такі як вага, регулюються під час виконання вправ. Однак є три гіперпараметри, що діють на розмір виведення, і їх необхідно визначити перед навчанням нейронної мережі. Це включає:

- Чисельність фільтрів впливає на глибину виробництва.
- Фазова відстань або кількість початково рухомих пікселів у полі введення. Кроки, що складаються з двох або більше кроків, можуть бути невеликими, але більші кроки дають менші результати.
- Якщо фільтри не відповідають вхідному зображенню, хеш не застосовується. Встановлює всі елементи у зовнішньому масиві на нуль і видає вихід, більший або рівний цьому значенню.

Активний хеш - хешу немає. Якщо критерії не збігаються, остаточну відповідь буде відхилено.

Рівне заповнення - це заповнення гарантує, що поведінка виводу буде такою ж, як і введення.

Повне заповнення - даний тип заповнення збільшує розмір виведення шляхом додавання нулів до поля введення.

Після кожної операції CNN застосовує перетворення ReLU (блок лінійного перетворення) і представляє зміни в моделі.

Згортковий рівень може супроводжуватися іншими методами перетворення. У цьому випадку дизайн може бути унікальним. Припустімо, ми намагаємося визначити, чи містить зображення цикл. Шар перетворює зображення в числове значення, яке дозволяє нейронній мережі інтерпретувати та витягувати відповідні шаблони.

Об'єднання шарів, також відоме як скорочення, зменшує розмірність шляхом зменшення чисельності вхідних параметрів. Як і шар згортки, функція агрегації застосовує фільтр до всього вхідного сигналу, але різниця полягає в тому, що фільтр не є зваженим. Замість цього ядро застосовує функцію суми до значень в полі прийому, таким чином заповнюючи вихідний масив. Розрізняють два основних типи реакцій.

Максимальний обсяг пам'яті - під час проходження фільтром вхідних даних він вибирає найцінніші пікселі для надсилання у вихідний масив. Крім того, цей метод ширше використовується, ніж центральне об'єднання.

MeanPool: коли вхідні дані фільтруються, приймач визначає середнє значення поля, яке буде надіслано до вихідної таблиці.

Більша частина інформації втрачається на рівні пулу, CNN має багато переваг. Це допомагає зменшити складність, підвищити ефективність і уникнути надмірного ризику.

Дані пікселів вхідного відображення не пов'язані прямо з вихідним шаром частково відображеного шару. Однак на повністю підключеному рівні кожен вузол вихідного рівня безпосередньо з'єднаний з вузлом попереднього рівня.

Даний шар виконує завдання класифікації на основі ознак, отриманих із попередніх шарів і різних фільтрів. Тим часом як рівні згортки та агрегації переважно застосовують функцію ReLu, рівень FC зазвичай застосовує функцію активації для генерування ймовірностей від 0 до 1 для правильної класифікації вхідних даних.

2.7 Рекурентні нейронні мережі

Повторювані нейронні мережі (RNN) — є видом штучної нейронної мережі, призначеної для даних послідовності. Вони особливо ефективні в задачах, де вхідні або вихідні дані мають послідовний характер, наприклад

прогнозування часових рядів, обробка природної мови та розпізнавання мовлення. На відміну від нейронних мереж прямого зв'язку, RNN мають з'єднання, які утворюють спрямовані цикли, що дозволяє їм підтримувати прихований стан, який фіксує інформацію про попередні входні дані в послідовності.

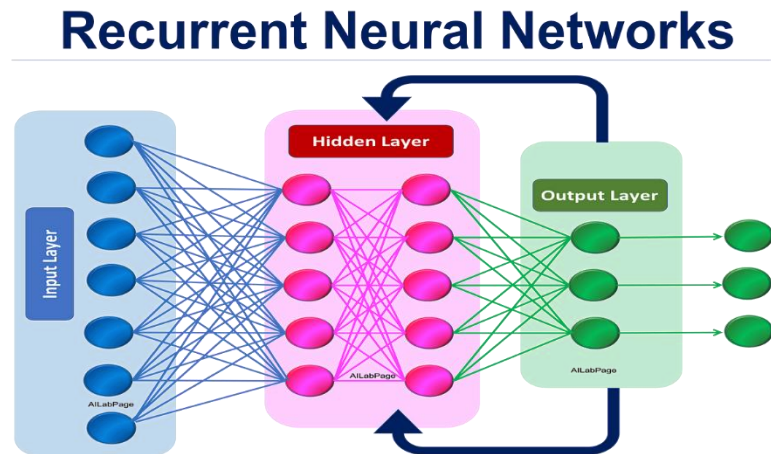


Рисунок 2.5 – Рекурентна нейронна мережа

Ключові функції та компоненти рекурентних нейронних мереж:

Визначальною характеристикою RNN є наявність повторюваних зв'язків, які дозволяють інформації зберігатися протягом різних часових кроків у послідовності.

RNN зберігають прихований стан, який розвивається з часом, коли мережа обробляє кожен елемент у входній послідовності. Цей прихований стан служить пам'яттю про минулі входи.

Навчання RNN може бути складним через проблему зникаючого градієнта. Градієнти можуть стати дуже малими під час зворотного поширення в часі, що ускладнює для мережі вивчення довгострокових залежностей.

Щоб вирішити проблему зникнення градієнта, були введені більш просунуті архітектури RNN, такі як довготривала короткочасна пам'ять (LSTM) і стробований рекурентний блок (GRU). Ці архітектури включають механізми для вибіркового збереження або відкидання інформації в прихованому стані.

RNN використовуються в різних програмах.

Обробка природної мови (NLP): моделювання мови, машинний переклад, аналіз настроїв.

Розпізнавання мовлення: перетворення розмовної мови на текст.

Прогноз часових рядів: прогнозування цін на акції, прогноз погоди.

Аналіз відео: розпізнавання дій, відстеження об'єктів.

RNN навчаються за допомогою зворотного поширення в часі (BPTT). Градієнти обчислюються за кілька часових кроків, що дозволяє мережі вивчати залежності в послідовних даних.

Незважаючи на свою ефективність, RNN мають обмеження. Їм може бути важко вловити довготривалі залежності в послідовностях, а навчання може бути дорогим з обчислювальної точки зору.

Двонаправлені RNN обробляють вхідну послідовність як у прямому, так і в зворотному напрямках, захоплюючи інформацію з минулих і майбутніх елементів послідовності. Було введено механізми звернення уваги, щоб покращити обробку довгих послідовностей. Вони дозволяють мережі зосередитися на певних частинах вхідної послідовності під час прогнозування.

Методи проектування RNN:

Двонаправлена рекурентна нейронна мережа (BRNN). Односторонні RNN можуть використовувати лише наявні вхідні дані для передбачення поточного стану, тим часом як двонаправлені RNN виймають майбутню інформацію для збільшення точності.

Довготривала пам'ять (LSTM) це популярний алгоритм RNN, призначений для вирішення проблеми зникнення градієнта. Такі моделі RNN не можуть точно прогнозувати поточну ситуацію, якщо не оновити попередню частину, яка має труднощі з поточним прогнозом. У LSTM три комірки — вхідний та вихідний шлюзи і забутий шлюз — контролюють рівень інформації, необхідної для прогнозування результату.

Gated Recurrent Units (GRU) цей RNN схожий на LSTM, тому він теж розв'язує проблему короткотривалої пам'яті, властиву моделі RNN. Замість інформації про правила на рівні комірки він застосовує прихований шар і має два шлюза замість трьох — шлюзи оновлення та скидання. Подібно до шлюзу в LSTM, ви можете оновити або знизити шлюз, щоб контролювати швидкість передачі даних і зберігання.

Побудова рекурентної нейронної мережі (RNN) для визначення статі та віку людини за зображенням є багатоетапним процесом. На практиці згорткові нейронні мережі (CNN) часто використовуються для завдань, пов'язаних із зображеннями, а RNN частіше застосовуються до послідовних даних. Однак ви

можете комбінувати обидві архітектури для обробки як просторових, так і послідовних аспектів даних.

Схема кроків, які виконуються:

Збирають набір даних із зображень, позначених інформацією про статі і вік.

Попередньо обробляють зображення, включаючи зміну розміру, нормалізацію та збільшення.

Використовують CNN для вилучення функцій із зображень. Популярні архітектури включають VGG, ResNet або EfficientNet.

Після CNN можливо використовувати RNN або комбінацію повністю пов'язаних рівнів для обробки послідовної інформації.

Витягують функції з CNN і змінюють їх у формат, придатний для введення в RNN.

Створюють RNN для обробки послідовної інформації. Можливо використовувати LSTM або GRU (Gated Recurrent Unit).

Об'єднують функції CNN із послідовною інформацією з RNN.

Створюють вихідні шари для передбачення статі та віку. Для статі підходить бінарна класифікація (чоловічий/жіночий). Для віку ви можете мати кілька вихідних вузлів, що відповідають різним віковим групам.

Визначають відповідні функції втрат для прогнозів статі та віку. Загальні варіанти включають двійкову крос-ентропію для статі та середню квадратичну помилку або категоричну крос-ентропію для віку.

Для навчання моделі використовують алгоритми оптимізації.

Розділити набір даних на набори для навчання та перевірки.

Навчити модель на навчальному наборі, відстежуючи продуктивність на перевірочному наборі.

Оцінити модель на окремому наборі тестів, щоб оцінити її ефективність узагальнення.

Налаштувати гіперпараметри, архітектуру моделі або стратегії навчання на основі оцінок продуктивності.

Отримавши задовільний результат від продуктивності моделі, розгорніть її для реальних прогнозів.

Успіх моделі залежить від якості та репрезентативності набору даних, а також від ретельного проектування та налаштування архітектури нейронної мережі.

2.8 Висновок до розділу

В данному розділі описано основи систем розпізнавання облич, включаючи розпізнавання облич, виявлення, вирівнювання та визначення функцій, а також зосереджено на технічних проблемах створення систем ідентифікації. Особливу увагу було приділено важливості успішного рішення. У цьому розділі описано методи виявлення облич, їх застосування та потенційні застосування у сфері розпізнавання облич.

Розпізнавання облич є проблемою комп'ютерного зору. Останнім часом він широко використовується і популярний у багатьох сферах. Незважаючи на значні дослідження в цій галузі, системи розпізнавання облич недостатньо досконалі, щоб ідеально працювати в усіх практичних ситуаціях. В данному розділі наведено огляд методів виявлення та застосування технологій для визначення статі та віку людини.

3 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ СТАТІ ТА ВІКУ ЛЮДИНИ

3.1 Огляд програмного забезпечення

Наш зір дозволяє нам швидко бачити речі. Але комп'ютери бачать нас інакше. Він використовує багато даних і інструментів (камери, датчики). Подібно до того, як людське око допомагає нам бачити навколишній світ і реагувати на нього, комп'ютерний зір може розуміти, трансформувати та реагувати на те, що вони бачать.

Комп'ютерний зір сьогодні має багато застосувань у безпеці, господарстві, медицині та багатьох інших галузях. Тому потреба в комп'ютерних бібліотеках і ресурсах зростає. Існує багато програм для зору, бібліотек розпізнавання зображень і розпізнавання облич.

Бібліотеки комп'ютерного бачення зберігають документи та сценарії, які використовуються для створення та керування комп'ютерними системами. Існує багато бібліотек комп'ютерного зору, адаптованих до конкретних потреб.

Список найкращих бібліотек комп'ютерного зору:

OpenCV — найпопулярніша та найвідоміша бібліотека комп'ютерного зору, орієнтована на візуалізацію в реальному часі. Це кросплатформна бібліотека, яка підтримує операційні системи Windows, Linux, Android і macOS і доступна на Python, Java і C++. OpenCV є фреймворком Python і використовує модель CUDA для GPU. Спочатку розроблений компанією Intel, він доступний у вільному доступі за ліцензією BSD Open Source License. Крім того, багато прикладів можна перетворити на приклади TensorFlow.

Scikit-Image — вважається найпростішою бібліотекою Python, «розширенням» Scikit-Learn. Це одна з найкращих технік для контрольованого та неконтрольованого машинного навчання. Scikit-Learn — це пакет Python для створення графіків і роботи з масивами NumPy як об'єктами графіків. Scikit-Image побудовано на Python і використовує модуль Scikit-Cuda, тому він безкоштовний і необмежений.

Keras — це бібліотека Python з відкритим вихідним кодом, яка дуже корисна для початківців, оскільки дозволяє швидко створювати моделі нейронних мереж і забезпечує базову підтримку. Це інструмент моделювання,

який дозволяє інженерам комп'ютерного зору швидко візуалізувати шаблони, практики реального світу та процеси мислення. Він використовує TensorFlow.

TensorFlow — спрямований на спрощення процесу створення моделей ШІ. Він має такі стандартні рішення, як TensorFlow.js, бібліотека JavaScript, Node.js, TensorFlow Lite, бібліотеки для розгортання мобільних моделей і моделей пристроїв.

Pillow — це повна бібліотека для мови програмування Python. Доступна для Windows, Mac OS X і Linux. Бібліотека зображень Python надає можливості роботи із зображеннями для Python і оптимізована для швидкої пам'яті. Цей інструмент широко використовується для читання та збереження зображень різними способами та включає базове редагування зображень.

TorchVision — внутрішня бібліотека PyTorch містить найпопулярніші конвертери відео для комп'ютерного зору. Також включає збір даних і обчислювальне моделювання нейронних мереж. Однією з головних цілей TorchVision є забезпечення природного способу використання функцій NumPy і PyTorch, і навпаки, для обробки зображень комп'ютерного зору. Цей пакет містить базові функції для великих баз даних, моделювання та комп'ютерного зору. TorchVision написаний на Python і може використовуватися на Python і C++.

OpenVino — це інструмент комп'ютерного зору, який можна використовувати для розробки програм, що імітують людський зір. OpenVino — це інструмент встановлення та налаштування, тому потрібне навчання моделі.

MMCV — це плагін PyTorch, який забезпечує маніпуляції та обертання зображень/відео, зображення та текст, а також фреймворки CNN. Він підтримує такі платформи, як Linux, Windows і macOS, що робить його корисним інструментом для дослідників комп'ютерного зору. Він працює з Python, C++, CUDA і є оболонкою Python.

PyTorch — це бібліотека машинного навчання Python з відкритим вихідним кодом. Використовує високопродуктивні комп'ютерні моделі, які дозволяють гнучко будувати складні конструкції. Pytorch використовує такі концепції Python, як класи, структури та умовні цикли, і сумісний із C++.

3.2 Мова програмування Python

Python — це гнучка мова, яку можна використовувати для різноманітних програм. Причини використання Python включають аналітику, веб-розробку, машинне навчання та DevOps. Python можна використовувати в інформатиці, ІІІ, дизайні, кодуванні, програмуванні.

Поєднання легкої для вивчення мови та великої бібліотеки Python робить його популярним вибором як для початківців, так і для досвідчених розробників. Його прості, але потужні функції допомагають вирішувати складні проблеми, що робить його популярною мовою програмування, створює потужні веб-сайти, проводить дослідження та візуалізацію.

Python також використовується для розробки ігор, мобільних додатків, комп'ютерного програмного забезпечення та настільних програм.

Простота використання, ефективність і велика спільнота розробників роблять Python ідеальною мовою для компаній, які хочуть швидко й ефективно розробляти програмне забезпечення.

Як безкоштовне та відкрите поле є великою та активною спільнотою. Це означає, що існує багато ресурсів, бібліотек та інструментів, які підтримують його використання. Крім того, стандартна бібліотека Python велика і пропонує багато вбудованих функцій.

Це стандартна мова програмування, яку можна використовувати в різноманітних додатках, від веб-розробки та мобільної розробки до аналітики, обробки даних, штучного інтелекту та машинного навчання.

Акцент Python на простоті та легкості використання означає, що розробники можуть стати більш досвідченими з меншим кодом. Це допомагає скоротити час і витрати на розробку, що робить його економічно ефективним рішенням для бізнесу.

Python сумісний з багатьма операційними системами, включаючи Windows, Linux і MacOS. Тому це загальна мова для створення програм, які можуть працювати на різних платформах.

Гнучкість, простота використання та велика спільнота розробників роблять Python ідеальною мовою для машинного навчання (ML).

Однією з переваг Python для ML є колекція бібліотек, розроблених для ML. TensorFlow, PyTorch, Keras і Scikit-Learn — бібліотеки, які надають різноманітні інструменти для створення, проектування та розгортання моделей машинного навчання. Чиста та зрозуміла мова допомагає розробникам легко розвивати та вивчати машинне навчання. Так як це універсальна мова, розробники можуть

використовувати її на різних етапах процесу машинного навчання, від проектування та технічного процесу до навчання та розробки.

Середовище легко сумісне з іншими мовами програмування, такими як C++ і Java, що дозволяє розробникам використовувати Python для створення великих концепцій систем машинного навчання та вирішення невеликих проблем.

Завдяки багатій бібліотеці Python та інтуїтивно зрозумілій інтеграції можливо бути впевненим, що складні форми можна створювати та впроваджувати з легкістю. Головною перевагою цієї мови є кількість бібліотек, особливо для програм машинного навчання та науки про дані. Бібліотека надає стандартні функції та алгоритми, які розробники можуть використовувати для виконання своїх завдань, заощаджуючи час і зусилля на написання цих алгоритмів та функцій з початку. Бібліотеки призначені для вирішення конкретних завдань, розпізнавання зображень або обробки природної мови та надання спеціальних інструментів для полегшення цих завдань.

Python зазвичай вважається мовою, яку легко тестувати, завдяки численним функціям і бібліотекам, які підтримують розробку за принципом тестування (TDD).

Одна з переваг Python полягає в тому, що код Python може працювати послідовно на різних платформах. Інтерпретатор Python доступний для багатьох платформ, включаючи Windows, MacOS і Linux. Тому розробники можуть писати код Python на одній платформі та використовувати його на іншій платформі, не змінюючи код. Можливості Python особливо корисні для великих розподілених проектів, що працюють на кількох платформах. Python дозволяє розробникам писати код, який бездоганно працює на всіх платформах і пристроях. Це економить час і зусилля порівняно зі створенням окремої бази коду для кожної платформи. Це особливо важливо для компаній, які хочуть перенести свої програми на нову хмару або платформу.

3.3 Бібліотеки Keras та TensorFlow

Keras — це передова бібліотека нейронної мережі з відкритим кодом, створена на Python, яка може працювати з Theano, TensorFlow або CNTK. Програма проста у використанні, масштабована та модульна, тому ви можете швидко тестувати глибокі нейронні мережі.

Він не може обробляти обчислення низького рівня, тому для цього використовує інженерну бібліотеку. Бібліотека Backend діє як високорівнева оболонка API навколо низькорівневого API, що дозволяє їй працювати з TensorFlow, Theano або CNTK.

Для його належної роботи потрібне додаткове програмне забезпечення. TensorFlow — це широко використовувана бібліотека Python для машинного навчання. Бібліотека — це набір рекомендованого коду, який використовують розробники, щоб уникнути написання коду з нуля. У зв'язку з цим Keras також додав бібліотеку поверх бібліотеки для підтримки завдань ШІ.

Keras забезпечує простий інтерфейс. TensorFlow забезпечує низьку продуктивність, тоді як Keras забезпечує високу продуктивність. Архітектура на основі Keras вимагає менше коду. Однак, якщо потрібно, можливо отримати доступ до додаткових функцій TensorFlow.

Keras має загальні алгоритми та функції машинного навчання. Це економить час і кодування, що дуже корисно для розробників. Keras дозволяє легко налаштувати параметри моделі. Це дозволяє розробникам проектувати нейронні мережі структурованим і неструктурованим способом, дозволяючи їм проектувати архітектуру для конкретного завдання.

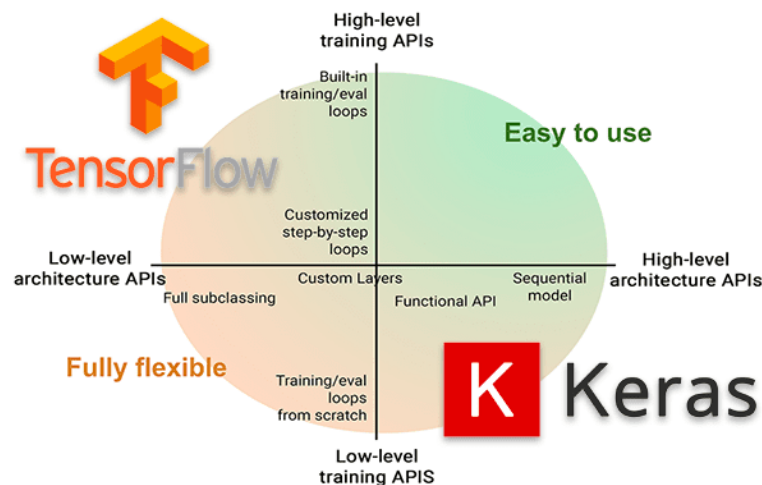


Рисунок 3.1 — Взаємозв'язок між Keras та TensorFlow

Як довідкова бібліотека Keras допомагає розробляти моделі глибокого навчання, надаючи розширену архітектуру. Усі обчислення низького рівня, такі як тензори, згортки тощо. Він не керується самим Keras, але базується на бібліотеці керування Tensor, яка використовується у внутрішньому двигуні.

Keras скористався цією можливістю і замість додавання єдиної бібліотеки тензорів і функцій, пов'язаних із цією бібліотекою, пропонує інтеграцію з декількома серверними механізмами Keras.

TensorFlow — це інструмент глибокого навчання, який використовується для машинного навчання та глибокого аналізу нейронних мереж на різних мовах, таких як Java, C++ або Python.

Theano — це широко використовувана бібліотека Python з відкритим кодом для багатовимірних обчислень, включаючи Skippy і NumPy. Він використовує графічний процесор для швидкого й ефективного виконання обчислень шляхом автоматичного відтворення графічної моделі. Він підходить для мобільних додатків, оскільки вони спочатку чисельно спостерігаються, а потім вирішуються за допомогою алгоритмів стабільності.

CNTK (Microsoft Cognitive Toolkit) — це версія з відкритим вихідним кодом Cognitive Tools від Microsoft. Він містить усі основні будівельні блоки для побудови нейронної мережі, моделі реалізовані на C++ або Python, а моделі для прогнозування створені на C# або Java.

Keras має такі переваги:

- Він простий для розуміння та дозволяє налаштовувати живі веб-інструменти;
- Він має величезну підтримку спільноти, оскільки багато компаній ШІ хочуть його використовувати;
- Він підтримує багато Backend, усі з яких можна використовувати TensorFlow, CNTK і Keras;
- Він підтримує розпаралелювання даних, що означає, що Keras можна інсталиювати на кількох графічних процесорах одночасно, що прискорює виконання великих наборів даних.

Єдина проблема полягає в тому, що Keras має рівень обробки, і якщо ви хочете створити цілий масив, він не дозволить вам, оскільки він не може працювати з API нижчого рівня. Він підтримує API високого рівня для механізмів виконання (TensorFlow, Theano та CNTK).

3.4 Бібліотека комп'ютерного зору OpenCV

OpenCV (Open Source Computer Vision), розроблений компанією Intel, є міжплатформною бібліотекою комп'ютерного бачення для обробки зображень у

реальному часі. OpenCV є корисним інструментом для всіх завдань, пов'язаних із комп'ютерним зором. OpenCV написаний на C і C++.

Бібліотека OpenCV містить понад 2500 алгоритмів, та велику документацію. Містить вихідний код і приклади коду комп'ютерного бачення в реальному часі. Розробники, які використовують пакети Python, можуть інтегрувати OpenCV у свої проекти.

OpenCV призначений для підвищення продуктивності та ефективності ресурсомістких завдань комп'ютерного зору. Тому компанія зосереджується на додатках бачення ШІ в реальному часі. Це програмне забезпечення з відкритим вихідним кодом написано оптимізованою мовою C і працює на багатоядерних (багатопоточних) процесорах. OpenCV створює просте у використанні середовище комп'ютерної візуалізації, яке дозволяє швидко створювати складні програми комп'ютерного зору, що охоплюють понад 500 різних функцій і полів зору. Використовується для тестування продукту, обробки медичних зображень, аналізу безпеки, інтерфейсі і машинному баченні. Розширені функції обробки зображень включають обробку відеопотоку, з'єднання кількох камер. Він підтримує масштабування та різні функції обробки зображень. Машинне навчання відіграє важливу роль у комп'ютерному зорі, тому OpenCV включає публічні бібліотеки машинного навчання, зосереджені на кластеризації та статистичному розпізнаванні образів.

Комп'ютерне бачення – це непросте завдання, але досягнення кращої продуктивності за менших витрат є ціллю комп'ютерного зору. Під час роботи камери, дані зображення часто спотворюються шумом і фізичними змінами, такими як освітлення, відбиття, рух і недоліки лінз, датчики або механічне регулювання. Щоб вирішити ці складні проблеми, розробники повинні створити складні конвеєри комп'ютерного зору, які імітують потоки даних. Ця логіка програми об'єднує все, від отримання зображення до обробки (зменшення шуму, фільтрування, усунення шумів), і подається в один або кілька алгоритмів комп'ютерного зору. OpenCV надає розробникам інструменти для вирішення проблем комп'ютерного зору. Розширеного функціоналу бібліотеки може бути достатньо для вирішення більш складних завдань з точки зору штучного інтелекту.

Платформа Viso Suite використовує потужність OpenCV за допомогою технології нульового коду для прискорення процесу розробки. Це дозволяє розробникам створювати конвеєри комп'ютерного зору за допомогою

модульних компонентів. Автоматизоване керування моделлю штучного інтелекту без використання коду спрощує співпрацю та прискорює створення та підтримку конвеєрів комп'ютерного зору. Використання OpenCV принесе користь як експертам, так і розробникам-початківцям.

OpenCV — це найпотужніший обчислювальний інструмент, який використовується для різних завдань комп'ютерного зору, починаючи від розпізнавання зображень 2D або 3D аналізу до відстеження руху, розпізнавання обличчя тощо. Технологія розпізнавання зображень використовується для ідентифікації реальних об'єктів на відео або зображеннях, таких як автомобілі, люди, тварини та інші частини будівель або машин. Сегментація зображення використовує алгоритм зображення для поділу зображення на сегменти. Сегментація часто використовується для згладжування, трансформації або покращення зображень, часто з використанням функцій комп'ютерного зору. Жести, руки чи обличчя можна розпізнавати та сегментувати для створення певних сегментів. Аналіз рухів часто є частиною аналізу патернів рухів за допомогою елементів. Обчислення положення об'єкта дозволяє зрозуміти положення об'єкта в тривимірному просторі, наприклад його обертання.

Автоматичне розпізнавання обличчя використовується для розпізнавання обличчя людей і пов'язування їх із базами даних на основі розпізнавання обличчя. OpenCV FaceRecoincer надає популярний алгоритм розпізнавання обличчя для реальних програм.

3.5 Бібліотека комп'ютерного зору OpenVino

OpenVino — це інструмент глибокого навчання, розроблений Intel під назвою «Відкрита візуалізація та оптимізація нейронних мереж». OpenVINO зосереджується на покращенні продуктивності нейронних мереж.

Використовуючи OpenVINO, розробники можуть налаштувати модель, включаючи бажану структуру моделі та глибоку модель із найпопулярнішим механізмом C++ API. OpenVINO надає вбудовані можливості для прискорення розробки програм і рішень, які виконують різноманітні завдання за допомогою комп'ютерного зору, автоматичного розпізнавання мовлення, обробки природної мови (NLP), машинного навчання.

Глибокі нейронні мережі (DNN) досягли успіху в багатьох галузях промисловості та підвищили точність комп'ютерних алгоритмів. Однак

впровадження та побудова цих точних і корисних моделей вимагає розробки апаратного забезпечення та обчислювальних систем. OpenVINO допомагає розробляти моделі DNN за допомогою простого та ефективного способу прийняття кращих рішень.

OpenVino базується на новому поколінні штучних мереж (ANN), як згорточні нейронні мережі (CNN), рекурентні мережі. Апаратне забезпечення OpenVINO включає апаратне забезпечення Intel і комп'ютерну графіку, що підвищує продуктивність і прискорює розробку програмного забезпечення. OpenVINO розроблено для завдань і продуктів штучного інтелекту з бібліотекою попередньо визначених і добре розроблених функцій. Набір інструментів OpenVINO також включає інші інструменти комп'ютерного зору, такі як OpenCV, OpenCL.

OpenVINO Accelerator забезпечує простий механізм прискорення для багатьох процесорів і прискорювачів Intel. Глибинне навчання на основі згорткової нейронної мережі (CNN) містить понад 30 попередньо підготовлених і зразкових кодів, написаних за допомогою публічного API. З більш ніж 100 функціями та загальними функціями. Метод з відкритим кодом який інтегрує частини робочого процесу з кодом OpenCL (Open Computing Language) та іншими інструментами, перетворює частини стандартної моделі глибокого навчання та реалізує паралельне програмування. Широкий набір інструментів дозволяє користувачам розширювати штучний інтелект на локальні та хмарні програми за допомогою таких функцій, як оптимізація моделі, віддалене моделювання та інтегрована адаптація nGraph.

OpenVINO дозволяє встановлювати засоби комп'ютерного зору на різні пристрої. Завантажувати та створювати моделі за допомогою різних інструментів, таких як PyTorch, TensorFlow.

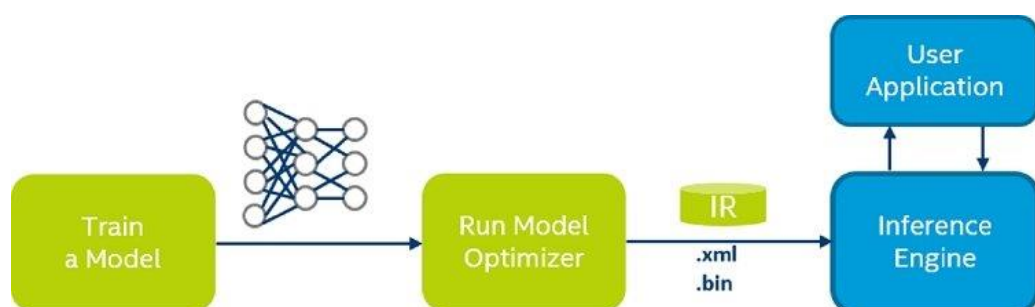


Рисунок 3.2 – Робочий процес OpenVINO

Цей інструмент має кілька унікальних функцій, деякі з яких дуже корисні. Але ці функції є лише частиною функцій, які пропонує програмне забезпечення. Унікальною особливістю OpenVINO є його здатність працювати з кількома пристроями. Ця функція дозволяє авторам комп'ютерного бачення підвищити свою продуктивність. Також це дозволяє користувачам повною мірою скористатися перевагами поєднання апаратного та програмного забезпечення. Більшість програм використовують ЦП (центральний процесор) і графічний процесор (GPU), щоб отримати повну продуктивність.

Інструменти OpenVINO для зменшення програмного забезпечення, такі як Software Manager, дозволяють користувачам швидко та легко зменшувати розмір програм. Розмір програми означає обсяг простору та ресурсів, які програма використовує на комп'ютері користувача. Тому зменшення розміру програмного забезпечення є важливою метою розробки.

Програми OpenVINO доступні для різноманітних функцій і цілей. Наприклад, нові моделі глибокого навчання мають окремий компонент точності, коли інші попередні завантаження не забезпечують необхідної точності. Цей компонент називається Neural Network Compression Framework (NNCF). NNCF охоплює функції OpenVINO, які не включені в продукт для завантаження. Методи оптимізації та моделі, які підтримує ця програма, взяті безпосередньо з інструментів OpenVINO.

Функції NNCF:

- Автоматична конвертація моделі: використання підходу до оптимізації для покращення існуючої моделі глибокого навчання не потребує змін у вмісті користувача.
- Уніфікований API (інтерфейс прикладного програмування). Усі методи стиснення спираються на певну абстракцію, надану фреймворком, комбінація алгоритмів.
- Комбінація алгоритмічних конвеєрів дозволяє виконувати кілька алгоритмів одночасно, а оптимальна модель будується з одним рівнем уточнення
- Підтримка розподіленого навчання: оптимізацію моделі глибокого навчання можна організувати в багатовузловому розподіленому кластері.
- Процес оптимізації можна налаштувати стандартним способом за допомогою файлу конфігурації JSON. Файл конфігурації JSON — це найпростіший спосіб налаштувати параметри, які використовуються моделлю глибокого навчання.

- ONNX цей плагін має можливість експорту у формат ONNX, відкритий стандарт для взаємодії машинного навчання.

3.6 Бібліотека з відкритим кодом Dlib

Набір інструментів і бібліотек з відкритим кодом, написаних за ліцензією Dlib Boost мовою C++. Dlib надає широку функціональність у різних сферах машинного навчання у тому числі математичні методи, такі як квадратичні розв’язувачі, інструменти обробки зображень і різні мережеві функції.

Dlib містить розширені інструменти для розпізнавання об’єктів у просторі, відстеження об’єктів, розпізнавання обличчя (частина об’єкта, яка виглядає як обличчя) і розпізнавання обличчя.

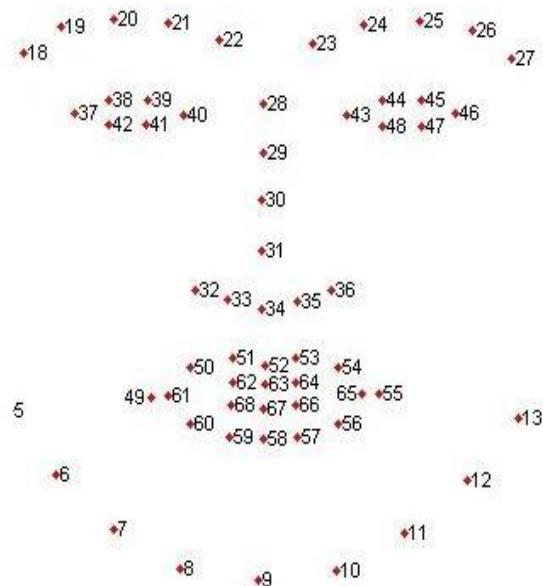


Рисунок 3.3 – Орієнтири у Dlib

Незважаючи на те, що Dlib є кросплатформним інструментом, багато існуючих функцій, пов’язаних із малюванням обличчя та аналізом (виявленням), реалізовано в Python за допомогою бібліотеки функцій OpenCV.

У Dlib є багато інтерфейсів та API, деякі з них дають додаткові функції, які не надавалися розробниками, наприклад перегляд різних об’єктів. Цей метод кодує зображення обличчя Dlib із 128 точками даних на обличчя та витягує унікальні хеш-параметри для кожного обличчя з інших зображень. Машина опорного вектора (SVM) навчається на даних, зібраних за допомогою scit-learn,

для створення ефективної FR-моделі з мінімальною затримкою та чудовою продуктивністю.

На практиці розпізнавання обличчя працює так само, як розширене розпізнавання обличчя, або перший крок, коли система фокусується на відомих обличчях і використовує доступні ідентифікаційні дані для активації функції. Конкретне використання цих інструментів викликає суперечки і обмежується повсякденною діяльністю поліції та інших державних органів.

Від отримання прибутку до створення сезонних робочих місць і ряду послуг, приватні компанії можуть використовувати розпізнавання обличчя для спрощення організації зустрічей і безпечного доступу.

Dlib дуже швидкий і простий. Він працює зі швидкістю 30 кадрів/с і розпізнає вираз обличчя за мілісекунди, але лише за правильних умов. Dlib також можна навчити розпізнавати особливі риси обличчя для досліджень і медичних застосувань.

Dlib має дві функції для розпізнавання обличчя.

Алгоритм орієнтованого градієнта (HoG) + гистограма Dlib лінійної опорної векторної машини (SVM) забезпечує швидке розпізнавання обличчя, але має обмежені можливості розпізнавання обличчя в різних середовищах.

Датчики HoG+SVM підходять для нестандартних додатків, де обличчя учасників є прямими та непомітними, наприклад, банкомати, центри розпізнавання автомобілів і програми виявлення руху транспортних засобів, де камери записують дані в реальному часі.

MMOD — це потужний і надійний детектор обличчя на основі графічного процесора, що застосовує згорткові нейронні мережі (CNN) для захоплення обличчя навіть у темних кутах і важкодоступних місцях, що робить його ідеальним для трекерів і тих, хто часто подорожує.

Незважаючи на те, що MMOD не є повною модифікацією послідовності HoG + SVM, його можливо використовувати до окремої HoG, що розглядають розріджені пікселі як особливості, що представляють інтерес для прихованих сигналів, включаючи виявлення обличчя.

Перевагою HoG + Linear SVM є менші ресурси порівняно з DLIB.

Недоліком є те, що стандартні налаштування вимагають мінімального розміру 80 x 80 пікселів. Більш того, цей метод дає негативні результати і з боку обличчя.

Перевагою ММОД є базова здатність визначати складні методи які можуть відрізнятися згідно цільового домену.

Недоліком є те, що процесор повільніший, ніж HoG/LSVM. І це не включає в себе здатність HoG/LSVM розпізнавати обличчя розміром менше 80 x 80 пікселів.

3.7 Бібліотека PyTorch

PyTorch — це бібліотека тензорної оптимізації глибокого навчання на основі Python і Torch, яка в основному використовується для додатків GPU і CPU. PyTorch є кращим перед іншими фреймворками глибокого навчання, такими як TensorFlow і Keras, тому що він використовує динамічну комп'ютерну графіку та є повністю написаний на Python. Це дозволяє дослідникам хмарних мереж, програмістам і розробникам запускати та тестувати фрагменти коду в реальному часі. Користувачам не потрібно чекати, поки весь код буде встановлено, щоб побачити, чи працює частина коду.

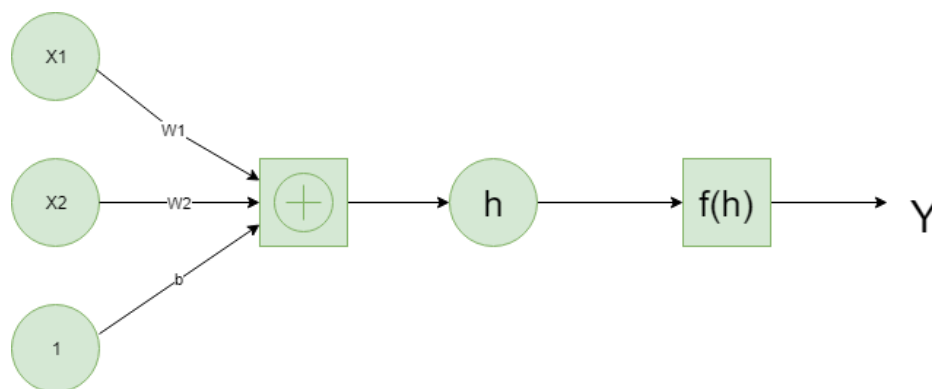


Рисунок 3.4 – Глибоке навчання з PyTorch

PyTorch має основні функції:

- Тензорне обчислення (схоже на NumPy) із сильною підтримкою прискорення GPU.
- Автоматичний контраст для створення та навчання глибоких нейронних мереж.

Основні функції PyTorch подібні до NumPy. Коли ми представляємо дані в машинному навчанні, нам потрібно робити це в цифровому вигляді. Тензор — це простий пристрій, який може зберігати дані в кількох вимірах. Але для математичної точки зору тензор — це основна одиниця даних, яка може стати основою складних математичних дій. Такі як числа, вектори, матриці або змішані

матриці, такі як числові матриці. Процесор або процесори можуть обробляти тензори для підвищення продуктивності. Існують різні типи тензорів, наприклад тензор з плаваючою речовиною, подвійний тензор, напівтензор, тензор `int` і довгий тензор, але PyTorch використовує 32-бітний тензор з плаваючою точкою.

Код для виконання математичних операцій у PyTorch схожий на код у NumPy. Користувачі повинні створити два тензори, а потім виконувати такі операції, як додавання, віднімання, множення та ділення.

Щоб створити масив випадкових чисел у PyTorch, скористайтеся функцією `randn()`, яка повертає тензор, повний випадкових чисел, отриманих із нормального розподілу. Якщо розмістити випадкове початкове число на початку, ті самі числа генеруватимуться під час кожного запуску коду.

PyTorch використовує моделі для представлення нейронних мереж.

Модуль `AutoGrade` — це автоматизований інструмент PyTorch для швидкого обчислення оцінки. `AutoGrade` створює орієнтований ациклічний графік із листами як вхідним тензором і коренями як вихідним тензором.

Модуль `Optim` — це готовий набір алгоритмів оптимізації, які можна використовувати для побудови нейронних мереж.

Модуль `nn` містить кілька класів, які допомагають створювати моделі нейронних мереж. Усі модулі PyTorch є підкласами модуля.

Можливо використовувати діаграму розрахунку PyTorch для розрахунку значень градієнта згенерованих мереж. PyTorch використовує динамічні статистичні таблиці. У прямому обчисленні графік визначається неявно за допомогою оператора додавання. Динамічні діаграми більш гнучкі, ніж статичні, і користувачі можуть перемикатися між діаграмами та обчисленнями. Їх легко налагодити, виконуючи код рядок за рядком. Потужні графіки PyTorch дозволяють легко візуалізувати проблеми у вашому коді.

У PyTorch комп'ютерна графіка відображається на кожній ітерації, що дозволяє командам керування потоком Python впливати на загальну форму та розмір графіка на кожній ітерації. Перевагою є те, що не потрібно кодувати всі методи перед початком навчання.

Робота з великим обсягом даних вимагає завантаження всіх даних разом. В результаті обробка пам'яті припиняється, і програма починає працювати повільніше. Код, який контролює структуру даних, важко підтримувати. PyTorch має два інтерфейси `DataLoader` і `DataSet`, які автоматично обробляють завантаження даних, читання коду та форматування. `DataSet` та `DataLoader`

дозволяють користувачам використовувати попередньо встановлені набори даних і власні дані. Коли набори даних містять зображення та пов'язані символи, завантажувачі даних дозволяють користувачам легко отримувати доступ до зображень, пов'язуючи набори даних із зображеннями та надсилаючи повторювані дані.

3.8 Бібліотека Pillow

Бібліотека зображень Python (PIL) — це пакет віртуальних зображень для мови Python. Він містить прості інструменти для редагування, створення та збереження зображень. Бібліотека Visual Python застаріла, але проект під назвою Pillow виділив оригінальний проект PIL і додав підтримку для Python3.x. Цей файл підтримує декілька форматів файлів зображень, зокрема BMP, PNG, JPEG і TIFF. Бібліотека дозволяє легко створювати нові декодери файлів, які підтримують нові формати.

Редагування цифрових зображень означає обробку цифрових зображень на комп'ютері. Зображення можна збільшити, розмити, витягти текст тощо за допомогою обробки зображень. Є багато способів створення цифрового зображення.

Бібліотека зображень Python підходить для програм зберігання зображень і пакетної обробки. Пакет Python Pad дозволяє створювати малюнки, конвертувати з одного формату в інший, друкувати зображення.

Можливо відображати зображення через інтерфейси PhotoImage, BitmapImage та Windows DIB, а також може використовуватися з PythonWin та іншими інструментами на базі Windows, а також багатьма іншими графічними інтерфейсами користувача (GUI). Зовнішній дисплей має метод `show()` для редагування зображення для збереження його на диску.

Бібліотеки Pillow містять основні функції обробки зображень. Ви можете змінювати розмір, змінювати розмір і повертати зображення. Модуль Pillow дозволяє отримувати детальні статистичні дані із зображень за допомогою методу гістограми, які потім можна використовувати для статистичного аналізу та автоматичного підвищення контрастності.

4 ДОСЛІДЖЕННЯ АЛГОРИТМУ ДЛЯ ВИЗНАЧЕННЯ СТАТІ ТА ВІКУ ЛЮДИНИ

Основна мета цього розділу - застосування бази даних для ідентифікації віку та статі. Визначення віку та статі за допомогою Python та бібліотек OpenCV, argparse, Keras, TensorFlow.

4.1 Реалізація проекту для розпізнавання статі та віку людини за її зображенням

Цей проект Python використовує глибоке навчання для точного визначення статі та віку людини за одним зображенням обличчя. Для навчання моделі використана база даних Adience Benchmark Gender And Age Classification. Припускаючи, що стать чоловіча чи жіноча, передбачуваний вік знаходиться в одному з таких діапазонів: (0-2), (4-6), (8-12), (15-20), (25 – 32), (38 – 43), (48 – 53), (60 – 100). Такі фактори, як макіяж, освітлення, перешкоди, риси обличчя, ускладнюють визначення точного віку людини за фотографією.

У цьому проекті Python використовує інструмент бази даних Adience. Цей документ працює як зразок для системи розпізнавання зображень обличчя. Він також містить багато візуальних ефектів, таких як перешкоди, яскравість і рух. Також вона містить 8 вікових груп, 2284 елементи, 26 580 зображень, зібраних з альбому Vision Flickr і опублікованих за ліцензією Creative Commons (CC). Я використав цю інформацію як довідкову.

У проекті є файл .pb(protobuf) для розпізнавання обличчя. Він містить опис модельної програми і тренувань моделі. Ми можемо використати це для використання навченої моделі. Крім того, файл .pbtxt має текстовий формат, а файл .pb protobuf — у двійковому форматі. Це файли TensorFlow. Файл .prototxt описує макет мережі для визначення віку та статі людини, а файл .caffemodel описує внутрішню структуру шарів.

Для визначення статі та віку особи за зображенням використовується команда:

```
python detect.py --image <image_name>
```

Для визначення статі та віку особи за допомогою веб-камери використовується команда:

```
python detect.py
```

```

1 Завантаження зображення, які знаходяться в папці database_images
2
3 import config as cfg
4 import os
5 from my_face_recognition import f_main
6 import cv2
7 import numpy as np
8 import traceback
9
10 def load_images_to_database():
11     list_images = os.listdir(cfg.path_images)
12     # файли фільтра, які не є зображеннями
13     list_images = [File for File in list_images if File.endswith(('.jpg', '.jpeg', 'JPEG'))]
14
15     # ініціалізація змінних
16     name = []
17     Feats = []
18
19     # перевантажити зображення
20     for file_name in list_images:
21         im = cv2.imread(cfg.path_images+os.sep+file_name)
22
23         # отримаємо риси обличчя
24         box_face = f_main.rec_face.detect_face(im)
25         feat = f_main.rec_face.get_features(im, box_face)
26         if len(feat)!=1:
27
28             це означає, що обличчя немає або є більше одного обличчя
29
30             continue
31         else:
32             # вставити нові функції в базу даних
33             new_name = file_name.split(".")[0]
34             if new_name == "":
35                 continue
36             name.append(new_name)
37             if len(Feats)==0:
38                 Feats = np.frombuffer(feat[0], dtype=np.float64)
39             else:
40                 Feats = np.vstack((Feats,np.frombuffer(feat[0], dtype=np.float64)))
41             return name, Feats
42
43 def insert_new_user(rec_face,name,feat,im):
44     try:
45         rec_face.db_names.append(name)
46         if len(rec_face.db_features)==0:
47             rec_face.db_features = np.frombuffer(feat[0], dtype=np.float64)
48         else:
49             rec_face.db_features = np.vstack((rec_face.db_features,np.frombuffer(feat[0], dtype=np.float64)))
50         # збережимо зображення
51         cv2.imwrite(cfg.path_images+os.sep+name+".jpg", im)
52         return 'ok'
53     except Exception as ex:
54         error = ''.join(traceback.format_exception(etype=type(ex), value=ex, tb=ex.__traceback__))
55         return error

```

Рисунок 4.1 –Реалізація проекту для розпізнавання

Важливу роль відіграє функція по розпізнаванню обличчя на зображенні. Ця функція спочатку створює неглибоку копію кадру зображення, визначає її висоту та ширину. Використовуючи ці розміри, виділяється крапка зображення, яка спрощує обробку зображення. Об'єкт передається через нейронну мережу, призначену визначення осіб. Щойно особи ідентифіковані, функція повторює ці виявлення, визначаючи точні координати кожної особи. Використовуючи ці координати, навколо кожної виявленої особи малюється прямокутник, виділяючи його на зображенні.

```

1  Встановлюється бібліотека deepface.
2  pip install deepface
3  Потім створюється екземпляр моделі
4  emo = f_my_Age_Model()
5  Вводиться зображення, де видно лише обличчя
6  emo.predict_age(face_image)
7
8  # импорт базових моделей VGG Face
9  from deepface.basemodels import VGGFace
10 import os
11 from pathlib import Path
12 import gd
13 import numpy as np
14 from keras.models import Model, Sequential
15 from keras.layers import Convolution2D, Flatten, Activation
16 from keras.preprocessing import image
17 import cv2
18
19 class Age_Model():
20 def __init__(self):
21 self.model = self.loadModel()
22 self.output_indexes = np.array([i for i in range(0, 101)])
23
24 def predict_age(self, face_image):
25 image_preprocessing = self.transform_face_array2age_face(face_image)
26 age_predictions = self.model.predict(image_preprocessing)[0,:]
27 result_age = self.findApparentAge(age_predictions)
28 return result_age
29
30 def loadModel(self):
31 model = VGGFace.baseModel()
32

```

Рисунок 4.2 –Реалізація прекуту для визначення віку (частина 1)

Оцінка віку - це метод ідентифікації людей вікової групи. Цей вік може бути фактичним віком, указаним віком або орієнтовним віком. Фактичний вік — це кількість років, які людина прожила з моменту народження. Фактичний вік обчислюється на основі даних про зовнішній вигляд людини. Хоча пристрій обчислює вік відповідно до фізичного стану людини, але існують відмінності в тому, як люди старіють. Оцінка віку може бути використана як проблема класифікації, регресії або комбінація класифікації та регресії. Іншим фактором, який може вплинути на оцінку віку, є складність поєднання великих хронологічних баз зображень.

```

33 classes = 101
34 base_model_output = Sequential()
35 base_model_output = Convolution2D(classes, (1, 1), name='predictions')(model.layers[-4].output)
36 base_model_output = Flatten()(base_model_output)
37 base_model_output = Activation('softmax')(base_model_output)
38
39 age_model = Model(inputs=model.input, outputs=base_model_output)
40
41 home = str(Path.home())
42 if os.path.isfile(home+'/.deepface/weights/age_model_weights.h5') != True:
43 print("age_model_weights.h5 will be downloaded...")
44 url = 'https://drive.google.com/uc?id
45 output = home+'/.deepface/weights/age_model_weights.h5'
46 gd.download(url, output, quiet=False)
47 age_model.load_weights(home+'/.deepface/weights/age_model_weights.h5')
48 return age_model
49
50 def findApparentAge(self, age_predictions):
51 apparent_age = np.sum(age_predictions * self.output_indexes)
52 return apparent_age
53
54 def transform_face_array2age_face(self, face_array, grayscale=False, target_size = (224, 224)):
55 detected_face = face_array
56 if grayscale == True:
57 detected_face = cv2.cvtColor(detected_face, cv2.COLOR_BGR2GRAY)
58 detected_face = cv2.resize(detected_face, target_size)
59 img_pixels = image.img_to_array(detected_face)
60 img_pixels = np.expand_dims(img_pixels, axis = 0)
61 #normalize input in [0, 1]
62 img_pixels /= 255
63 return img_pixels

```

Рисунок 4.3 – Реалізація прекуту для визначення віку (частина 2)

```

1 # импорт базових моделей VGG Face
2 from deepface.basemodels import VGGFace
3 import os
4 from pathlib import Path
5 import gd
6 import numpy as np
7 from keras.models import Model, Sequential
8 from keras.layers import Convolution2D, Flatten, Activation
9 from keras.preprocessing import image
10 import cv2
11
12 class Gender_Model():
13     def __init__(self):
14         self.model = self.loadModel()
15
16     def predict_gender(self, face_image):
17         image_preprocessing = self.transform_face_array2gender_face(face_image)
18         gender_predictions = self.model.predict(image_preprocessing)[0,1]
19         if np.argmax(gender_predictions) == 0:
20             result_gender = "Woman"
21         elif np.argmax(gender_predictions) == 1:
22             result_gender = "Man"
23         return result_gender
24
25     def loadModel(self):
26         model = VGGFace.baseModel()
27
28         classes = 2
29         base_model_output = Sequential()
30         base_model_output = Convolution2D(classes, (1, 1), name='predictions')(model.layers[-4].output)
31         base_model_output = Flatten()(base_model_output)
32         base_model_output = Activation('softmax')(base_model_output)
33
34         gender_model = Model(inputs=model.input, outputs=base_model_output)
35         home = str(Path.home())
36         if os.path.isfile(home+'/.deepface/weights/gender_model_weights.h5') != True:
37             print("gender_model_weights.h5 will be downloaded...")
38             url = 'https://drive.google.com/uc?id
39             output = home+'/.deepface/weights/gender_model_weights.h5'
40             gd.download(url, output, quiet=False)
41         gender_model.load_weights(home+'/.deepface/weights/gender_model_weights.h5')
42         return gender_model
43
44
45     def transform_face_array2gender_face(self, face_array, grayscale=False, target_size = (224, 224)):
46         detected_face = face_array
47         if grayscale == True:
48             detected_face = cv2.cvtColor(detected_face, cv2.COLOR_BGR2GRAY)
49             detected_face = cv2.resize(detected_face, target_size)
50         img_pixels = image.img_to_array(detected_face)
51         img_pixels = np.expand_dims(img_pixels, axis = 0)
52         #normalize input in [0, 1]
53         img_pixels /= 255
54         return img_pixels

```

Рисунок 4.4 – Реалізація проекту для визначення статі

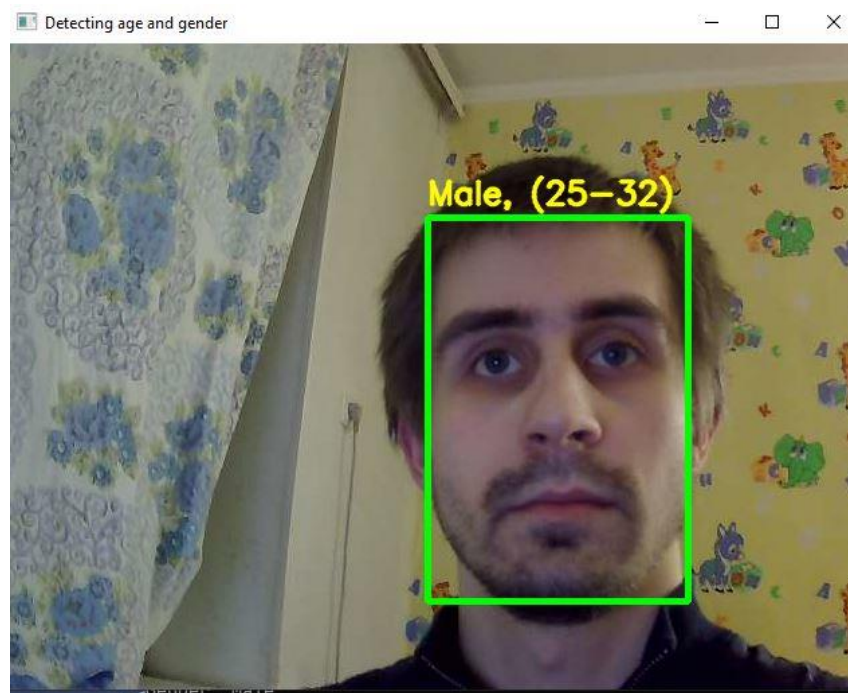


Рисунок 4.5 –Визначення статі та віку за допомогою Web-камери під час виконання проекту

4.2 Аналіз проекту для розпізнавання статі та віку людини за її зображенням

Визначити стать та вік можливо за зображеннями осіб, для цього був розроблен проект, що дозволяє це зробити. Щоб виконати оцінку, пропонується ланцюжок, який складається з наступних кроків: попередня обробка, отримання ознак і оцінка (регресія віку та класифікація статі). Щоб дізнатися, наскільки достовірний результат визначення за такими характеристиками як оцінка статі, та оцінка віку.

Продуктивність визначення статі людини доволі висока, оскільки помилка близько 11%. Оцінка статі, зроблена людьми, помилка близько 1%. Таким чином, важко досягти кращої продуктивності, але отримані результати близькі до людських можливостей.

Результати проекту по визначенню віку, застосованої до набору даних Adience Benchmark Gender And Age Classification, показують, що 40% оцінених зразків мають відносну помилку менше 20%, а 60% менше 25%. Щоб дізнатися, наскільки точні результати дослідження, їх було порівнянно з тестом, де люди

оцінювали вік деяких об'єктів з набору даних Adience Benchmark Gender And Age Classification.

Потім порівнювалась продуктивність нашої системи з результатами людей і дійшли висновку, що результати доволі близькі. Системі важко оцінити краще, ніж людям, тому ми можемо зробити висновок, що наші результати задовільні. Реалізована система намагається передбачити реальні стать і вік. Однак ми працюємо із зображеннями обличчя, і вони просто дають нам інформацію про зовнішній вигляд людини. Тому складно оцінити реальний вік людини, а не те на скільки років вона виглядає.

Для того щоб адаптувати систему і зробити її корисною, є ключова вимога — навчити її за допомогою набору даних, який дійсно представляє реальні вхідні зображення, оскільки, як могли бачити при розробці, результати змінюються залежно від набору даних, що використовується. Також необхідно мати набір навчальних даних з ширшим віковим діапазоном, ніж необхідно, тому що оцінка межі вікових груп не така точна, як в інших вікових діапазонах.

Іншою реалізацією проекту є адаптація системи для відеовходів замість нерухомих зображень, що може бути ближчим до потреб реальних додатків. Оскільки не вік, ні стать істотно не змінюються в тривалості відео, прогноз не потрібно робити для кожного окремого кадру. Найпростіший спосіб зробити це взяти з відео лише одне зображення та діяти так само, але оскільки відео може містити багато різних зображень одного й того самого об'єкта, можливо цим скористатися і намагатися взяти якнайбільше зображень одного об'єкта, оцінити вік і стать для всіх з них і повернути значення залежно від результатів кожного зображення.

Цікавим додатком могла б стати реалізація системи для мобільних пристроїв з урахуванням якості камери, обчислювальної складності та обсягу пам'яті пристрою. При цьому продуктивність системи може погіршитися, але виконання має стати набагато простіше, що в деяких випадках може бути корисним.

ВИСНОВКИ

У кваліфікаційній роботі за темою “Дослідження методів розпізнавання статі та віку людини за її зображенням” було розглянуто технології для ідентифікації віку та статі людини. Були опрацьовані поняття, як штучні нейронні мережі та комп’ютерний зір.

У процесі дослідження було досліджено сферу застосування цих технологій, проблеми реалізації та способи їх подолання. Був проведений аналіз сфер де можуть використовуватися ці технології та як вони застосовуються. Розробки та створення таких технологій є корисними для розробників, тому доля їх використання збільшується з кожним роком.

Були розглянуті методи та алгоритми для ідентифікації віку та статі осіб за зображенням. Такі як Вейвлети Габора, дескриптор функції, тривимірне розпізнавання облич, згорткова нейронна мережа, рекурентна нейронна мережа та глибокі нейронні мережі.

Були розглянуті засоби розробки програмного забезпечення для ідентифікації віку та статі людини, огляд мови програмування Python. Був зроблен огляд бібліотеки комп’ютерного зору OpenVino, бібліотеки комп’ютерного зору OpenCV, бібліотеки Keras та TensorFlow, бібліотека з відкритим кодом Dlib, бібліотеки PyTorch, бібліотеки Pillow.

Таким чином, запропонована реалізація методу визначення віку та статі людини, розроблено програмне забезпечення для розпізнавання та виконано дослідження ефективності розпізнавання статі та віку особи за її зображенням, яке підтверджує можливість їх застосувань на практиці.

ПЕРЕЛІК ПОСИЛАНЬ

1. P. Viola and M.J. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001.
2. Md. Abdur Rahim, Md. Najmul Hossain, Tanzillah Wahid and Md. Shafiul Azam. "Face recognition using local binary patterns..
3. Robert Laganieri. OpenCV 3 Computer Vision Application Programming Cookbook / Robert Laganieri. – Birmingham: Packt Publishing, 2017. – 444 с.
4. Bhadeshia H. K. D. H. (1999). Neural Networks in Materials Science⁸.
5. Ibrahim Mousa Al-Zuabi. Predicting customer's gender and age depending on mobile phone data [Электронный ресурс] / Ibrahim Mousa AlZuabi, Assef Jafar, Kadan Aljoumaa. – 2019. – Режим доступа до ресурсу: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0180-9>.
6. Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation [Электронный ресурс] . – Режим доступа: [www / URL](http://www.robots.ox.ac.uk/~tvvg/publications/2017/CRFMeetCNN4SemanticSegmentation.pdf). – <https://www.robots.ox.ac.uk/~tvvg/publications/2017/CRFMeetCNN4SemanticSegmentation.pdf>
7. Thermal Infrared Face Recognition [Электронный ресурс] . – Режим доступа: www/URL. – <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8021210/>
8. Нейронные сети: распознавание образов и изображений с помощью ИИ [Электронный ресурс] . – <https://center2m.ru/ai-recognition>
9. Serrano, S., Diego, I., Conde, C., Cabello, E.: 'Recent advances in face biometrics with Gabor wavelets: a review', Pattern Recognit. Lett., 2010.
10. Chingovska, I., Anjos, A., Marcel, S.: 'On the effectiveness of local binary patterns in face anti-spoofing'. IEEE Int. Conf. Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 2012.
11. Best-Rowden, L., Han, H., Otto, C., Klare, B., Jain, A.: 'Unconstrained face recognition: identifying a person of interest from a media collection'. Technical Report, Technical Report MSU-CSE-14-1, Michigan State University, 2014.
12. Al-Osaimi, F., Bennamoun, M., Mian, A.: 'An expression deformation approach to non-rigid 3D face recognition', Int. J. Comput. Vis., 2009.

13. Lei, Y., Bennamoun, M., El-Sallam, A.: 'An efficient 3D face recognition approach based on the fusion of novel local low-level features', *Pattern Recognit.*, 2013.
14. Bereta, M., Karczmarek, P., Pedrycz, W., Reformat, M.: 'Local descriptors in application to the aging problem in face recognition', *Pattern Recognit.*, 2013.
15. Kanan, H., Faez, K.: 'Recognizing faces using adaptively weighted sub-Gabor array from a single sample image per enrolled subject', *Image Vis. Comput.*, 2010.
16. Park, C., Ki, M., Namkung, J., Paik, J.K.: 'Multimodal priority verification of face and speech using momentum back-propagation neural network', *Lect. Notes Comput. Sci.*, 2006.
17. Mark Lutz. *Learning Python* / Mark Lutz. – Sebastopol, United States: O'Reilly Media, 2013.
18. Naseem, I., Togneri, R., Bennamoun, M.: 'Linear regression for face recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.
19. Robert Laganiere. *OpenCV 3 Computer Vision Application Programming Cookbook* / Robert Laganiere. – Birmingham: Packt Publishing, 2017. – 444 c.