

МАРТИНИК Владислав Миколайович

**Модуль розпізнавання віку та статі людей на
основі нейронних мереж / Age and gender
recognition module based on neural networks**

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
В. М. Мартиник

Науковий керівник:
к.і.н., Г. В. Сапожник

Кваліфікаційну роботу
допущено до захисту:

" ____ " _____ 20__ р.

Завідувач кафедри
_____ **М. П. Комар**

ТЕРНОПІЛЬ - 2024

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
« ____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
МАРТИНИКУ Владиславу Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Модуль розпізнавання віку та статі людей на основі нейронних мереж / Age and gender recognition module based on neural networks

керівник роботи Сапожник Григорій Вікторович, к.іст.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 12 грудня 2023 р. № 753.

2. Строк подання студентом закінченої кваліфікаційної роботи 15 травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- Аналіз предметної області та постановка задачі дослідження.
- Огляд існуючих рішень у сфері розпізнавання віку та статі.
- Розробка алгоритмічного та інформаційного забезпечення системи.
- Розробка програмно-технологічного забезпечення.
- Аналіз результатів та валідація системи.

5. Перелік графічного матеріалу в роботі:

- Архітектура модуля розпізнавання віку та статі людей
- Схема алгоритму розпізнавання віку та статі людей на основі нейронних мереж
- графіки з результатами експериментальних досліджень.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2024 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2024 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2024 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2024 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2024 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 20.05.2024 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту у системі «Unichек».	до 10.06.2024 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 14.06.2024 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.06. 2024 р.	

Студент _____ В. М. Мартиник
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ Г.В. Сапожник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Модуль розпізнавання віку та статі людей на основі нейронних мереж» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 33 сторінки і містить 8 ілюстрацій, 1 таблицю та 11 використаних джерел. Метою роботи є розробка та оптимізація модуля розпізнавання віку та статі людей на основі нейронних мереж. Методами розроблення обрано метод аналізу (для дослідження існуючих підходів до розпізнавання), метод синтезу (для поєднання переваг існуючих методів), методи моделювання (для представлення та дослідження процесів розпізнавання), метод порівняльного аналізу (для оцінювання адекватності моделі розпізнавання). Внаслідок виконання роботи обґрунтовано раціональний підхід до розроблення моделей розпізнавання віку та статі людей та розроблено програмний засіб, який дозволяє створювати і досліджувати моделі розпізнавання віку та статі. Результати дослідження можуть бути використані в науково-дослідних установах і підрозділах підприємств, що займаються розробленням моделей розпізнавання.

Ключові слова: РОЗПІЗНАВАННЯ ВІКУ ТА СТАТІ, НЕЙРОННІ МЕРЕЖІ, ГЛИБИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, МОДЕЛЬ АДЕКВАТНОСТІ.

ANNOTATION

Qualification work on the topic «Age and gender recognition module based on neural networks» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 33 pages and it contains 8 figures, 1 table, and 11 sources.

The purpose of the work is to develop and optimize a module for age and gender recognition based on neural networks.

Research methods include analysis (to study existing approaches to recognition), synthesis (to combine the advantages of existing methods), modeling (to represent and study recognition processes), and comparative analysis (to evaluate the adequacy of the recognition model).

As a result of the work, a rational approach to the development of age and gender recognition models was substantiated, and a software tool was developed that allows creating and researching age and gender recognition models.

The research results can be used in research institutions and enterprise departments involved in the development of recognition models.

Keywords: AGE AND GENDER RECOGNITION, NEURAL NETWORKS, DEEP LEARNING, COMPUTER VISION, MODEL ADEQUACY.

ЗМІСТ

Вступ	7
1 Аналіз предметної області і постановка задачі дослідження	10
1.1 Огляд предметної області	10
1.2 Огляд існуючих рішень	13
1.3 Постановка задачі дослідження	16
2 Алгоритмічне та інформаційне забезпечення	19
2.1 Архітектура модуля розпізнавання віку та статі людей	19
2.2 Моделі для передбачення віку	21
2.2.1 VGG16	21
2.2.2 ResNet152V2	22
2.2.3 Xception	22
2.3 Моделі для класифікації статі	23
2.3.1 MobileNetV3Small та MobileNetV3Large	23
2.3.2 InceptionV3	23
2.3.3 ResNet50V2	24
2.4 Алгоритм розпізнавання віку та статі людей на основі нейронних мереж	24
3 Програмно-технологічне забезпечення	27
3.1 Опис набору даних	27
3.2 Аналіз архітектур базових моделей	29
3.3 Оптимізація моделі вікової класифікації	31
3.4 Оптимізація моделі гендерної класифікації	32
3.5 Емпіричний аналіз передбачень моделей	33
Висновки	35
Список використаних джерел	37
Додаток А Псевдокод алгоритму	39
Додаток Б Код для реалізації	40
Додаток В Апробація отриманих результатів	47

ВСТУП

Актуальність даної роботи обумовлена широким застосуванням систем розпізнавання облич у різних аспектах сучасного життя. Завдяки постійному зростанню обсягів цифрових даних та швидкому розвитку технологій штучного інтелекту, системи розпізнавання облич стають все більш важливими в різних сферах, таких як безпека, медицина, маркетинг, реклама, та багато інших. Розпізнавання віку та статі особи є ключовими завданнями у таких системах, оскільки ці дані можуть бути використані для персоналізованої аналітики, адаптації продуктів та послуг, таргетованої реклами та інших цілей.

Однією з актуальних проблем у сфері розпізнавання облич є забезпечення точності та ефективності алгоритмів, особливо в умовах різноманітності осіб, зміни освітлення, а також наявності артефактів або змін у зовнішньому вигляді. Розвиток нових методів та технологій для розпізнавання облич, зокрема використання глибинного навчання та нейронних мереж, є актуальною та перспективною галуззю досліджень.

Крім того, із зростанням обізнаності про етичні та приватні аспекти використання систем розпізнавання облич, увага дослідників та суспільства звертається до розробки методів та алгоритмів, які не тільки ефективно функціонують, але й забезпечують конфіденційність та захист персональних даних користувачів. Таким чином, актуальність робіт у сфері розпізнавання облич продовжує зростати, викликаючи неабиякий інтерес у наукових та прикладних колах.

Метою роботи є розробка та оптимізація модуля розпізнавання віку та статі людей на основі нейронних мереж. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Аналіз предметної області та постановка задачі дослідження.
2. Огляд існуючих рішень у сфері розпізнавання віку та статі.

3. Розробка алгоритмічного та інформаційного забезпечення системи.
4. Розробка програмно-технологічного забезпечення.
5. Аналіз результатів та валідація системи.

Об'єктом дослідження є процес розпізнавання віку та статі людей на основі нейронних мереж.

Предметом дослідження є алгоритми та технології, що використовуються для розпізнавання віку та статі на основі зображень облич.

Методи дослідження включають аналіз існуючих методів розпізнавання, експериментальні дослідження з використанням реальних даних, розробку та валідацію нейронних мереж для розпізнавання віку та статі, а також порівняння результатів з існуючими рішеннями у цій галузі.

Практичне значення даної роботи полягає в створенні і оптимізації модуля розпізнавання віку та статі людей на основі нейронних мереж. Одержані результати можуть знайти широке застосування у різних сферах, включаючи відеоспостереження, маркетингові дослідження, персоналізовану рекламу та безпеку. Наприклад, відомості про вік та стать особи можуть бути використані для аналізу поведінки покупців у магазинах, підбору цільової аудиторії для рекламних кампаній, а також для підвищення рівня безпеки в громадських місцях. Такий модуль може стати потужним інструментом для бізнесу та суспільства, сприяючи вдосконаленню процесів управління, аналізу та безпеки з використанням сучасних технологій штучного інтелекту.

Опис структури та обсягу роботи. Кваліфікаційна робота включає в себе вступ, три розділи, висновки та список використаних джерел. Загальний обсяг роботи складає 33 сторінки тексту, який містить 8 рисунків та 1 таблицю. У списку використаних джерел вказано 11 джерел, що займають 2 сторінки.

Апробація результатів дослідження. Основні теоретичні положення роботи й практичні результати дослідження доповідалися й обговорювалися V Всеукраїнської мультидисциплінарної студентської наукової конференції

«Науковий простір: аналіз, сучасний стан, тренди та перспективи», яка відбулася 17 травня 2024 року у місті Київ, Україна.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Огляд предметної області

Розпізнавання образів — це галузь комп'ютерних наук, яка включає методи для інтерпретації та аналізу зображень з метою ідентифікації об'єктів, людей, написів, сцен і багато чого іншого. Історія цієї дисципліни бере свій початок з 1960-х років, коли вчені почали розробляти перші системи, здатні обробляти цифрові зображення. Спочатку вони були досить примітивними і обмеженими до простих текстур і форм, але з часом, завдяки розвитку обчислювальної техніки та алгоритмів машинного навчання, можливості цих систем значно розширилися.

У сучасному світі розпізнавання образів стало невід'ємною частиною багатьох технологічних рішень, що змінюють наше повсякденне життя. Застосування цієї технології варіюється від безпеки та нагляду, медичної діагностики, до автономних транспортних систем і розумного дому. Таке широке впровадження є можливим завдяки прогресу в обчислювальній потужності, збільшенню обсягів даних і вдосконаленню алгоритмів глибокого навчання, які підвищують точність та швидкість обробки зображень.

Основними додатками розпізнавання образів є системи біометричної ідентифікації, які використовують відбитки пальців, сканування сітківки ока або розпізнавання облич для перевірки особистості. В автомобільній промисловості, алгоритми розпізнавання образів застосовуються в системах допомоги водію та розробці автономних транспортних засобів. У сфері здоров'я, ця технологія використовується для аналізу медичних знімків з метою виявлення захворювань. А в роздрібній торгівлі та рекламі

розпізнавання образів допомагає в аналізі поведінки споживачів та персоналізації рекламних матеріалів.

Аналіз віку та статі особи базується на комплексі алгоритмів машинного зору та навчання, які використовують анатомічні та фізіологічні особливості обличчя для визначення демографічних характеристик. Методи класифікації та регресії у поєднанні з технологіями комп'ютерного зору дозволяють ідентифікувати стать та оцінити вік, виходячи із структурних відмінностей обличчя, що змінюються з часом.

Аналіз віку та статі має велике соціальне та комерційне значення, оскільки використовується в широкому спектрі додатків, від цільової реклами та персоналізованих рекомендацій до покращення систем безпеки та нагляду. У маркетинговій сфері, це може допомогти компаніям краще зрозуміти свою аудиторію, а в галузі охорони здоров'я — дозволяє підвищити точність медичних діагностик.

Попри значний прогрес, аналіз віку та статі залишається складним завданням через різноманіття людських облич, включаючи варіації в освітленні, позах, емоціях та аксесуарах. Існує також виклик у забезпеченні точності прогнозування через різні вікові групи та етнічні особливості, що вимагає розробки більш універсальних та адаптивних систем.

В домені розпізнавання обличчя, сучасні технічні рішення інтегрують широкий спектр алгоритмічних підходів, що охоплюють шаблонний аналіз, виявлення структурних атрибутів обличчя та застосування методів глибинного навчання. Зокрема, конволюційні нейронні мережі (CNNs) та рекурентні нейронні мережі (RNNs) забезпечують передові можливості для автоматизації розпізнавання та класифікації облич за допомогою ідентифікації складних ознак із зображень. Новітні дослідження акцентують на створенні таких алгоритмів, які можуть адаптивно функціонувати з обмеженими наборами даних, що відкриває можливості для застосування в різноманітних

та динамічних умовах, підвищуючи ефективність та доступність технологій розпізнавання облич на глобальному рівні.

Оцінювання продуктивності моделей розпізнавання віку та статі здійснюється за допомогою ряду кількісних метрик. Для задач класифікації статі часто використовують точність (accuracy), F1-скор та AUC-ROC криву, які відображають збалансованість між чутливістю та специфічністю моделі. З іншого боку, аналіз віку, як правило, вимагає регресійних метрик, таких як середня абсолютна помилка (MAE) та середньоквадратична помилка (MSE), щоб виміряти відхилення прогнозованого віку від істинного значення.

Ефективні стратегії валідації є ключовими для оцінювання та покращення продуктивності моделей. Хрестова перевірка (cross-validation) дозволяє оцінити стійкість моделі до перенавчання, забезпечуючи її надійність на різних вибірках даних. Використання розділених наборів даних для тренування, валідації та тестування моделі дає змогу точно визначити її здатність узагальнювати на нових даних, що є критично важливим для реальних застосувань.

Виконання оцінки моделі у реальних сценаріях вимагає детального розуміння контексту застосування. Моделі повинні бути оцінені не лише з погляду статистичної продуктивності, але й здатності працювати в різних реальних умовах, включаючи різноманітність демографічних характеристик осіб та змінливість умов зйомки. Стрес-тестування на різних даних і в середовищах з різною освітленістю, позиціями обличчя та іншими можливими варіаціями є важливим для забезпечення міцності та адаптивності систем.

Глибинне навчання продовжує відкривати нові перспективи для розпізнавання образів, пропонуючи більш складні та ефективні моделі, які можуть використовуватись у широкому спектрі додатків. Міждисциплінарні дослідження, які об'єднують когнітивну науку, нейробіологію та комп'ютерні науки, обіцяють нові інновації в області. Однак, з розвитком цих технологій, зростає потреба в увазі до етичних питань, пов'язаних із конфіденційністю,

зловживанням та упередженістю в системах розпізнавання образів, що має стати предметом ретельного вивчення та регулювання.

1.2 Огляд існуючих рішень

Дослідження [1] розглядає методику передбачення віку та статі за допомогою алгоритмів глибокого навчання на основі зображень облич. У дослідженні використовуються передові техніки екстракції ознак та навчання моделей для оцінки точності класифікації. Результати виявили значущі переваги глибокого навчання в порівнянні з традиційними методами машинного навчання у контексті обробки зображень облич.

У дослідженні [2] представлено GRA_Net, інноваційну модель глибокого навчання для класифікації віку та статі на основі зображень облич. Модель використовує модифіковану мережу з увагою, що покращує здатність до розпізнавання складних ознак на обличчях за допомогою спеціально розробленого 'Gate' механізму.

Дослідження [3] зосереджується на використанні глибоких конволюційних нейронних мереж та трансферного навчання для передбачення віку та статі з зображень облич. Автори порівнюють різні архітектури мереж, включаючи VGG_f та ResNet50_f, демонструючи, як налаштування та оптимізація мережі можуть вплинути на точність класифікації.

Дослідження [4] вивчає застосування штучного інтелекту для передбачення віку та статі на основі зображень облич. Розроблена модель використовує комбінацію бібліотек глибокого навчання, таких як Keras і TensorFlow, з метою покращення точності та ефективності процесу класифікації.

У дослідженні [5] пропонується новий підхід до визначення віку та статі за допомогою глибоких навчальних мереж з механізмом уваги. Автори використовують увагу для підвищення точності розпізнавання, демонструючи

значні поліпшення в плані точності передбачення порівняно з існуючими методами.

Дослідження [6] аналізує ефективність систем оцінки віку та статі без використання фільтрів на зображеннях облич. Автори впроваджують методики глибокого навчання з використанням технік відмови навчання для покращення точності передбачення.

У дослідженні [7] представлено глибоке навчання з механізмом уваги для розпізнавання віку та статі "у дикій природі". Використання передтренированих конволюційних нейронних мереж для розпізнавання облич дозволило значно підвищити точність системи.

Дослідження [8] фокусується на використанні передавального навчання з глибокими конволюційними нейронними мережами для розпізнавання статі та оцінки віку. Зокрема, використання VGGFace та VGG16 з вагами для розпізнавання облич значно покращило результати.

Дослідження [9] розробляє покращений підхід до глибоких конволюційних нейронних мереж для оцінки віку та статі, використовуючи обличчя. Автори зосереджуються на точності та ефективності моделей у складних умовах, таких як зміна віку або особливості обличчя.

Далі проведемо порівняння плюсів і мінусів (Таблиця 1.1) для вибраних досліджень про передбачення віку та статі на основі зображень облич за допомогою алгоритмів глибокого навчання:

Таблиця 1.1 – Порівняння досліджень

Дослідження	Плюси	Мінуси
Haseena et al. (2022) [1]	Використовує інноваційні методи екстракції ознак.	Обмежена доступність деталей моделі.
Garain et al. (2021) [2]	Підтримка Residual Attention Network.	Висока складність реалізації.
Sheoran et al. (2021) [3]	Застосування трансферного навчання для покращення результатів.	Основою лише на двох моделях (VGG і ResNet).
Ghildiyal et al. (2020) [4]	Велика увага до точності класифікації.	Може потребувати значних обчислювальних ресурсів.
Abdolrashidi et al. (2020) [5]	Використання механізму уваги для підвищення точності.	Дослідження на ранній стадії, обмежена перевірка на

		різноманітних даних.
Eidinger et al. (2014) [6]	Розглядає "натуральні" зображення без фільтрації.	Відсутність деталей про датасети, використані у дослідженні.
Rodríguez et al. (2017) [7]	Глибокий аналіз із застосуванням механізму уваги.	Специфіка моделі може не працювати добре з усіма типами даних.
Smith & Chen (2018) [8]	Оцінювання використання VGGFace та VGG16.	Обмежена інформація щодо практичного застосування.
Sharma et al. (2022) [9]	Покращення точності завдяки покращеній CNN архітектурі.	Можлива перенавчаність моделі без належного регулювання.

З порівняльної таблиці 1.1 видно, що розробка моделей глибокого навчання для передбачення віку та статі за допомогою зображень обличчя постійно еволюціонує з використанням різноманітних підходів та технологій. Більшість досліджень демонструють високу ефективність завдяки інтеграції новітніх технік, таких як механізми уваги та трансферне навчання, але водночас стикаються з викликами, пов'язаними зі складністю моделей та їх адаптацією до різноманітних умов. Такий аналіз допомагає визначити потенціал кожного підходу в контексті специфічних додатків і можливостей подальшої оптимізації та покращення. Важливою є також потреба в уважному виборі датасетів та перевірці моделей на різноманітності даних, щоб забезпечити їхню універсальність та точність у реальних сценаріях використання.

Наше дослідження відрізняється інтеграцією найсучасніших технологій та алгоритмів глибокого навчання для створення більш ефективних та адаптивних моделей, які здатні точно визначати вік та стать особи на зображенні. Особливість нашої роботи полягає в застосуванні гібридних підходів, що об'єднують конволюційні нейронні мережі з механізмами уваги для підвищення точності та редукції помилок, особливо в умовах варіативності зовнішнього вигляду людей. Крім того, ми акцентуємо на оптимізації обчислювальних ресурсів, що робить наше дослідження не тільки

технологічно продуктивним, але й доступним для використання в реальних умовах із різним обладнанням.

1.3 Постановка задачі дослідження

Актуальність теми модуля розпізнавання віку та статі людей на основі нейронних мереж визначається її широким застосуванням в багатьох сферах, включаючи безпеку, маркетинг, охорону здоров'я та соціальні медіа. Розвиток технологій комп'ютерного зору та машинного навчання дозволив створювати все більш точні та ефективні системи, здатні аналізувати людські обличчя і визначати демографічні характеристики осіб з високою точністю. Відповідно, актуальність таких систем стає все більш відчутною на фоні зростаючого попиту на персоналізовані сервіси та засоби ідентифікації осіб.

Науковий інтерес до автоматизованого розпізнавання віку та статі є високим, оскільки ця область включає складні задачі обробки зображень і аналізу даних. Завдяки прогресу в алгоритмах глибокого навчання, сучасні дослідження зосереджуються на підвищенні точності і швидкості обробки даних, що в свою чергу відкриває нові можливості для реалізації автоматизованих систем, які можуть бути інтегровані в широкий спектр застосунків.

Однією з важливих аспектів застосування технологій розпізнавання віку та статі є підвищення рівня безпеки в громадських місцях та покращення систем відеоспостереження. Інтеграція нейронних мереж дозволяє розробляти системи, які можуть в реальному часі ідентифікувати осіб та аналізувати поведінку, сприяючи підвищенню ефективності правоохоронних заходів та систем безпеки.

У сфері маркетингу та реклами автоматизоване розпізнавання віку та статі відіграє ключову роль у персоналізації рекламних кампаній. Використання цих технологій дозволяє компаніям точно націлювати свої

рекламні звернення, оптимізуючи маркетингові стратегії та забезпечуючи вищу взаємодію з потенційними клієнтами. Це, в свою чергу, може суттєво підвищити ефективність рекламних витрат.

Крім того, область охорони здоров'я також може виграти від використання технологій розпізнавання віку та статі. Автоматизоване визначення цих параметрів може сприяти більш точній діагностиці та адаптації лікувальних процедур, зорієнтованих на певні вікові групи або статі, підвищуючи таким чином ефективність медичного обслуговування і допомоги.

Метою цієї бакалаврської роботи є розробка та оптимізація модуля розпізнавання віку та статі людей на основі нейронних мереж. Цей проект передбачає створення ефективної системи, здатної точно аналізувати зображення облич для визначення демографічних характеристик осіб, що включають вік та стать. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Аналіз предметної області та постановка задачі дослідження: Вивчення сучасного стану технологій розпізнавання облич та визначення ключових проблем та можливостей, які можуть бути вирішені за допомогою глибокого навчання.
2. Огляд існуючих рішень у сфері розпізнавання віку та статі: Проаналізувати поточні наукові та практичні підходи до розпізнавання віку та статі, включаючи традиційні методи і новітні розробки на основі машинного навчання.
3. Розробка алгоритмічного та інформаційного забезпечення системи: Спроектувати архітектуру модуля та вибрати архітектуру глибокої нейронної мережі, зокрема VGG16, ResNet152V2, та Xception, для розпізнавання віку та статі. Розробити алгоритм розпізнавання віку та статі.

4. Розробка програмно-технологічного забезпечення: Виконати підготовку даних, навчання моделі, оцінку отриманих результатів та оптимізацію моделі. Забезпечити документацію всього процесу розробки та результатів експериментів.
5. Аналіз результатів та валідація системи: Провести тестування розробленої системи на реальних даних, аналізувати її ефективність та точність. Запропонувати рекомендації щодо подальших напрямків дослідження та можливості комерціалізації проекту.

Завдання 1-2 мають бути розглянуті в теоретичних розділах роботи, а завдання 3-5 – у практичних розділах, що охоплюють розробку, оцінку та тестування алгоритму.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Архітектура модуля розпізнавання віку та статі людей

Архітектура модуля розпізнавання віку та статі людей базується на використанні нейронних мереж, що дозволяє ефективно обробляти великі набори даних зображень облич. Цей модуль складається з декількох ключових компонентів, кожен з яких відіграє важливу роль у процесі підготовки, тренування та оцінки моделі.

Архітектура модуля розпізнавання віку та статі людей структурована таким чином (Рисунок 3.1):

1. Налаштування середовища:

- **Імпорт:** Цей клас обробляє імпорт необхідних бібліотек та фреймворків, таких як TensorFlow, Keras, NumPy, і інших, потрібних для розробки моделі.
- **Константи:** Визначає ключові константи, такі як розмір зображення, розмір пакету та інші параметри, які використовуються у всьому модулі для обробки та тренування.

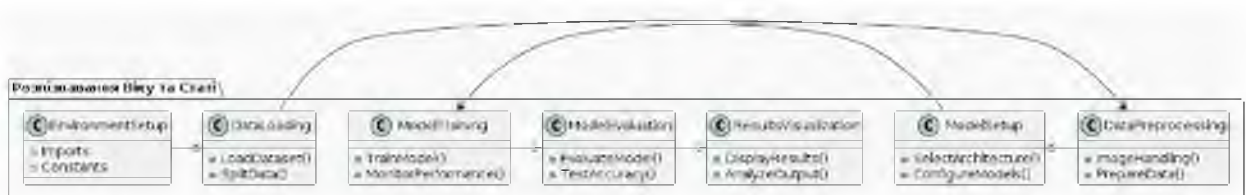


Рисунок 3.1 - Архітектура модуля розпізнавання віку та статі людей

2. Завантаження даних:

- **Завантаження набору даних (LoadDataset()):** Відповідає за завантаження набору даних UTKFace, який включає зображення з анотаціями віку, статі та етнічності.

- Розділення даних (SplitData()): Ця функція ділить набір даних на тренувальні, тестові та валідаційні набори, щоб забезпечити належне розподілення для тренування моделі та оцінки.

3. Попередня обробка даних:

- Обробка зображень (ImageHandling()): Реалізує функції обробки зображень, такі як зміна розміру, нормалізація та можливо аугментація, для підготовки зображень до ефективного тренування моделі.
- Підготовка даних (PrepareData()): Підготовлює фінальний набір даних, організовуючи оброблені зображення разом з їх мітками у форматі, придатному для тренування.

4. Налаштування моделі:

- Вибір архітектури (SelectArchitecture()): Вибирає архітектуру нейронної мережі (наприклад, VGG16, ResNet чи Xception), яка буде використовуватися як основа для завдань розпізнавання.
- Конфігурація моделей (ConfigureModels()): Налаштовує вибрану модель для конкретних завдань, налаштовуючи шари, функції активації та компілюючи модель з відповідними функціями втрат та оптимізаторами.

5. Тренування моделі:

- Тренування моделі (TrainModel()): Керує процесом тренування моделі, використовуючи підготовлені набори даних, застосовуючи колбеки, такі як рання зупинка та контроль точок збереження моделі, для оптимізації тренування.
- Моніторинг продуктивності (MonitorPerformance()): Слідкує та реєструє продуктивність моделі під час тренування для відстеження покращень і внесення необхідних коригувань.

6. Оцінка моделі:

- Оцінка моделі (EvaluateModel()): Оцінює продуктивність навченої моделі на окремому тестовому наборі, щоб виміряти її ефективність у реальних сценаріях.
- Тестування точності (TestAccuracy()): Конкретно вимірює точність моделі з точки зору передбачення віку та класифікації статі.

7. Візуалізація результатів:

- Відображення результатів (DisplayResults()): Візуалізує результати передбачень моделі за допомогою графіків та діаграм для ясного уявлення передбачень моделі у порівнянні з реальними мітками.
- Аналіз результатів (AnalyzeOutput()): Забезпечує глибший аналіз результатів, виявляючи тенденції, упередження або області, де модель може показувати недостатню продуктивність.

Цей структурований підхід дозволяє систематично розробляти і оцінювати надійну модель, здатну точно розпізнавати вік і стать з обличчя. Кожен компонент в архітектурі відіграє важливу роль у забезпеченні не тільки точності, але й загальної застосовності моделі до нових, невідомих даних.

2.2 Моделі для передбачення віку

2.2.1 VGG16

Модель VGG16, розроблена Оксфордським університетом, є однією з відомих конволюційних нейронних мереж (CNN), яка досягла значних успіхів у змаганні ImageNet. Вона включає 16 шарів, що мають ваги, у тому числі 13 конволюційних шарів та 3 повнозв'язних шари. Архітектура VGG16 вирізняється своєю однорідністю: вона використовує лише конволюційні фільтри розміром $3 \times 3 \times 3$ з кроком 1 та падінгом 1, а також максимальні

пулінги розміром $2 \times 2 \times 2$ з кроком 2. Це дозволяє ефективно накопичувати відомості про особливості на різних рівнях в глибину мережі. Функція активації в моделі – ReLU (Rectified Linear Unit), що застосовується після кожного конволюційного шару.

Математично, конволюційний шар можна описати як:

$$f(i, j) = \sigma \left(\sum_{u=0}^{U-1} \sum_{v=0}^{V-1} x(i+u, j+v) \cdot w(u, v) + b \right)$$

де x – вхідні дані, w – ядро фільтра, b – зміщення, а σ – функція активації ReLU.

2.2.2 ResNet152V2

ResNet152V2 є частиною родини ResNet (Residual Networks), які використовують ідею пропускових з'єднань або залишкових блоків для боротьби з проблемою згортання градієнтів в глибоких мережах. У цій архітектурі, сигнал від вхідних даних або попереднього шару може "пропускатися" через один або кілька шарів і додаватися до виходу пізніших шарів, що дозволяє тренування дуже глибоких мереж.

Залишковий блок можна представити як:

$$y = F(x, \{W_i\}) + x$$

де x і y – вхід і вихід блоку відповідно, F – залишкова функція, зазвичай складається з кількох конволюційних шарів, а $\{W_i\}$ – набір вагових матриць цих шарів.

2.2.3 Xception

Xception, модель розроблена Google, є розширенням архітектури Inception, яка використовує концепцію глибоких згорток, де просторове відділення і каналне відділення обробляються окремо. Це дозволяє мережі вчитися більш ефективно та вимагає менше параметрів в порівнянні з традиційними конволюційними підходами.

У Xception основною операцією є глибока згортка, яка описується як:

$$y'_k = \sum_{i,j} x_{i,j} \cdot W_k(i,j)$$

де x – вхідні дані, W_k – вагова матриця для каналу k , а індекси i,j проходять по просторових вимірах. За цим слідує точкова згортка, яка комбінує виходи глибоких згорток різних каналів.

Ці моделі створюють міцну основу для складних завдань аналізу зображень завдяки їх архітектурним особливостям та здатності ефективно обробляти великі обсяги даних із зображень.

2.3 Моделі для класифікації статі

2.3.1 *MobileNetV3Small та MobileNetV3Large*

MobileNetV3, розроблені Google, представляють собою набір моделей, оптимізованих для високої продуктивності на мобільних пристроях. Ці моделі використовують архітектуру, що базується на ідеї глибоких згорток та включає спеціальні блоки, які називаються "bottleneck layers". MobileNetV3Small та MobileNetV3Large відрізняються кількістю шарів та глибиною, що дозволяє адаптувати їх до різних обчислювальних потужностей.

Глибинна згортка у MobileNetV3 описується як:

$$y_k = \sigma \left(\sum_{i,j} K(i,j) \cdot x_{k,i,j} + b_k \right)$$

де $x_{k,i,j}$ - вхід каналу k у пікселі (i,j) , $K(i,j)$ - ядро фільтру для каналу, b_k - зміщення для каналу, а σ - функція активації, зазвичай ReLU.

2.3.2 *InceptionV3*

InceptionV3 є поліпшенням попередніх версій Inception, яке включає оптимізації для зменшення обчислювального навантаження, такі як розширення конволюційних фільтрів у формі "факторизації" замість великих фільтрів. Основна особливість Inception - це "модулі Inception", що дозволяють мережі вибирати оптимальні фільтри на кожному шарі.

Модуль Inception можна математично виразити як злиття кількох згорткових операцій:

$$y = [y(1), y(2), y(3), y(4)]$$

де кожен $y(n)$ - це вихід згорткового шару з різним розміром ядра, а згортки виконуються паралельно та об'єднуються разом.

2.3.3 ResNet50V2

ResNet50V2 є вдосконаленою версією оригінальної моделі ResNet. Ця модель включає удосконалення, такі як використання попередньої активації у залишкових блоках, що покращує нормалізацію та розподіл обучення. ResNet50V2 використовує 50 шарів і зберігає ідею залишкових з'єднань для боротьби з затуханням градієнтів.

Залишковий блок у ResNet50V2 може бути описаний як:

$$y = x + F(x, \{W_i\})$$

де F - функція, яка представляє послідовність згорток та активацій, а залишкове з'єднання дозволяє прямий прохід вхідного сигналу x через блок, що покращує здатність мережі до навчання.

Ці моделі демонструють різні підходи до архітектури нейронних мереж та їх оптимізацію для ефективної роботи з великими масивами даних та обмеженими обчислювальними ресурсами, що є критично важливим для застосувань у реальному часі та на мобільних пристроях.

2.4 Алгоритм розпізнавання віку та статі людей на основі нейронних мереж

У сучасному світі технологій розпізнавання облич стає все більш важливим інструментом у різних галузях, від безпеки до персоналізації послуг. Нейронні мережі відіграють ключову роль у цих процесах, забезпечуючи точність і ефективність у визначенні особливостей облич, таких як вік і стать людини. Детальний алгоритм, використовує передові методи

глибокого навчання для аналізу зображень облич, інтегруючи різноманітні технічні підходи та інструменти.

Далі представлено по-кроковий алгоритм розпізнавання віку та статі людей на основі нейронних мереж (Рисунок 2.2, див. додаток А):

1. Підготовка середовища:

- Імпорт необхідних бібліотек та модулів (NumPy, Pandas, TensorFlow, Keras, matplotlib).
- Налаштування сталих параметрів (розмір зображень, швидкість навчання).

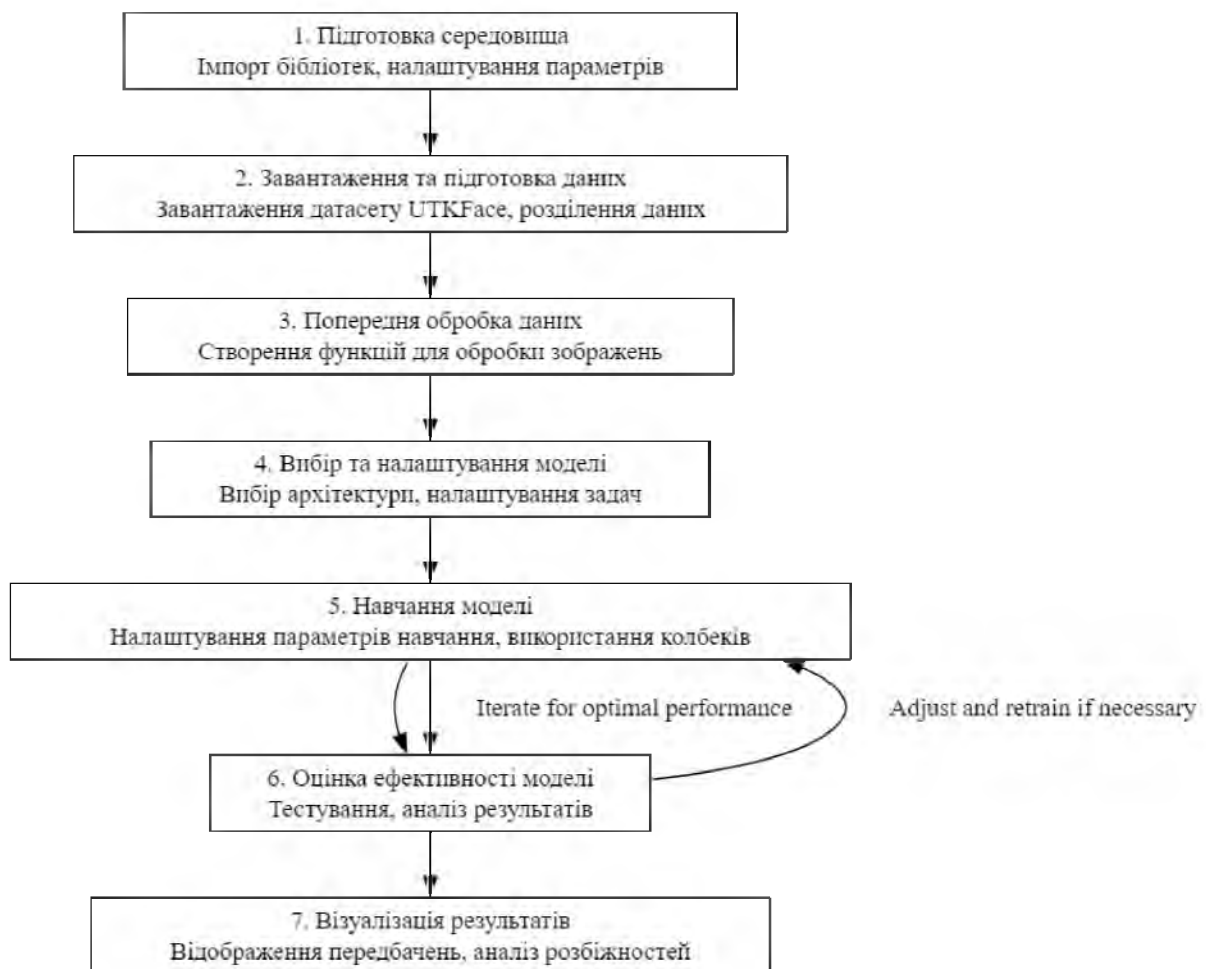


Рисунок 2.2 – Схема алгоритму розпізнавання віку та статі людей на основі нейронних мереж

2. Завантаження та підготовка даних:

- Завантаження датасету UTKFace, який містить зображення облич з мітками віку та статі.

- Розділення шляхів до зображень на тренувальні, тестові та валідаційні набори.
3. Попередня обробка даних:
 - Створення функцій для завантаження та обробки зображень: масштабування та нормалізація.
 - Формування TensorFlow датасетів для ефективної роботи з великою кількістю зображень.
 4. Вибір та налаштування моделі:
 - Вибір базових моделей для архітектури (наприклад, VGG16, ResNet).
 - Налаштування моделей для задач регресії віку та класифікації статі.
 5. Навчання моделі:
 - Налаштування параметрів навчання (функції втрат, оптимізатор).
 - Використання колбеків для ранньої зупинки та збереження кращих вагів.
 6. Оцінка ефективності моделі:
 - Тестування моделей на тестовому наборі.
 - Аналіз результатів для виявлення найкращої моделі.
 7. Візуалізація результатів:
 - Відображення прикладів передбачень моделей на тестових зображеннях.
 - Аналіз розбіжностей між реальними та передбачуваними значеннями.

Цей алгоритм використовує поєднання підходів з різних бібліотек та технік глибокого навчання для створення ефективної системи розпізнавання віку та статі людей.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Опис набору даних

Для розробки та випробування алгоритму розпізнавання віку та статі особи застосовуватимуться два набори даних. Перший набір, UTKFace, є великою колекцією зображень облич людей різного віку (від 0 до 116 років), статі та етнічної приналежності, що охоплює широкий спектр постав, виразів облич, освітлення та інших змінних, що можуть впливати на розпізнавання. Цей набір містить понад 20,000 оброблених зображень і часто використовується для завдань детекції облич, оцінювання віку, прогресування та регресії віку, локалізації особливих точок та багатьох інших досліджень у галузі обробки зображень і машинного зору. Різноманітність умов зображень робить UTKFace ідеальним джерелом для тренування та тестування алгоритмів у реалістичних сценаріях, забезпечуючи міцну основу для оцінки їх загальної ефективності і адаптації.

Другий набір даних, "Age and Gender Prediction Models: Trained Weights and Visualizations Dataset", включає треновані ваги для моделей на основі VGG16 та ResNet152V2, спеціально адаптовані для задач розпізнавання віку та статі. Цей набір даних містить також візуалізації навчальних кривих, приклади даних та інші корисні зображення, які допомагають у глибокому аналізі ефективності алгоритмів. Завдяки добре документованій структурі цього набору, дослідники можуть легко відтворити експерименти та адаптувати моделі під свої унікальні потреби, аналізуючи та порівнюючи різні архітектури та їх вплив на точність прогнозування.

Під час роботи з UTKFace, завантаження та обробка цих даних виявили ключові особливості розподілу віку та статі, що демонструється на гістограмі розподілу віку та діаграмах розподілу статі. Аналіз (рис.3.1) розподілу статі та віку в наборі даних UTKFace виявив цікаві відмінності. Згідно з круговою

діаграмою, чоловіки становлять більшу частину даних, з невеликою перевагою 52.3% порівняно з жінками, які становлять 47.7%. Стовпчикова діаграма підтверджує цю тенденцію, демонструючи чисельність обох груп у великому діапазоні. Дані про вік, представлені на гістограмі, показують два виділених піку: перший найбільший пік близько 1 року з приблизно 1123 зображеннями та другий значний пік біля віку 26 років з 2197 зображеннями. Віолончельовий графік та боксплот підтверджують бімодальний розподіл віку в даних, де віолончельовий графік наголошує на широкому розподілі в молодшій віковій групі та зменшення виразності у старшій. Боксплот показує, що медіана віку зосереджена нижче 30 років, що свідчить про перекид у бік молодших вікових груп.



Рисунок 3.1 – Розподіл даних в наборі UTKFace

Додатковий аналіз демонструє, як моделі, треновані на цьому наборі даних, можуть відображати або відтворювати існуючі упередження в даних, що має важливе значення для розробки справедливих та точних систем розпізнавання облич (Рис.3.2). Під час візуалізації, зображення з вибірковими прикладами облич показують різноманітність даних і їх виклики, як зазначено на зображенні вибіркових даних. Ці зображення допомагають зрозуміти, як алгоритми аналізують і класифікують різні вікові категорії та статі, що є критично важливим для подальших етапів проектування та оптимізації системи розпізнавання.



Рисунок 3.2 - Вибіркові приклади облич

3.2 Аналіз архітектур базових моделей

При виборі оптимальної базової моделі для розробки системи розпізнавання віку та статі особи, було здійснено детальне порівняння декількох кандидатів. За допомогою засобів візуалізації тренувальних та

валідаційних втрат, встановлено, що модель VGG16 показала найкращу здатність до навчання, з постійним зменшенням помилки, досягаючи середньої абсолютної помилки приблизно 8.0, як видно на графіку тренувальних втрат (Рис.3.3). У порівнянні, модель ResNet152V2 продемонструвала високу точність як на тренувальних, так і на валідаційних даних, що вказує на її ефективність для задачі класифікації статі, як ілюстровано на графіку точності (Рис.3.4).

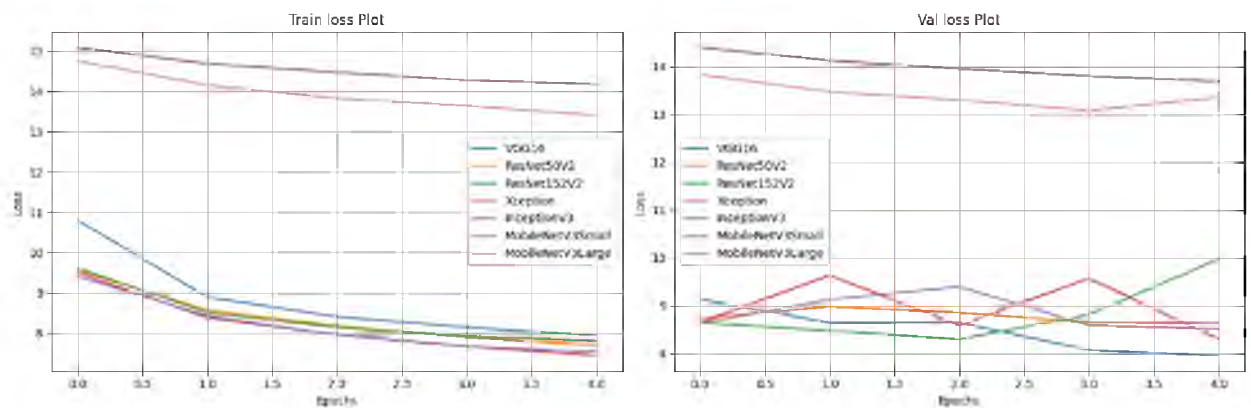


Рисунок 3.3 - Графік тренувальних втрат

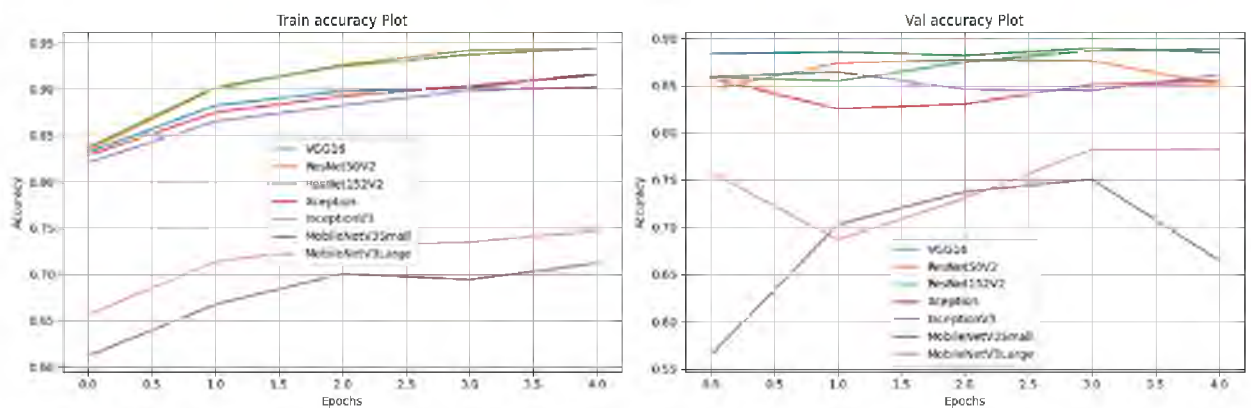


Рисунок 3.4 - Графік точності

Моделі MobileNetV3Small та MobileNetV3Large продемонстрували найменшу продуктивність з точки зору втрат та точності на валідаційних даних, що вказує на їхню непридатність для складних завдань розпізнавання. Ці результати підкреслюють важливість вибору відповідної архітектури для конкретних вимог завдань та масштабів даних.

3.3 Оптимізація моделі вікової класифікації

У процесі визначення оптимальної базової моделі для задачі прогнозування віку особи за допомогою зображень обличчя, VGG16 підтвердила свої переваги, про що свідчить аналіз навчальних кривих. Початкова середня абсолютна помилка (MAE) нашої моделі з VGG16 на початку навчання була близько 9.0, яка поступово знизилася до 6.7 після 13 епох навчання, що видно на графіку навчальних кривих (рис.3.5). Це свідчить про здатність моделі ефективно адаптуватися і точно прогнозувати вік на основі даних про обличчя.

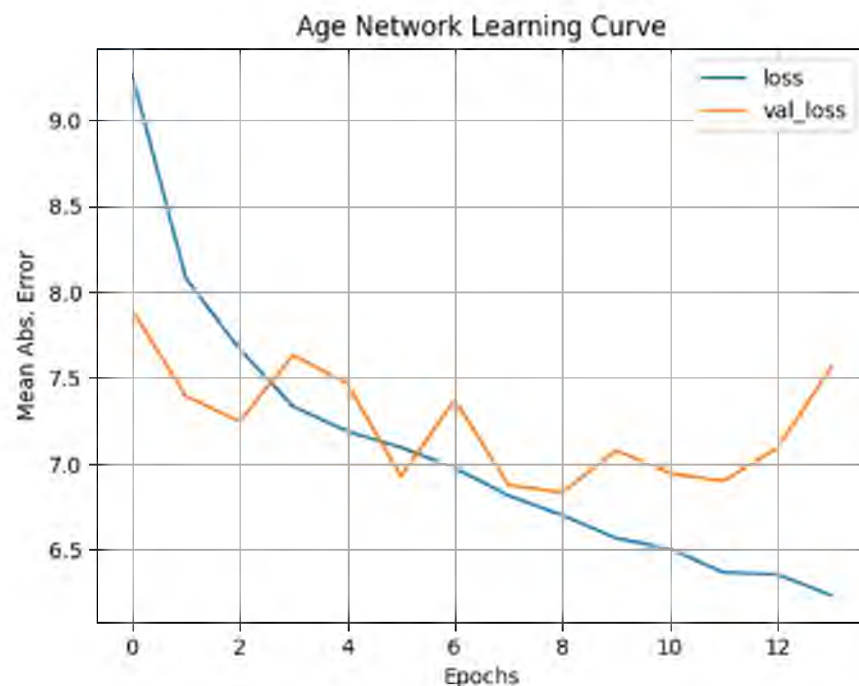


Рисунок 3.5 - Графік навчальних кривих

Валідаційна втрата зазнала коливань, але залишалася в діапазоні близько 6.8-7.5, що підтверджує адекватність моделі до різних вибірок даних. Раннє припинення навчання (early stopping) впроваджено як захід проти перенавчання, дозволяючи моделі уникнути переоснащення і залишатися гнучкою до нових даних. Моментом вибору найкращої версії моделі стала 7

епоха, де було зафіксовано найменшу різницю між тренувальними та валідаційними втратами.

3.4 Оптимізація моделі гендерної класифікації

Модель розпізнавання статі на основі ResNet152V2 демонструє суттєвий прогрес у навчанні та валідації, що ілюструється зменшенням втрати від початкових 0.3888 до 0.2671 після 20 епох навчання, як видно на графіку навчання (рис.3.6). Валідаційна втрата також показує позитивну тенденцію до зниження, свідчаючи про хорошу узагальнюючу здатність моделі. З точки зору точності, модель досягає високого рівня, стабілізуючи валідаційну точність на рівні 87.64% у порівнянні з тренувальною точністю, що виходить на 88.63%.

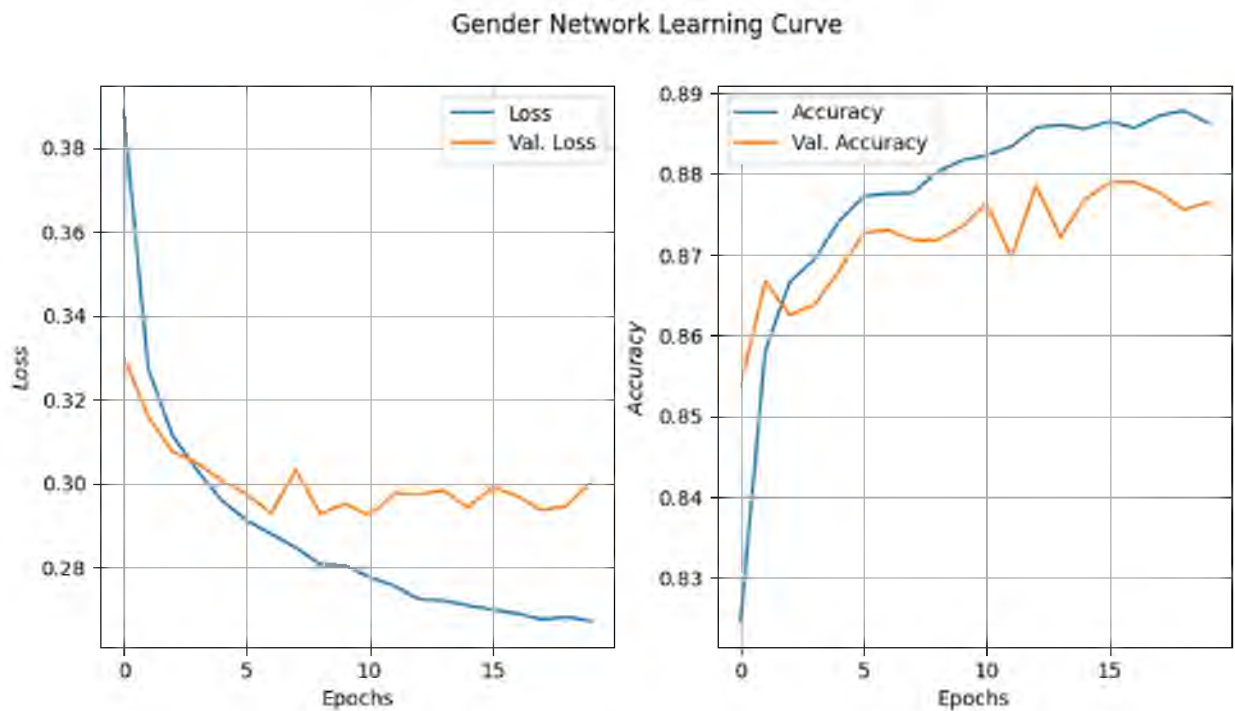


Рисунок 3.6 - Графік навчання

Структурні корективи у моделі гендерної класифікації, такі як заміна плоского шару (Flatten layer) на глобальний середньопулінг (GlobalAveragePooling), сприяли підвищенню стійкості моделі до перенавчання. Попри деяке зниження тренувальної точності, валідаційні

показники зберегли високу робастність. Застосування ранньої зупинки (early stopping) та збереження кращих варіантів (model checkpointing) дозволило вчасно відреагувати на ознаки перенавчання, що з'явилися близько 12-13 епох, забезпечуючи вибір найкращої моделі для подальшої експлуатації.

3.5 Емпіричний аналіз передбачень моделей

В контексті останнього етапу оцінювання моделі, виконано ретельний аналіз результатів тестування нейронної мережі для прогнозування віку та статі. Використання моделі, заснованої на архітектурі ResNet152V2, демонструє суттєвий прогрес у точності та здатність моделі до генералізації. На тестовому наборі даних, валідаційна втрата для моделі розпізнавання статі була знижена до 0.2540, а точність досягла 89.84%, що підкреслює високу надійність та узагальнюючу здатність моделі.

Прогнозування віку осіб, засноване на тих же засадах, показало, що валідаційна втрата на тестових даних досягла показника 5.9889. Це вказує на достатньо високу якість навчання та здатність моделі точно оцінювати вік на основі зображення обличчя, що є важливим для практичного застосування в різних додатках, які вимагають вікової верифікації або аналізу демографічних характеристик.

Графічне представлення результатів, отриманих від моделей, надає значущий внесок у оцінювання їх продуктивності. Візуалізація передбачень, надана у рисунку 3.7 прогнозування, дозволяє бачити як модель визначає вік та стать різних осіб, починаючи від немовлят до літніх людей. Для більшості індивідуальних випадків, модель виконує точне ідентифікування статі та віку, але залишаються складнощі при розпізнаванні малюків, де діапазони помилок є вищими.

Застосування моделей у реальних умовах вимагає не лише високої точності, але й здатності до адаптації до різноманітних сценаріїв. Хоча тестові дані демонструють обнадійливі результати, розглядається потреба у

додаткових перевірок на більш широкому та різноманітному наборі зображень. Це допоможе забезпечити, що модель буде коректно функціонувати у різних середовищах та з різними групами населення.



Рисунок 3.7 – Прогноз результатів

Підсумовуючи, результати тестування та валідації свідчать про успішне налаштування моделей. Водночас важливо визнати обмеження цих моделей, особливо у контексті викликів, пов'язаних із розпізнаванням віку у молодших вікових категоріях. Інтеграція спеціалізованих підходів та розширення набору даних можуть бути наступними кроками для подальшого вдосконалення моделей.

ВИСНОВКИ

У підсумку, розпізнавання образів займає центральне місце в сучасному комп'ютерному світі, забезпечуючи широкий спектр можливостей у різних сферах життя, від безпеки та медицини до маркетингу та транспорту. Завдяки поєднанню новітніх технологій глибокого навчання з вдосконаленням алгоритмів і обчислювальних ресурсів, сучасні моделі розпізнавання облич можуть ефективно функціонувати в різних умовах та відповідати високим стандартам точності та швидкості. Однак, неупинний розвиток цієї галузі вимагає постійного удосконалення та уваги до етичних питань, що стає предметом ретельного вивчення та регулювання. Наше дослідження, спрямоване на оптимізацію та розробку передових моделей розпізнавання віку та статі на основі глибокого навчання, має потенціал не лише революціонізувати цю галузь, але й позитивно вплинути на різноманітні аспекти нашого повсякденного життя.

У результаті, розроблена архітектура модуля розпізнавання віку та статі людей на основі нейронних мереж відображає систематичний підхід до обробки та аналізу зображень облич. Вона включає в себе етапи підготовки даних, налаштування моделі, тренування та оцінки її ефективності. Кожен компонент цієї архітектури відіграє важливу роль у забезпеченні якості та надійності моделі, що дозволяє точно розпізнавати вік та стать осіб на зображеннях. Застосування цього алгоритму в практичних задачах може значно полегшити процес автоматичного аналізу облич і мати широкий спектр застосувань у різних сферах життя.

У цьому дослідженні було проведено докладний аналіз архітектур базових моделей для розробки системи розпізнавання віку та статі особи на основі зображень облич. Результати показали, що модель VGG16 виявилася найбільш ефективною для вирішення завдання розпізнавання віку, тоді як модель ResNet152V2 демонструвала високу точність у класифікації статі.

Оптимізація моделей була здійснена шляхом налаштування параметрів навчання та використання раннього припинення навчання для уникнення перенавчання. Емпіричний аналіз передбачень моделей підтвердив їхню високу точність, особливо в контексті класифікації статі, хоча було виявлено деякі складнощі при розпізнаванні віку у молодших категоріях. В цілому, результати свідчать про успішне налаштування моделей, але для подальшого вдосконалення може бути необхідно розширити набір даних та вдосконалити підходи до розпізнавання віку у молодших групах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Haseena, S., Saroja, S., Madavan, R., & Karthick, A. (2022). Prediction of the Age and Gender Based on Human Face Images Based on Deep Learning Algorithm. *Computational and Mathematical Methods in Medicine*. <https://www.hindawi.com/journals/cmmm/2022/1413597/>
2. Garain, A., Ray, B., Singh, P. K., Ahmadian, A., & Senu, N. (2021). GRA_Net: A Deep Learning Model for Classification of Age and Gender from Facial Images. *IEEE Transactions*. <https://ieeexplore.ieee.org/abstract/document/9446083/>
3. Sheoran, V., Joshi, S., & Bhayani, T. R. (2021). Age and Gender Prediction Using Deep CNNs and Transfer Learning. *Computer Vision and Image Processing*. https://link.springer.com/chapter/10.1007/978-981-16-1092-9_25
4. Ghildiyal, A., Sharma, S., & Verma, I. (2020). Age and Gender Predictions Using Artificial Intelligence Algorithm. *2020 3rd International Conference on Information and Computer Technologies*. <https://ieeexplore.ieee.org/abstract/document/9316053/>
5. Abdolrashidi, A., Minaei, M., Azimi, E., & Minaee, S. (2020). Age and Gender Prediction from Face Images Using Attentional Convolutional Network. *arXiv preprint arXiv:2010.03791*. <https://arxiv.org/abs/2010.03791>
6. Eidingen, E., Enbar, R., & Hassner, T. (2014). Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*. Retrieved from https://www.openu.ac.il/home/hassner/Adience/EidingenEnbarHassner_tifs.pdf
7. Rodríguez, P., Cucurull, G., Gonfaus, J. M., Roca, F. X., & González, J. (2017). Age and gender recognition in the wild with deep attention. *Pattern*

- Recognition, 70, 1-14. Retrieved from <https://refbase.cvc.uab.es/files/RCG2017b.pdf>.
8. Smith, P., & Chen, C. (2018). Transfer learning with deep CNNs for gender recognition and age estimation. 2018 IEEE International Conference on Big Data. Retrieved from <https://arxiv.org/pdf/1811.07344>.
 9. Sharma, N., Sharma, R., & Jindal, N. (2022). Face-based age and gender estimation using improved convolutional neural network approach. *Wireless Personal Communications*, 122(3), 2683-2701. <https://doi.org/10.1007/s11277-022-09501-8>.
 10. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.
 11. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Лип'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

ДОДАТОК А

ПСЕВДОКОД АЛГОРИТМУ

1. Initialize Environment
 - Import Libraries:
 - os, keras, numpy as np, pandas as pd, tensorflow as tf
 - plotly.express as px, matplotlib.pyplot as plt
 - Set Constants:
 - BATCH_SIZE = 32
 - IMAGE_SIZE = 224
 2. Load and Prepare Data
 - Load UTKFace Dataset from Directory:
 - dir_path = "/kaggle/input/utkface-new/UTKFace/"
 - image_paths = os.listdir(dir_path)
 - Initialize Gender Mapping:
 - gender_mapping = ["Male", "Female"]
 3. Preprocess Data
 - Define Functions:
 - preprocess_image(image_path, age, gender) -> (image, age, gender)
 - Load Image
 - Decode and Resize Image to (IMAGE_SIZE, IMAGE_SIZE)
 - Normalize Image Pixel Values
 - Generate TensorFlow Datasets:
 - Shuffle image_paths
 - Split into train, test, and validation sets
 - Use tf.data.Dataset to manage batching and shuffling
 4. Select and Setup Model
 - Initialize Backbone Models:
 - VGG16, ResNet50, etc.
 - Setup Model Architectures:
 - For Age Prediction:
 - Use a regression layer for age output
 - For Gender Prediction:
 - Use a classification layer with sigmoid activation for gender output
 5. Train Models
 - Compile Models:
 - Loss Function: MAE for age, Binary Crossentropy for gender
 - Optimizer: Adam
 - Implement Callbacks:
 - EarlyStopping on validation loss
 - ModelCheckpoint to save the best model weights
 - Train Models:
 - Fit models on training data
 - Validate using validation data
 6. Evaluate Model Performance
 - Load Best Models:
 - Load models with best weights from checkpoints
 - Test Models:
 - Evaluate models on test dataset
 - Print accuracy and loss metrics
 7. Visualize Predictions
 - For a batch of test images:
 - Predict age and gender
 - Display images with predicted and actual labels
- End Algorithm

ДОДАТОК Б

КОД ДЛЯ РЕАЛІЗАЦІЇ

```

# %% [code] {"id": "8Y0_sYPLdTEb", "execution": {"iopub.status.busy": "2024-02-10T08:59:53.573845Z", "iopub.execute_input": "2024-02-10T08:59:53.574265Z", "iopub.status.idle": "2024-02-10T09:00:12.623343Z", "shell.execute_reply.started": "2024-02-10T08:59:53.574233Z", "shell.execute_reply": "2024-02-10T09:00:12.621991Z"}}
# General
import os
import keras
import numpy as np
import pandas as pd
import tensorflow as tf
# Data
import plotly.express as px
import matplotlib.pyplot as plt
# Data Preprocessing
import tensorflow.data as tfds
from sklearn.model_selection import train_test_split
# Model
from keras.applications import VGG16
from keras.applications import Xception, InceptionV3
from keras.applications import ResNet50V2, ResNet152V2
from keras.applications import MobileNetV3Small, MobileNetV3Large
# Model training
from keras import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten, GlobalAveragePooling2D
from keras.layers import InputLayer
# Model Callbacks
from keras.callbacks import EarlyStopping
from keras.callbacks import ModelCheckpoint
# %% [code] {"id": "LvF-vjKxeBww", "execution": {"iopub.status.busy": "2024-02-10T09:00:12.625737Z", "iopub.execute_input": "2024-02-10T09:00:12.626511Z", "iopub.status.idle": "2024-02-10T09:00:12.632451Z", "shell.execute_reply.started": "2024-02-10T09:00:12.626476Z", "shell.execute_reply": "2024-02-10T09:00:12.631212Z"}}
# Setting constants for reproducibility
np.random.seed(42)
tf.random.set_seed(42)
# Constants
BATCH_SIZE = 32
IMAGE_SIZE = 224
# %% [code] {"id": "DHkgJyg2bjlw", "outputId": "5c93959d-327f-4867-d7f9-045b15b0b534", "execution": {"iopub.status.busy": "2024-02-10T09:00:12.634072Z", "iopub.execute_input": "2024-02-10T09:00:12.635184Z", "iopub.status.idle": "2024-02-10T09:00:13.944234Z", "shell.execute_reply.started": "2024-02-10T09:00:12.635123Z", "shell.execute_reply": "2024-02-10T09:00:13.943367Z"}}
# Initialize the directory path
dir_path = "/kaggle/input/utkface-new/UTKFace/"
image_paths = os.listdir(dir_path)
# Initialize a Gender Mapping
gender_mapping = ["Male", "Female"]
# Choose and load an image randomly
rand_image_path = np.random.choice(image_paths)
rand_image = plt.imread(dir_path + rand_image_path)/255.
sample_age, sample_gender, *_ = rand_image_path.split("_")
print(f"Total number of images : {len(image_paths)}")
print(f"Sample Image path : {rand_image_path}")
print(f"Sample Age : {sample_age}")
print(f"Sample Gender : {gender_mapping[int(sample_gender)]}\n")
# Show the image
plt.figure(figsize = (5,5))
plt.title("Sample Image")
plt.imshow(rand_image)
plt.axis("off")
plt.show()
# %% [markdown] {"id": "2SEhYdu9jlw8"}
# %% [code] {"id": "1_44RCxmjlP", "outputId": "ed70067b-95c8-4419-9fca-49827ed0c695", "execution": {"iopub.status.busy": "2024-02-10T09:00:13.94654Z", "iopub.execute_input": "2024-02-10T09:00:13.947094Z", "iopub.status.idle": "2024-02-10T09:00:16.578694Z", "shell.execute_reply.started": "2024-02-10T09:00:13.947062Z", "shell.execute_reply": "2024-02-10T09:00:16.577491Z"}}
# Initialize a male counter variable
male_count = 0
# Initialize variable to store all the ages.
ages = []
# Loop over the paths and check for male images.
for path in image_paths:
    path_split = path.split("_")
    if "0" == path_split[1]:
        male_count += 1
    ages.append(int(path_split[0]))
# Compute total female counts
female_count = len(image_paths) - male_count
# Visualizing The Class Imbalance
pie_chart = px.pie(
    names = gender_mapping,
    values = [male_count, female_count],
    hole = 0.4,
    title = "Gender Distribution (Donut Chart)",
    height = 500
)

```

```

pie_chart.show()
bar_graph = px.bar(
    y = gender_mapping,
    x = [male_count, female_count],
    title = "Gender Distribution (Bar Graph)",
    color = gender_mapping,
    height = 500
)
bar_graph.update_layout(
    yaxis_title = "Gender",
    xaxis_title = "Frequency Count"
)
bar_graph.show()
# %% [markdown] {"id": "Bvp2Gt0l6RR"}
# %% [code] {"id": "S595_TTlmXfT", "outputId": "4b039aa1-c13e-47cb-e4f5-06a1bb6656ec", "execution": {"iopub.status.busy": "2024-02-10T09:00:16.580203Z", "iopub.execute_input": "2024-02-10T09:00:16.580744Z", "iopub.status.idle": "2024-02-10T09:00:17.023061Z", "shell.execute_reply.started": "2024-02-10T09:00:16.580696Z", "shell.execute_reply": "2024-02-10T09:00:17.021744Z"}}
# Histogram
fig = px.histogram(sorted(ages), title = "Age Distribution")
fig.update_layout(
    xaxis_title = "Age",
    yaxis_title = "Value Counts"
)
fig.show()
# Violin Plot
fig = px.violin(x = sorted(ages), title = "Age Distribution")
fig.update_layout(
    xaxis_title = "Age",
    yaxis_title = "Distribution"
)
fig.show()
# Box Plot
fig = px.box(x = sorted(ages), notched=True, title = "Age Distribution")
fig.update_layout(
    xaxis_title = "Age",
)
fig.show()
# %% [markdown] {"id": "2d61l3YMkM3C"}
# %% [code] {"id": "qG52SXVridSr", "outputId": "87745e54-9b31-440c-843c-86164432bef6", "execution": {"iopub.status.busy": "2024-02-10T09:00:17.024487Z", "iopub.execute_input": "2024-02-10T09:00:17.02487Z", "iopub.status.idle": "2024-02-10T09:00:17.092801Z", "shell.execute_reply.started": "2024-02-10T09:00:17.024838Z", "shell.execute_reply": "2024-02-10T09:00:17.091489Z"}}
# SHuffling the Images
np.random.shuffle(image_paths)
# Split data into training, testing and validation set
train_images, test_images = train_test_split(
    image_paths,
    train_size = 0.9,
    test_size = 0.1
)
train_images, valid_images = train_test_split(
    image_paths,
    train_size = 0.9,
    test_size = 0.1
)
print(f"Training Size : {len(train_images)}")
print(f"Testing Size : {len(test_images)}")
# Extract age and gender
train_ages = [int(path.split("_")[0]) for path in train_images]
train_genders = [int(path.split("_")[1]) for path in train_images]
valid_ages = [int(path.split("_")[0]) for path in valid_images]
valid_genders = [int(path.split("_")[1]) for path in valid_images]
test_ages = [int(path.split("_")[0]) for path in test_images]
test_genders = [int(path.split("_")[1]) for path in test_images]
# %% [markdown] {"id": "3707BmRHsTHq"}
# Because we have a **huge number of images**, it will be better to use **Tensorflow data set** for **efficient processing**.
# %% [code] {"id": "nXR61jOPsz98", "execution": {"iopub.status.busy": "2024-02-10T09:00:17.094605Z", "iopub.execute_input": "2024-02-10T09:00:17.095027Z", "iopub.status.idle": "2024-02-10T09:00:17.108801Z", "shell.execute_reply.started": "2024-02-10T09:00:17.094995Z", "shell.execute_reply": "2024-02-10T09:00:17.10729Z"}}
def show_image(image, show=False):
    """
    Displays the provided image without axis.
    Args:
    - image (array-like): The image data to be displayed.
    - show (bool): If True, displays the image immediately. Defaults to False.
    Returns:
    - None
    """
    plt.imshow(image)
    plt.axis("off")
    if show:
        plt.show()

def preprocess_age_data(image_path, age, gender, dir_path=dir_path, IMAGE_SIZE = IMAGE_SIZE):
    """

```

```

Args:
- image_path (str): The path to the image file.
- dir_path (str): The directory path where the image is located. Defaults to `dir_path`.
Returns:
- tuple: A tuple containing the preprocessed image as a TensorFlow tensor,
        the age (int), and the gender (int) extracted from the image path.
"""

# Load the Image
image = tf.io.read_file(dir_path + image_path)
image = tf.io.decode_jpeg(image)
# Resize and Normalize the Image
image = tf.image.resize(image, (IMAGE_SIZE, IMAGE_SIZE))
image = image / 255.
image = tf.cast(image, tf.float32)
return image, age

def preprocess_gender_data(image_path, age, gender, dir_path=dir_path, IMAGE_SIZE = IMAGE_SIZE):
    """
    Preprocesses an image for analysis by extracting age and gender from the image path,
    loading and decoding the image, resizing it to (IMAGE_SIZE, IMAGE_SIZE), normalizing pixel values,
    and returning the preprocessed image along with age and gender labels.
    Args:
    - image_path (str): The path to the image file.
    - dir_path (str): The directory path where the image is located. Defaults to `dir_path`.
    Returns:
    - tuple: A tuple containing the preprocessed image as a TensorFlow tensor,
            the age (int), and the gender (int) extracted from the image path.
    """

    # Load the Image
    image = tf.io.read_file(dir_path + image_path)
    image = tf.io.decode_jpeg(image)
    # Resize and Normalize the Image
    image = tf.image.resize(image, (IMAGE_SIZE, IMAGE_SIZE))
    image = image / 255.
    image = tf.cast(image, tf.float32)
    return image, gender

# %% [code] {"id": "VG28fSQ3sAJM", "execution": {"iopub.status.busy": "2024-02-10T09:00:17.111414Z", "iopub.execute_input": "2024-02-10T09:00:17.111801Z", "iopub.status.idle": "2024-02-10T09:00:17.803537Z", "shell.execute_reply.started": "2024-02-10T09:00:17.111762Z", "shell.execute_reply": "2024-02-10T09:00:17.802233Z"}}
# Obtain training, testing and validation datasets
train_ds = tfds.Dataset.from_tensor_slices((train_images, train_ages, train_genders)).shuffle(2000)
train_age_ds = train_ds.map(preprocess_age_data, num_parallel_calls=BATCH_SIZE).batch(BATCH_SIZE).prefetch(tfds.AUTOTUNE)
train_gender_ds = train_ds.map(preprocess_gender_data, num_parallel_calls=BATCH_SIZE).batch(BATCH_SIZE).prefetch(tfds.AUTOTUNE)
valid_ds = tfds.Dataset.from_tensor_slices((valid_images, valid_ages, valid_genders)).shuffle(2000)
valid_age_ds = valid_ds.map(preprocess_age_data, num_parallel_calls=BATCH_SIZE).batch(BATCH_SIZE).prefetch(tfds.AUTOTUNE)
valid_gender_ds = valid_ds.map(preprocess_gender_data, num_parallel_calls=BATCH_SIZE).batch(BATCH_SIZE).prefetch(tfds.AUTOTUNE)
test_ds = tfds.Dataset.from_tensor_slices((test_images, test_ages, test_genders)).shuffle(500)
test_age_ds = test_ds.map(preprocess_age_data, num_parallel_calls=BATCH_SIZE).batch(BATCH_SIZE).prefetch(tfds.AUTOTUNE)
test_gender_ds = test_ds.map(preprocess_gender_data, num_parallel_calls=BATCH_SIZE).batch(BATCH_SIZE).prefetch(tfds.AUTOTUNE)
# %% [markdown] {"id": "NE74kskww7gT"}
# We have **successfully** processed and loaded the images**. It's time to **visualize some of them**.
# %% [code] {"id": "JLJ0O7Us8D8F", "outputId": "47264cb9-f021-434e-acda-fbdac4e47b8e", "execution": {"iopub.status.busy": "2024-02-10T09:00:17.805305Z", "iopub.execute_input": "2024-02-10T09:00:17.805701Z", "iopub.status.idle": "2024-02-10T09:00:22.329677Z", "shell.execute_reply.started": "2024-02-10T09:00:17.805668Z", "shell.execute_reply": "2024-02-10T09:00:22.327529Z"}}
plt.figure(figsize=(15, 10))
for images, ages, genders in train_ds.batch(BATCH_SIZE).take(1):
    for index in range(len(images)):
        image = tf.io.read_file(dir_path + images[index])
        image = tf.io.decode_jpeg(image)
        plt.subplot(4, 8, index + 1)
        plt.imshow(image)
        plt.title(f"Age: {ages[index]}\nGender: {gender_mapping[genders[index]]}")
        plt.axis("off")
plt.tight_layout()
plt.show()
backbones = [
    (
        "VGG16",
        VGG16(
            input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3),
            weights = "imagenet",
            include_top = False
        )
    ),
    (
        "ResNet50V2",
        ResNet50V2(
            input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3),
            weights = "imagenet",
            include_top = False
        )
    ),
    (
        "ResNet152V2",
        ResNet152V2(

```

```

input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3),
    weights = "imagenet",
    include_top = False
)
),
(
    "Xception",
    Xception(
        input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3),
        weights = "imagenet",
        include_top = False
    )
),
(
    "InceptionV3",
    InceptionV3(
        input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3),
        weights = "imagenet",
        include_top = False
    )
),
(
    "MobileNetV3Small",
    MobileNetV3Small(
        input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3),
        weights = "imagenet",
        include_top = False
    )
),
(
    "MobileNetV3Large",
    MobileNetV3Large(
        input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3),
        weights = "imagenet",
        include_top = False
    )
),
]
# %% [markdown] {"id": "netDJuY35iBd"}
# %% [code] {"id": "DdC8xZCH5TSR", "outputId": "6ccdfbb6-ab49-4730-f670-076dc66ecd5"}
BACKBONE_HISTORIES = {}
for (name, backbone) in backbones:
    print(f"Testing : {name}")
    # Freeze the Model weights
    backbone.trainable = False
    # Creating a base model
    model = keras.Sequential([
        InputLayer((IMAGE_SIZE, IMAGE_SIZE, 3), name = "InputLayer"),
        backbone,
        Dropout(0.2, name = "SlightDropout"),
        Flatten(name = "FlattenEmbeddings"),
        Dense(1, name = "Age")
    ])
    # Train the model for few iterations
    model.compile(
        loss = ["mae"],
        optimizer = "adam",
        weighted_metrics=[]
    )
    history = model.fit(
        train_age_ds,
        validation_data = valid_age_ds,
        epochs = 5,
        batch_size = BATCH_SIZE
    )
    BACKBONE_HISTORIES[name] = pd.DataFrame(history.history)
    cls()
    print("\n")
# %% [code] {"id": "SPU8-_-CDeZk", "outputId": "b7f4b4fe-c971-41c5-afc6-ed8beb0d59f8", "_kg_hide-input": true}
metric = "loss"
plt.figure(figsize=(15, 5))
for i, sub in enumerate(['Train', 'Val']):
    plt.subplot(1, 2, i+1)
    plt.title(f"{sub} {metric} Plot")
    for name, history in BACKBONE_HISTORIES.items():
        plt.plot(history[metric] if sub=="Train" else history[f"val_{metric}"], label = name)
        plt.xlabel("Epochs")
        plt.ylabel(metric.title())
        plt.legend()
        plt.grid()
plt.tight_layout()
plt.show()
# %% [code] {"id": "I3VWTQCbjUHW", "outputId": "bf309da9-889f-49ed-aec2-a2a5ab8790e4"}
BACKBONE_HISTORIES = {}

```

```

for (name, backbone) in backbones:
    print(f"Testing : {name}")
    # Freeze the Model weights
    backbone.trainable = False
    # Creating a base model
    model = keras.Sequential([
        InputLayer((IMAGE_SIZE, IMAGE_SIZE, 3), name = "InputLayer"),
        backbone,
        Dropout(0.2, name = "SlightDropout"),
        Flatten(name = "FlattenEmbeddings"),
        Dense(1, activation="sigmoid", name = "Age")
    ])
    # Train the model for few iterations
    model.compile(
        loss = ["binary_crossentropy"],
        optimizer = "adam",
        metrics = ['accuracy'],
        weighted_metrics=[]
    )
    history = model.fit(
        train_gender_ds,
        validation_data = valid_gender_ds,
        epochs = 5,
        batch_size = BATCH_SIZE
    )
    BACKBONE_HISTORIES[name] = pd.DataFrame(history.history)
    cls()
    print("\n")
# %% [code] {"id": "iwZfIVET9FhQ", "outputId": "1db0a020-d532-40d6-c7f4-14866ebc5a96", "_kg_hide-input": true}
metric = "accuracy"
plt.figure(figsize=(15, 5))
for i, sub in enumerate(['Train', 'Val']):
    plt.subplot(1, 2, i+1)
    plt.title(f"{sub} {metric} Plot")
    for name, history in BACKBONE_HISTORIES.items():
        plt.plot(history[metric] if sub=="Train" else history[f"val_{metric}"], label = name)
    plt.xlabel("Epochs")
    plt.ylabel(metric.title())
    plt.legend()
    plt.grid()
plt.tight_layout()
plt.show()
# %% [markdown] {"id": "lIXEvePL-kI8"}
# %% [code] {"id": "43ou9f-0g14N"}
# Loading backbone
vgg_16 = VGG16(input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3), include_top = False, weights = "imagenet")
# Freezing the backbone weights
vgg_16.trainable = False
# Creating a Age Network
age_net = Sequential([
    InputLayer((IMAGE_SIZE, IMAGE_SIZE, 3), name="ImageInput"),
    vgg_16,
    Dropout(0.4, name = "SlightDropout"),
    Flatten(name="FlattenEmbeddings"),
    Dense(256, activation="relu", kernel_initializer="he_normal"),
    Dense(1, name="AgeOutput")
], name="AgeNet")
# Compiling Model
age_net.compile(
    loss = "mae",
    optimizer = 'adam',
    weighted_metrics=[]
)
# %% [code] {"id": "oXSyxpZuFZWu", "outputId": "17f9df95-1a84-4f09-91a6-44e37ae9a26c"}
# Trining the Age Model
age_history = age_net.fit(
    train_age_ds,
    validation_data = valid_age_ds,
    epochs = 20,
    batch_size = BATCH_SIZE,
    callbacks = [
        EarlyStopping(
            patience = 5,
            monitor = "val_loss",
            restore_best_weights = True
        ),
        ModelCheckpoint(
            "Age-VGG16.keras",
            save_best_only=True
        )
    ]
)
# %% [code] {"id": "iXn4neZeH0T2", "outputId": "312508cf-e79e-475f-ade3-2148f1fda3a3", "_kg_hide-input": true}
# Converting history into data frame.

```

```

age_his_df = pd.DataFrame(age_history.history)
# Visualization
age_his_df.plot()
plt.title("Age Network Learning Curve")
plt.xlabel("Epochs")
plt.ylabel("Mean Abs. Error")
plt.grid()
plt.show()
# %% [markdown] {"id": "_uDjLbHKJk8"}
# %% [markdown] {"id": "vW_KVWWzIOWw"}
# **ResNet152V2 Gender Network 🐶**
# %% [code] {"id": "Pw3XFISGFtYs"}
# Initializing the backbone layer
resnet = ResNet152V2(input_shape = (IMAGE_SIZE, IMAGE_SIZE, 3), include_top = False, weights="imagenet")
# Freeze the backbone weights
resnet.trainable = False
# Creating a gender model
gender_net = Sequential([
    InputLayer((IMAGE_SIZE, IMAGE_SIZE, 3), name="ImageInput"),
    resnet,
    Dropout(0.2, name = "SlightDropout"),
    GlobalAveragePooling2D(name="GlobalAvgPooling"),
    Dense(1, activation="sigmoid", name="gender")
], name="GenderNet")
gender_net.compile(
    loss = "binary_crossentropy",
    optimizer = 'adam',
    metrics = ['accuracy'],
    weighted_metrics=[]
)
# %% [code] {"id": "oj5mv-mkKnBh", "outputId": "93d2a51f-8e1c-46fb-9fcc-35901bf7fb58"}
# Training the Gender Model
gender_history = gender_net.fit(
    train_gender_ds,
    validation_data = valid_gender_ds,
    epochs = 20,
    batch_size = BATCH_SIZE,
    callbacks = [
        EarlyStopping(
            patience = 5,
            monitor = "val_accuracy",
            restore_best_weights = True
        ),
        ModelCheckpoint(
            "Gender-ResNet152.keras",
            save_best_only=True
        )
    ]
)
# %% [markdown] {"id": "qhMnLSqc4WD"}
gender_his_df = pd.DataFrame(gender_history.history)
# Visualization
plt.figure(figsize=(10, 5))
plt.subplot(1,2,1)
plt.suptitle("Gender Network Learning Curve")
plt.plot(gender_his_df['loss'], label="Loss")
plt.plot(gender_his_df['val_loss'], label="Val. Loss")
plt.xlabel("Epochs")
plt.ylabel("$Loss$")
plt.grid()
plt.legend()
plt.subplot(1,2,2)
plt.plot(gender_his_df['accuracy'], label="Accuracy")
plt.plot(gender_his_df['val_accuracy'], label="Val. Accuracy")
plt.xlabel("Epochs")
plt.ylabel("$Accuracy$")
plt.grid()
plt.legend()
plt.show()
# %% [markdown] {"id": "Y_igA6_XYUX"}
# %% [markdown] {"id": "5GYgnOLtp_k"}
# **Model Evaluation**
# %% [markdown] {"id": "Wo3djfnOOS1V"}
# %% [code] {"id": "N6xrOroCYiH0", "execution": {"iopub.status.busy": "2024-02-10T09:00:40.602385Z", "iopub.execute_input": "2024-02-10T09:00:40.602782Z", "iopub.status.idle": "2024-02-10T09:02:18.966225Z", "shell.execute_reply.started": "2024-02-10T09:00:40.602752Z", "shell.execute_reply": "2024-02-10T09:02:18.964895Z"}}
# Loading saved models
age_net = tf.keras.models.load_model("/kaggle/input/age-and-gender-prediction-vgg16-resnet152v2/Weights/Age-VGG16.keras", compile=True)
gender_net = tf.keras.models.load_model("/kaggle/input/age-and-gender-prediction-vgg16-resnet152v2/Weights/Gender-ResNet152.keras", compile=True)
# %% [code] {"id": "n7oYqFQe8YN", "outputId": "832dae58-b610-4b3f-a529-e81bbe2038ab", "execution": {"iopub.status.busy": "2024-02-10T09:02:20.84472Z", "iopub.execute_input": "2024-02-10T09:02:20.845159Z", "iopub.status.idle": "2024-02-10T09:22:50.386646Z", "shell.execute_reply.started": "2024-02-10T09:02:20.845106Z", "shell.execute_reply": "2024-02-10T09:22:50.385687Z"}}
# Evaluating test performance

```

```

age_net.evaluate(test_age_ds)
gender_net.evaluate(test_gender_ds)
# %% [markdown] {"id": "rj2nSOzrGhQ4"}
# %% [markdown] {"id": "hQKqiljPOzOT"}
# # **Model Predictions**
# %% [markdown] {"id": "m_m6vgX3O1On"}
# %% [code] {"id": "EjOjdJ95Ggdz", "execution": {"iopub.status.busy": "2024-02-10T09:26:15.373315Z", "iopub.execute_input": "2024-02-10T09:26:15.373697Z", "iopub.status.idle": "2024-02-10T09:26:44.370082Z", "shell.execute_reply.started": "2024-02-10T09:26:15.373668Z", "shell.execute_reply": "2024-02-10T09:26:44.368989Z"}}
plt.figure(figsize=(15, 10))
for images, ages, genders in test_ds.batch(BATCH_SIZE).take(1):
    for index in range(len(images)):
        # Image Processing
        image = tf.io.read_file(dir_path + images[index])
        image = tf.io.decode_jpeg(image)
        image = tf.cast(image, tf.float32)
        image = image/255.
        image = tf.image.resize(image, [224,224])
        image = tf.reshape(image, [-1,224,224,3])

        # Predictions
        pred_age = tf.round(tf.squeeze(age_net.predict(image, verbose=0)), 2)
        pred_gender = tf.round(tf.squeeze(gender_net.predict(image, verbose=0)))
        # Visualization
        plt.subplot(4, 8, index + 1)
        plt.imshow(image[0])
        plt.title(f"Age: {int(pred_age)}\nGender: {gender_mapping[int(pred_gender)]}")
        plt.axis("off")
plt.tight_layout()
plt.show()
# %% [markdown]

```

ДОДАТОК В
АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

МАТЕРІАЛИ

У ВСЕУКРАЇНСЬКОЇ СТУДЕНТСЬКОЇ НАУКОВОЇ

КОНФЕРЕНЦІЇ

17 ТРАВНЯ 2024 РІК • М. КИЇВ, УКРАЇНА

НАУКОВИЙ ПРОСТІР: АНАЛІЗ,
СУЧАСНИЙ СТАН, ТРЕНДИ ТА
ПЕРСПЕКТИВИ

ISBN 978-617-8312-44-2
DOI 10.36074/liga-uki-17.05.2024



**УДК 082:001
Н 34**

Голова оргкомітету: Коренюк І.О.

Верстка: Зрада С.І.

Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 36 від 16.05.2024 року.



Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та інформаційному бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення №29 від 05.01.2024).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії CC BY-SA 4.0 International.

Науковий простір: аналіз, сучасний стан, тренди та перспективи:
Н 34 матеріали V Всеукраїнської студентської наукової конференції, м. Київ, 17 травня, 2024 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2024. — 586 с.

ISBN 978-617-8312-44-2

DOI 10.62732/liga-ukr-17.05.2024

Викладено матеріали учасників V Всеукраїнської мультидисциплінарної студентської наукової конференції «Науковий простір: аналіз, сучасний стан, тренди та перспективи», яка відбулася 17 травня 2024 року у місті Київ, Україна.

УДК 082:001

© Колектив учасників конференції, 2024

© ГО «Молодіжна наукова ліга», 2024

© ТОВ «УКРЛОГОС Груп», 2024

ISBN 978-617-8312-44-2

АРХІТЕКТУРА МОДУЛЯ ПРОГНОЗУВАННЯ СЕРЦЕВИХ ЗАХВОРЮВАНЬ Закалюк П., Науковий керівник: Констанкевич Ю.В.	374
АРХІТЕКТУРА МОДУЛЯ РЕКОМЕНДАЦІЙ МУЗИКИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ Черній І., Науковий керівник: Констанкевич Ю.В.	376
АРХІТЕКТУРА МОДУЛЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ МАШИННОГО НАВЧАННЯ Філюк В., Науковий керівник: Констанкевич Ю.В.	378
АРХІТЕКТУРА МОДУЛЯ РОЗПІЗНАВАННЯ ВІКУ ТА СТАТІ ЛЮДЕЙ Мартиник В., Науковий керівник: Сапожник Г.В.	380
АРХІТЕКТУРА МОДУЛЯ РОЗПІЗНАВАННЯ ЯКОСТІ ФРУКТІВ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ Чіп С., Науковий керівник: Сапожник Г.В.	382
ВИБІР ТА ОБГРУНТУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ УКРИТТЯ Заровський С.В., Науковий керівник: Хімичева Г.І.	384
ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ОПТИМІЗАЦІЇ ПРОВЕДЕННЯ СПІВБЕСІД Бурлаков О.О., Науковий керівник: Левус Є.В.	387
ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ОСВІТНІХ КУРСІВ Горбовий В.Р., Науковий керівник: Левус Є.В.	389
ІНФОРМАЦІЙНА СИСТЕМА НАДАННЯ ТЕЛЕМЕДИЧНИХ КОНСУЛЬТАЦІЙ З ПІДТРИМКОЮ СТАНДАРТУ HL7 Смерек Л.Л., Науковий керівник: Кісь Я.П.	391
МЕТОДИ ТА ЗАСОБИ МОДУЛЯЦІЇ РАДІОТЕХНІЧНИХ СИГНАЛІВ АВТОГЕНЕРАТОРІВ Овчарук А.О., Науковий керівник: Осадчук В.С.	394
ПРОГНОЗУВАННЯ ЦІН НА АКЦІЇ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ Воробйов А.Р.	396
СИСТЕМА ДЛЯ ЗАБЕЗПЕЧЕННЯ ПЕРЕНАВЧАННЯ НЕЙРОННИХ СТРУКТУР МОЗКУ ЩОДО ФУНКЦІОНУВАННЯ ШЛЯХІВ ПЕРЕДАЧІ ЗОБРАЖЕННЯ ВІД ЗОРОВОГО НЕРВУ Гема О.Г., Науковий керівник: Кузьомін О.Я.	398

СЕКЦІЯ 17.

ФІЗИКО-МАТЕМАТИЧНІ НАУКИ

ОСОБЛИВОСТІ ДИСТАНЦІЙНОЇ ФОРМИ НАВЧАННЯ ПРЕДМЕТУ “ВИЩА МАТЕМАТИКА” НА СПЕЦІАЛЬНОСТІ “АРХІТЕКТУРА” Бондарець В.Ю., Науковий керівник: Турчанінова Л.І.	400
--	-----

Мартиник Владислав, здобувач вищої освіти факультету комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Науковий керівник: Сапожник Г.В., канд. іст. наук, канд. техн. наук, доцент,
доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

АРХІТЕКТУРА МОДУЛЯ РОЗПІЗНАВАННЯ ВІКУ ТА СТАТІ ЛЮДЕЙ

Архітектура модуля розпізнавання віку та статі людей базується на використанні нейронних мереж, що дозволяє ефективно обробляти великі набори даних зображень облич. Цей модуль складається з декількох ключових компонентів, кожен з яких відіграє важливу роль у процесі підготовки, тренування та оцінки моделі.

Архітектура модуля розпізнавання віку та статі людей структурована таким чином (Рисунок 1):

1. Налаштування середовища:

- **Імпорт:** Цей клас обробляє імпорт необхідних бібліотек та фреймворків, таких як TensorFlow, Keras, NumPy, і інших, потрібних для розробки моделі.
- **Константи:** Визначає ключові константи, такі як розмір зображення, розмір пакету та інші параметри, які використовуються у всьому модулі для обробки та тренування.

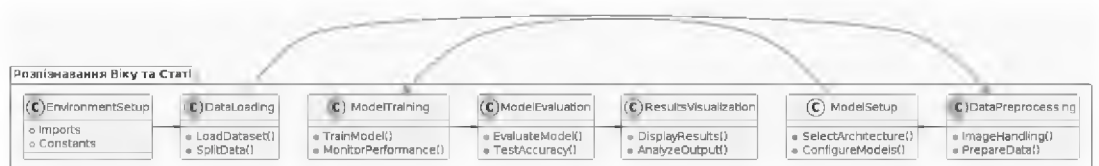


Рис. 1. Архітектура [1] модуля розпізнавання віку та статі людей

2. Завантаження даних:

- **Завантаження набору даних (LoadDataset()):** Відповідає за завантаження набору даних UTKFace, який включає зображення з анотаціями віку, статі та етнічності.
- **Розділення даних (SplitData()):** Ця функція ділить набір даних на тренувальні, тестові та валідаційні набори, щоб забезпечити належне розподілення для тренування моделі та оцінки.

3. Попередня обробка даних:

- **Обробка зображень (ImageHandling()):** Реалізує функції обробки зображень, такі як зміна розміру, нормалізація та можливо аугментація, для підготовки зображень до ефективного тренування моделі.
- **Підготовка даних (PrepareData()):** Підготує фінальний набір даних, організовуючи оброблені зображення разом з їх мітками у форматі, придатному для тренування.

4. Налаштування моделі:

- **Вибір архітектури (SelectArchitecture()):** Вибирає архітектуру нейронної мережі

(наприклад, VGG16, ResNet чи Xception), яка буде використовуватися як основа для завдань розпізнавання.

- Конфігурація моделей (ConfigureModels()): Налаштовує вибрану модель для конкретних завдань, налаштовуючи шари, функції активації та компілюючи модель з відповідними функціями втрат та оптимізаторами.

5. Тренування моделі:

- Тренування моделі (TrainModel()): Керує процесом тренування моделі, використовуючи підготовлені набори даних, застосовуючи колбеки, такі як рання зупинка та контроль точок збереження моделі, для оптимізації тренування.

- Моніторинг продуктивності (MonitorPerformance()): Слідкує та реєструє продуктивність моделі під час тренування для відстеження покращень і внесення необхідних коригувань.

6. Оцінка моделі:

- Оцінка моделі (EvaluateModel()): Оцінює продуктивність навченої моделі на окремому тестовому наборі, щоб виміряти її ефективність у реальних сценаріях.

- Тестування точності (TestAccuracy()): Конкретно вимірює точність моделі з точки зору передбачення віку та класифікації статі.

7. Візуалізація результатів:

- Відображення результатів (DisplayResults()): Візуалізує результати передбачень моделі за допомогою графіків та діаграм для ясного уявлення передбачень моделі у порівнянні з реальними мітками.

- Аналіз результатів (AnalyzeOutput()): Забезпечує глибший аналіз результатів, виявляючи тенденції, упередження або області, де модель може показувати недостатню продуктивність.

Цей структурований підхід дозволяє систематично розробляти і оцінювати надійну модель, здатну точно розпізнавати вік і стать з обличчя. Кожен компонент в архітектурі відіграє важливу роль у забезпеченні не тільки точності, але й загальної застосовності моделі до нових, невідомих даних.

Список використаних джерел:

1. Архітектура програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій / М. Дивак та ін. Measuring and computing devices in technological processes. 2024. № 1. С. 125–139. URL: <https://doi.org/10.31891/2219-9365-2024-77-15> (дата звернення: 13.05.2024).