МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних інформаційних систем

_____Юрій КОНДРАТЕНКО

«____»_____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ

Спеціальність 122 Комп'ютерні науки

Освітня програма «Інтелектуальні інформаційні системи»

Здобувач Богдан ВАЛЮШОК

«___»_____2024р.

Керівник д-р фіз.-мат. наук, професор Едуард ЛИСЕНКОВ

«___»_____ 2024р.

м. Миколаїв – 2024

Чорноморський національний університет імені Петра Могили
(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних інформаційних систем
_____Юрій КОНДРАТЕНКО
«____» _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу здобувача

Валюшка Богдана Ігоровича
((прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Інтелектуальна система для розпізнавання жестової мови з використанням нейронної мережі».

Керівник роботи: Лисенков Едуард Анатолійович, професор кафедри фізики та математики, д-р фіз-мат наук, професор.

Затверджена наказом ЧНУ ім. Петра Могили від «03» червня 2024 р. № 140/1.

2. Строк представлення кваліфікаційної роботи «16» грудня 2024 р.

3. Очікуваний результат роботи та початкові дані:

Очікуваний результат – створення автоматизованої системи розпізнавання жестової мови, яка здатна аналізувати як дактильну абетку, так і калькульовану жестову мову. Для цього будуть використаний датасет з анотованими зображеннями жестів, які охоплюють широкий спектр рухів і позицій пальців. Система базуватиметься на сучасних технологіях комп'ютерного зору (зокрема, YOLOv8), що забезпечить високу точність та швидкість розпізнавання. Початковими даними є структурований датасет з відповідною анотацією жестів, експертні оцінки важливості різних жестів для практичного використання, а також критерії оцінки точності й ефективності моделі.

4. Перелік питань, що підлягають розробці: Аналіз сучасного стану задачі розпізнавання жестової мови, включно з існуючими підходами та системами. Огляд методів машинного навчання та глибокого навчання, що застосовуються для задач комп'ютерного зору. Розробка та створення структурованого датасету для навчання та тестування моделі розпізнавання жестів. Вибір та налаштування оптимальної архітектури нейронної мережі для задачі розпізнавання жестової мови. Порівняльний аналіз ефективності різних методів для розпізнавання жестів на основі визначених критеріїв, таких як точність, швидкість та ресурсоємність.

5. Перелік графічних матеріалів: презентація.

Керівник роботи _____ Едуард ЛИСЕНКОВ

(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувач _____ Богдан ВАЛЮШОК

(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата видачі завдання «07» червня 2024 р.

КАЛЕНДАРНИЙ ПЛАН
кваліфікаційної роботи

Тема: Інтелектуальна система для розпізнавання жестової мови з використанням нейронної мережі

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	03.06.2024	07.06.2024	Виконано
2	Аналіз предметної області та постановка задачі	10.06.2024	20.06.2024	Виконано
3	Огляд літературних джерел за темою кваліфікаційної роботи, зокрема аналіз публікацій та подібних систем, щодо розпізнавання жестової мови	21.06.2024	01.07.2024	Виконано
4	Огляд існуючих архітектур штучних нейронних мереж для вирішення поставленої задачі	01.09.2024	25.10.2024	Виконано
5	Реалізація обраних технологій з аналізом отриманих результатів	26.10.2024	21.11.2024	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	22.11.2024	22.11.2024	Виконано
7	Корегування роботи за результатами попереднього захисту	23.11.2024	05.12.2024	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	06.12.2024	06.12.2024	Виконано
9	Доробка та остаточне оформлення КР	07.12.2024	14.12.2024	Виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.12.2024	16.12.2024	Виконано

Керівник роботи _____ Едуард ЛИСЕНКОВ
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувач _____ Богдан ВАЛЮШОК
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата складання календарного плану «24» червня 2024

АНОТАЦІЯ

до кваліфікаційної роботи
здобувача групи 601м ЧНУ ім. Петра Могили

Валюшка Богдана Ігоровича

на тему: **“ ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ”**

Актуальність дослідження полягає у необхідності вдосконалення методів розпізнавання жестів для підтримки комунікації людей із порушеннями слуху за допомогою сучасних технологій. Використання методологій навчання нейронних мереж дозволяє підвищити точність та швидкість розпізнавання жестів, що сприятиме створенню інклюзивних рішень у сфері освіти, обслуговування та інтеграції людей із вадами слуху. Розробка ефективних алгоритмів і програмного забезпечення зможе значно зменшити бар'єри у спілкуванні, підвищуючи якість соціальної взаємодії та забезпечуючи рівний доступ до інформаційних і соціальних ресурсів.

Об’єктом дослідження є українська жестова мова та її різновиди.

Предметом дослідження є методи навчання та розпізнавання мови жестів.

Метою роботи є покращення взаємодії людей з вадами слуху за допомогою розробки інтелектуальної системи з розпізнаванням образів за допомогою нейронної мережі

В результаті виконання роботи було розроблено систему для автоматизованого розпізнавання жестів із використанням моделі YOLOv8-pose. Було досліджено ефективність моделі, проаналізовано вплив ключових параметрів навчання та аугментації даних, визначено переваги та недоліки підходу для практичних застосувань. Результати роботи знайшли застосування у створенні програмного забезпечення, що сприяє інклюзивній комунікації.

Робота складається з чотирьох розділів: огляд предметної області, аналіз методів розпізнавання жестів, розробка й навчання моделі, а також аналіз результатів тестування та оцінка точності моделі.

Загальний обсяг роботи – 91 сторінок. Кваліфікаційна робота містить 1 додаток, 41 рисунок, 1 таблицю і 45 джерел посилання.

Ключові слова: Розпізнавання мови жестів, детекція об'єктів, оптимізація, дактильна абетка, глибоке навчання, YOLOv8, YOLOv8n-POSE.

ABSTRACT

to the qualification work by the student of the group 601m of Petro Mohyla Black Sea National University

Valiushok Bohdan

“ INTELLIGENT SYSTEM FOR SIGN LANGUAGE RECOGNITION USING A NEURAL NETWORK ”

The relevance of the research lies in the need to improve gesture recognition methods to support communication of people with hearing impairments using modern technologies. The use of neural network training methodologies allows to increase the accuracy and speed of gesture recognition, which will contribute to the creation of inclusive solutions in the field of education, service and integration of people with hearing impairments. The development of effective algorithms and software can significantly reduce barriers to communication, improving the quality of social interaction and ensuring equal access to information and social resources.

The object of the research is Ukrainian sign language and its varieties.

The subject of the research is methods of teaching and recognizing sign language.

The aim of the work is to improve the interaction of people with hearing impairments by developing an intelligent system with pattern recognition using a neural network.

As a result of the work, a system for automated gesture recognition using the YOLOv8-pose model was developed. The effectiveness of the model was investigated, the influence of key parameters of training and data augmentation was analyzed, and the advantages and disadvantages of the approach for practical applications were identified. The results of the work were used in the creation of software that promotes inclusive communication.

The work consists of four sections: overview of the subject area, analysis of gesture recognition methods, development and training of the model, as well as analysis of testing results and assessment of model accuracy.

The total volume of the work is 91 pages. The qualification work contains 1 application, 41 figures, 1 table and 45 references.

Keywords: Sign language recognition, object detection, dactylic alphabet, optimization, deep learning, YOLOv8, YOLOv8n-POSE.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

УМЖ – українська мова жестів

ASL – американська жестова мова

CIoU – Loss Complete IoU Loss

C2f – модуль YOLOv8

FPN – Feature Pyramid Networks

PAN – Path Aggregation Networks

SPPF – Spatial Pyramid Pooling Fast

CSP – Cross-Stage Partial Networks

GFLOPs – Giga Floating Point Operations per Second

IoU – Intersection over Union

mAP – mean Average Precision

FPS – Frames Per Second

ВСТУП

Розпізнавання мови жестів є важливим напрямом у сучасних дослідженнях комп'ютерного зору та штучного інтелекту, оскільки воно має значний вплив на соціальну інклюзію та комунікацію. Мова жестів є основним засобом спілкування для багатьох людей з порушенням слуху або мовлення, забезпечуючи їм можливість взаємодіяти з навколишнім світом. Однак, у традиційній мовній комунікації існує бар'єр між носіями мови жестів і людьми, які не володіють нею, що може призводити до соціальної ізоляції або обмеження можливостей у повсякденному житті, роботі та навчанні.

Технології розпізнавання мови жестів дозволяють автоматизувати переклад жестів у текст або мовлення, роблячи спілкування між глухими або слабочуючими людьми та звичайними простішим і доступнішим. Це знижує залежність від професійних перекладачів та сприяє самостійності людей з порушеннями слуху. Зокрема, в контексті інтеграції таких технологій у смартфони, комп'ютери та інші пристрої, відкриваються можливості для широкого використання цих рішень у різних галузях: від освіти та медицини до сфери обслуговування та розваг.

Окрім того, розробка технологій розпізнавання жестів має велике значення для створення більш доступних інтерфейсів користувача, зокрема для тих, хто має обмеження в мобільності або сенсорних можливостях. Наприклад, жести можуть слугувати як альтернативний спосіб управління комп'ютерами або пристроями, що особливо важливо в умовах, коли традиційні інтерфейси недоступні.

Таким чином, розпізнавання мови жестів є ключовим кроком у напрямку побудови більш інклюзивного та рівноправного суспільства, де технології допомагають подолати бар'єри в спілкуванні та сприяють розширенню можливостей для всіх людей.

1 АНАЛІЗ СИСТЕМ РОЗПІЗНАВАННЯ ЖЕСТОВИХ МОВ ТА ПРОБЛЕМАТИКА УКРАЇНСЬКОЇ МОВИ ЖЕСТІВ

1.1 Проблематика української мови жестів

Українська мова жестів (рис. 1.1) є повноцінною мовною системою, яка використовується здебільшого людьми з порушеннями слуху для щоденного спілкування. УМЖ має власну граматику, синтаксис та унікальні правила, що робить її не просто набором жестів для перекладу окремих слів, а самостійною, складною мовою. Розпізнавання нашої мови жестів є важливою проблемою, яка стоїть на перетині технологій, інклюзії та рівності, і її вирішення має суттєвий вплив на соціальне життя людей із порушенням слуху [1].

Основною причиною необхідності розвитку технологій для автоматичного розпізнавання української мови жестів є прагнення до соціальної рівності та інклюзії. Для людей із вадами слуху мова жестів є єдиним ефективним засобом спілкування, але її знання серед широкого загалу є вкрай обмеженим. Це призводить до соціальної ізоляції глухих або слабочуючих людей, особливо в повсякденному житті, де немає можливості скористатися послугами перекладача або де середовище не пристосоване для розуміння жестів. Відсутність розуміння мови жестів серед населення значно обмежує доступ людей із порушеннями слуху до освіти, медицини, робочих місць, державних послуг та громадських подій. Тому автоматизація розпізнавання УМЖ дозволила б людям із порушенням слуху більш вільно інтегруватися в соціум.

Однією з головних проблем є технологічна складність, пов'язана з розпізнаванням жестової мови загалом і української зокрема. На відміну від стандартних мов, де слова виражаються через голосовий апарат, мова жестів використовує координацію різних частин тіла: рук, обличчя, корпусу, а також зміну положення і руху рук у просторі. Це створює багаторівневу систему, яку складно перекласти на комп'ютерний алгоритм без втрати точності чи семантики. Крім того,

українська мова жестів має власну унікальну структуру, яка відрізняється від інших жестових мов, таких як американська або британська, тому адаптація іноземних технологій для української мови жестів є неповноцінним рішенням [2].

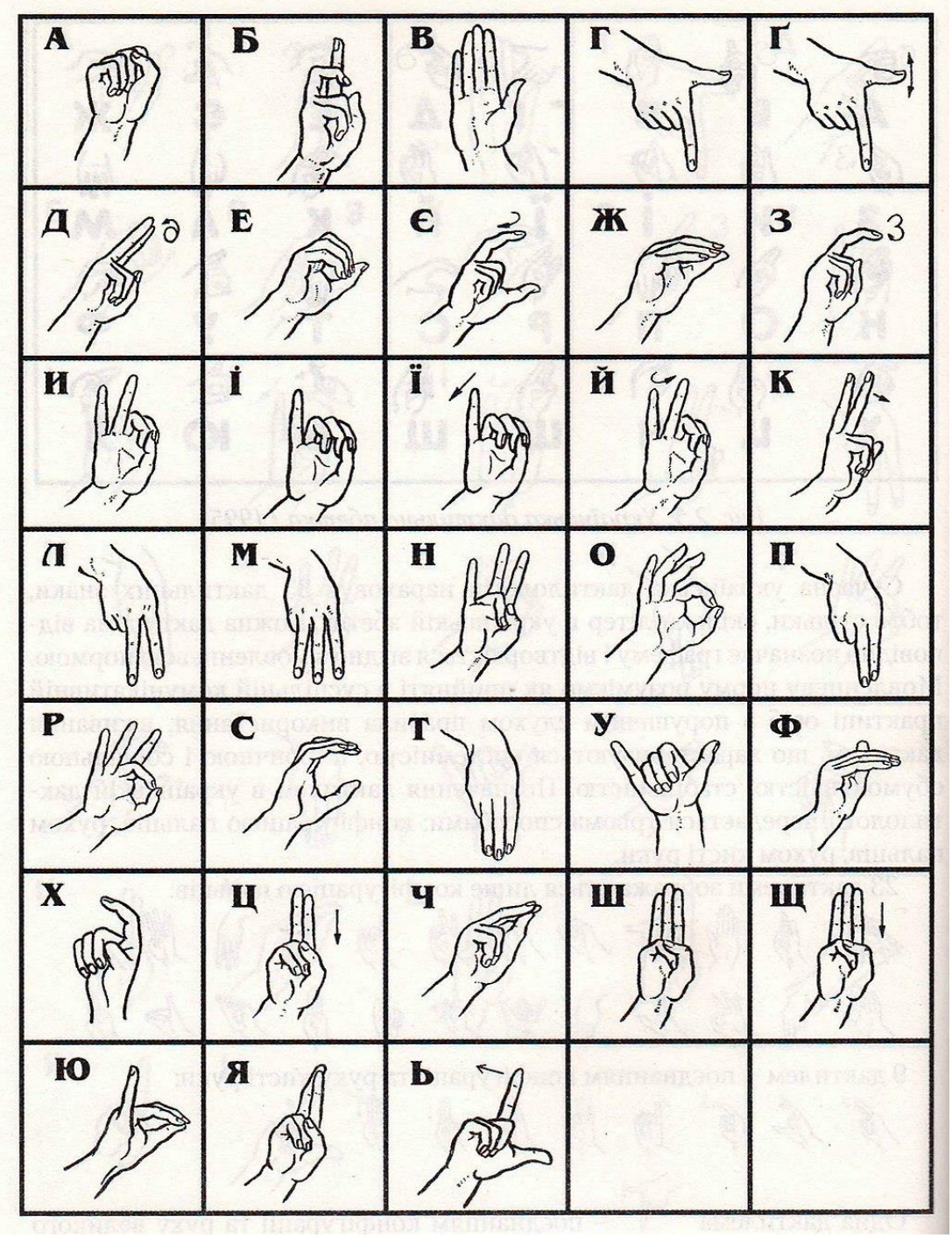

Рисунок 1.1 – Дактильна абетка мови жестів

Також викликом є відсутність датасетів для навчання штучного інтелекту розпізнавати УМЖ. Машинне навчання, на якому ґрунтуються сучасні системи розпізнавання жестів, потребує великих обсягів даних для навчання моделей. Оскільки в Україні ще не було створено повноцінних датасетів з відеозаписами жестів, це гальмує розвиток технологій для автоматизації перекладу УМЖ. Кожен жест може варіюватися залежно від індивідуальних фізичних особливостей мовця або контексту, що додатково ускладнює розробку універсальної моделі для розпізнавання [3].

Окрім технічних аспектів, проблемою є й відсутність достатнього фінансування та уваги до розробки таких рішень. Технології розпізнавання мови жестів розвиваються переважно у країнах з потужними ресурсами та підтримкою з боку держави або великих корпорацій. В Україні такі проекти здебільшого знаходяться на волонтерському або академічному рівні і не отримують достатньо фінансових ресурсів для повноцінного розвитку. Це сповільнює прогрес у створенні комерційних рішень для автоматизації перекладу жестової мови.

Ще однією важливою проблемою є необхідність стандартизації жестів. Хоча українська мова жестів має певні встановлені норми, у практиці існують різні варіанти одних і тих самих жестів у різних регіонах країни. Це може стати проблемою для автоматизованих систем, які мають розпізнавати різноманітні варіанти жестів і одночасно забезпечувати високу точність перекладу.

Розв'язання проблеми розпізнавання української мови жестів є важливим не лише з технологічної, але й з гуманітарної точки зору. Це сприяло б не тільки інтеграції глухих людей у суспільство, але й допомогло б забезпечити рівний доступ до освіти, медичних послуг, працевлаштування та громадських активностей. Діти з вадами слуху отримали б більше можливостей для інтеграції в навчальний процес, а дорослі – для професійного розвитку і самореалізації [4].

Окрім цього, розвиток технологій для розпізнавання УМЖ також підвищить обізнаність серед чуючих людей про важливість жестової мови, що може призвести

до її ширшого вивчення і розповсюдження. У довгостроковій перспективі це сприятиме побудові інклюзивного суспільства, в якому всі члени, незалежно від їхніх фізичних можливостей, матимуть рівний доступ до інформації та комунікації.

Таким чином, розробка і впровадження технологій для автоматичного розпізнавання української мови жестів є стратегічно важливою для соціальної інклюзії та рівності, створення безбар'єрного середовища та забезпечення доступу до важливих ресурсів і можливостей для людей з порушеннями слуху.

1.2 **Дактильна та калькуюча жестова мова**

В українській мові жестів існують дві системи комунікації: дактильна абетка та калькуюча жестова мова. Хоча обидві системи використовуються для передачі інформації за допомогою жестів, вони мають принципові відмінності в способах вираження та використання.

Дактильна абетка – це система жестів, за допомогою якої кожна буква українського алфавіту позначається відповідним положенням пальців. Ця система дозволяє буквально передавати написані слова, "написуючи" їх у повітрі. Кожен жест у дактильній абетці відповідає певній літері, що дає змогу передавати власні імена, специфічні терміни або інші слова, для яких не існує спеціальних жестів.

Дактильна абетка широко використовується в таких випадках:

- для позначення власних імен та прізвищ;
- для нових термінів або слів, які ще не мають жестового еквівалента;
- у ситуаціях, коли необхідна точність у передачі певних звуків або слів.

Однак використання дактилю є повільнішим, оскільки кожну літеру необхідно відтворювати окремим жестом. Це зручний інструмент для уточнень, але не завжди ефективний для швидкого спілкування.

Калькуюча жестова мова – це система жестів, яка безпосередньо калькує, тобто передає, структуру української розмовної мови. Вона намагається якомога точніше відтворити синтаксис, граматичні структури та порядок слів, характерні

для української мови. У цьому випадку жести використовуються для передачі кожного слова або поняття так, як вони би використовувалися в усному мовленні.

Калькуюча жестова мова є корисною для людей, які знають обидві системи - жестову та українську розмовну мови. Вона часто використовується у навчальних процесах або під час спілкування між глухими і чуючими людьми, які не є носіями жестової мови, оскільки калькуюча мова є ближчою до звичної структури усного мовлення.

Проте ця система має кілька важливих обмежень:

– мова жестів має власну граматику, і калькуюча жестова мова не завжди може адекватно відтворити всі мовні нюанси української мови;

– структура української розмовної мови, коли передається через жести, може бути штучною для носіїв жестової мови, оскільки вона не враховує специфіку природного розвитку жестової мови.

Основна різниця між дактильною абеткою та калькуючою жестовою мовою полягає в способі передачі інформації. Дактильна абетка використовує жести для позначення окремих букв, що дозволяє передавати кожне слово з максимальною точністю, але повільніше. Натомість калькуюча жестова мова намагається відтворювати структуру та порядок слів української мови, але не завжди може передати повний зміст і граматичні конструкції, характерні для жестової мови.

У щоденному використанні, носії мови жестів частіше комбінують ці дві системи. Наприклад, дактильна абетка може використовуватися для передачі власних імен, термінів, тоді як калькуюча жестова мова забезпечує більш природне спілкування в контексті.

Попри значний прогрес у сфері розпізнавання жестової мови, повноцінної системи, здатної точно розпізнавати всі аспекти жестових мов, зокрема УМЖ, поки що не існує. Поточні технології зосереджені переважно на дактильній абетці, яка є відносно простішою для реалізації з технічної точки зору, але не забезпечує

повного охоплення жестової комунікації, особливо коли йдеться про складніші системи, такі як калькуюча жестова мова.

Більшість сучасних програм і досліджень у сфері автоматичного розпізнавання жестової мови зосереджені на дактильній абетці. Дактильна абетка простіша для реалізації, оскільки кожен жест відповідає конкретній літері алфавіту. Такі системи можуть навчитися розпізнавати рухи пальців і передавати їх у вигляді тексту або аудіо, що робить їх корисними для передачі окремих слів, власних імен або технічних термінів, які не мають жестового еквівалента.

Наприклад, комп'ютерна модель може навчитися розпізнавати, коли людина формує рукою літеру "А" або "Б", і зіставляти це з відповідним символом у тексті. Завдяки цій методиці можна досить точно передавати літери й слова українського алфавіту. Проте ця методологія обмежена, оскільки вона працює лише на рівні букв, що робить комунікацію повільною і неефективною для більш складних розмов [5].

Калькуюча жестова мова, на відміну від дактильної абетки, є набагато складнішою для автоматичного розпізнавання. Вона передає не тільки окремі літери, але й цілі слова та фрази, які мають власний семантичний і граматичний зміст. Калькуюча жестова мова намагається зберігати порядок слів і синтаксис української мови, тому для її успішного розпізнавання система повинна враховувати не лише рухи рук, але й контекст, граматичні конструкції та правила поєднання слів.

Сучасні системи штучного інтелекту і комп'ютерного зору не здатні ефективно враховувати ці аспекти. Однією з основних причин є те, що калькуюча жестова мова включає не тільки жести рук, але й рухи тіла, вирази обличчя, зміни в темпі та інтонації жестів. Це вимагає від системи не просто відстежувати позиції рук, а також розуміти складні взаємозв'язки між різними елементами жестової мови. Окрім того, калькуюча система значно ближча до природного спілкування глухих людей, і тому вона включає більший обсяг інформації, ніж просто дактиль.

Технічна складність полягає в тому, що для успішної реалізації повноцінного розпізнавання жестової мови необхідно поєднувати кілька технологій одночасно:

– комп'ютерний зір, здатний точно відслідковувати не тільки положення рук і пальців, але й рухи обличчя, плечей і тіла;

– контекстне розпізнавання, що дозволяє враховувати послідовність жестів і їх значення в контексті фрази або речення, подібно до того, як це відбувається в розпізнаванні мови;

– нейронні мережі, здатні навчитися обробляти різні варіації одних і тих самих жестів, оскільки різні люди можуть виконувати їх по-різному залежно від індивідуальних особливостей [6].

Окрім цього, відсутність датасетів української калькуючої жестової мови значно ускладнює процес навчання таких моделей. Для точного розпізнавання кожного жесту необхідно мати велику кількість відеозаписів з реальними прикладами спілкування. Більшість поточних моделей розпізнавання жестів зосереджені на американській або британській мовах жестів, що не враховує культурних та мовних особливостей УМЖ.

Через те, що поточні технології здебільшого зосереджені на дактильній абетці, вони не забезпечують повноцінного спілкування глухих людей із чуючими. Використання дактилю є ефективним лише в обмежених контекстах, таких як передача імен або технічних термінів. Однак у реальних життєвих ситуаціях спілкування жестами відбувається значно швидше та більш інтерактивно, ніж це можливо при використанні дактилю.

Якщо системи розпізнавання жестової мови не здатні враховувати калькуючу жестову мову, це призводить до значних бар'єрів у спілкуванні. Глухі люди можуть бути змушені адаптуватися до технологічних обмежень, що ускладнює повноцінне вираження їхніх думок. Це також створює додаткові труднощі для чуючих людей, які бажають вивчати жестову мову або ефективно спілкуватися з людьми з порушеннями слуху.

Незважаючи на існуючі проблеми, розробка повноцінних систем для розпізнавання калькуючої жестової мови залишається важливою задачею. Подальші дослідження та розвиток технологій, таких як штучний інтелект, комп'ютерний зір і обробка природної мови, можуть допомогти у створенні більш досконалих рішень, здатних розпізнавати не тільки дактиль, але й калькуючу систему.

У майбутньому, інтеграція повноцінного розпізнавання обох систем жестової мови може значно підвищити рівень соціальної інклюзії людей з порушеннями слуху, забезпечуючи рівні можливості для спілкування, навчання та участі у суспільному житті. Однак для досягнення цього необхідно подолати значні технічні та ресурсні перешкоди, включно зі створенням датасетів та адаптацією алгоритмів для унікальних аспектів української мови жестів.

1.3 Аналіз систем для розпізнавання мови жестів

На сьогодні повноцінних комерційних чи широко доступних програм для розпізнавання української мови жестів практично немає. Ця сфера залишається у стадії дослідження й розробки, переважно на академічному рівні. Водночас, існує багато проєктів у світі, які спрямовані на розпізнавання жестової мови, зокрема англійської жестової мови (ASL), яка вже отримала певні комерційні рішення та технологічні досягнення. Ці проєкти можуть слугувати орієнтиром для майбутньої реалізації розпізнавання УМЖ.

Приклади реалізацій для англійської мови жестів.

1. SignAll є одним із найвідоміших проєктів, що працює над автоматичним розпізнаванням жестової мови. Ця платформа використовує камери для відслідковування рухів рук, а також технології штучного інтелекту для розпізнавання англійської жестової мови (ASL). Особливістю SignAll є те, що система здатна розпізнавати цілі речення жестової мови в реальному часі і перекладати їх на текстову чи звукову форму.

Система використовує мультикамерний підхід для відслідковування всіх рухів людини, зокрема виразів обличчя та рухів тіла, що є критичними елементами для правильного тлумачення жестів. Така багатошарова обробка дозволяє забезпечити більш точний переклад жестової мови на англійську.

2. Google’s Project Euphonia. Хоча Google більше зосереджений на покращенні розпізнавання голосових команд для людей з порушеннями мовлення, Project Euphonia також включає дослідження щодо покращення розуміння жестової мови. Використовуючи глибоке навчання та великі обсяги даних, проект працює над тим, щоб створювати моделі, які можуть інтерпретувати як голосові команди, так і жести. Команда Google використовує величезні датасети з реальними прикладами жестової мови для тренування моделей штучного інтелекту. Хоча проект наразі не спеціалізується на розпізнаванні жестів, він показує, як великі компанії можуть використовувати свої ресурси для розвитку таких технологій.

3. HandTalk це додаток для перекладу бразильської жестової мови (Libras), але він може стати прикладом для розробки схожих рішень для інших мов. HandTalk використовує аватар, який показує жести, перекладаючи текст чи звук на жестову мову. Додаток дозволяє спілкуватися людям із порушеннями слуху з чуючими, перетворюючи усну мову на жести, і навпаки.

4. DeepMind та AI для жестової мови . У дослідницькому відділі DeepMind також розглядають можливості використання штучного інтелекту для розпізнавання жестової мови. Використовуючи технології глибокого навчання та нейронних мереж, дослідники працюють над створенням моделей, які можуть розпізнавати жести з високою точністю. Для цього використовуються датасети з багатокамерними записами жестових діалогів. Подібні технології можуть бути застосовані і для української мови жестів, якщо будуть зібрані відповідні датасети.

5. Розпізнавання української жестової мови на основі ключових точок з використанням рекурентних нейронних мереж.

У рамках цього дослідження ведеться розробка системи для розпізнавання жестів дактильної абетки. Основна увага приділяється дактильній системі через її структурованість та стандартизованість. Калькулююча система жестів наразі ігнорується через її складність, контекстуальність та більшу залежність від граматики жестової мови. Такий підхід дозволяє зосередитися на створенні високоточної моделі, орієнтованої на специфічну задачу, що є важливим для подальшого розвитку автоматизованих систем розпізнавання жестової мови.

Реалізація розпізнавання української мови жестів наразі ускладнюється через кілька факторів:

– відсутність великих датасетів, для ефективного навчання штучного інтелекту потрібні великі набори даних, що включають різні варіанти жестів, які виконуються різними людьми у різних контекстах. для англійської жестової мови такі датасети вже існують, але для умж їх бракує;

– унікальність умж, українська мова жестів має власну граматику, синтаксис і правила, які відрізняються від інших жестових мов. це означає, що просто адаптувати існуючі рішення для asl або інших мов до умж буде недостатньо. необхідна розробка специфічних моделей, які враховують особливості саме української жестової мови;

– технічні обмеження, важливою частиною умж є не тільки рухи рук, але й міміка та положення тіла, що є складним для точного комп'ютерного розпізнавання. для повноцінного розуміння мовлення на жестах потрібно використовувати декілька камер та алгоритми, здатні аналізувати тривимірні рухи.

Для того, щоб реалізувати повноцінне розпізнавання української мови жестів, потрібно пройти кілька етапів:

– створення великих датасетів умж з записами жестової мови у різних ситуаціях;

– розробка моделей штучного інтелекту, які будуть тренуватися на цих даних і зможуть розпізнавати не тільки дактиль, а й калькуючу жестову мову;

– інтеграція систем комп'ютерного зору та глибокого навчання для розуміння тривимірних рухів рук і тіла.

Це дозволить створити системи, які не лише допоможуть глухим людям інтегруватися у суспільство, але й сприятимуть кращому розумінню жестової мови серед чуючих людей [7].

Висновки до розділу 1

Технології розпізнавання жестової мови перебувають на етапі активного розвитку, проте повноцінні рішення для української мови жестів досі відсутні. Більшість існуючих систем зосереджені на розпізнаванні дактильної абетки, яка є простішою для реалізації, оскільки кожен жест відповідає одній літері. Однак дактильна абетка лише частково вирішує проблему комунікації, оскільки не охоплює калькуючу жестову мову, яка є важливою для повноцінного спілкування. Калькуюча система жестів включає складніші елементи, такі як синтаксис, контекст, вирази обличчя та рухи тіла, що є критичними для передачі сенсу, проте сучасні системи розпізнавання не враховують ці аспекти. Основні труднощі в реалізації розпізнавання калькуючої мови полягають у відсутності великих датасетів УМЖ та складності технічної обробки не лише рухів рук, а й тіла та міміки. Приклади з англійської жестової мови, такі як SignAll та Google Project Euphonia, демонструють можливості штучного інтелекту й комп'ютерного зору для розпізнавання мови жестів, але ці моделі не можуть бути безпосередньо застосовані до УМЖ через її унікальні особливості. Отже, для створення ефективної системи розпізнавання УМЖ потрібні спеціалізовані дослідження та розробки, орієнтовані на калькуючу мову.

2 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ РОЗПІЗНАВАННЯ ЖЕСТВОЇ МОВИ

Планування та моделювання роботи застосунку для розпізнавання жестів є критично важливим етапом розробки, оскільки від цього залежить точність, ефективність і зручність використання системи. Такий застосунок повинен забезпечувати високу якість розпізнавання як статичних, так і динамічних жестів, враховуючи особливості української жестової мови, зокрема дактильної абетки та калькуючої мови.

Добре спроектована система дозволяє створити технологію, яка адаптується до різноманітних варіантів виконання жестів, враховує індивідуальні особливості користувачів і підтримує реальний час обробки. Крім того, планування включає вибір відповідних інструментів і методів, які сприятимуть швидкому навчанню моделі, інтеграції з іншими компонентами, такими як камера чи мобільний пристрій, і забезпеченню її надійності в різних умовах.

Моделювання роботи застосунку дозволяє заздалегідь передбачити всі технічні особливості та потенційні виклики, що сприяє економії ресурсів на етапі розробки та вдосконалення системи. Це ключовий момент, який забезпечує точність розпізнавання та зручність використання, створюючи додаток, здатний виконувати поставлені завдання в реальному середовищі.

2.1 Інструменти реалізації

Для створення застосунку з розпізнавання жестів було обрано мову програмування Python у поєднанні з бібліотекою YOLOv8. Цей вибір базується на зручності, потужності та ефективності цих інструментів у задачах комп’ютерного зору. Python є ідеальним вибором для проєктів у сфері машинного навчання та штучного інтелекту завдяки своїй простоті, великій кількості спеціалізованих бібліотек і активній спільноті розробників. Бібліотека YOLOv8 є однією з найбільш

сучасних і потужних архітектур для розпізнавання об'єктів, що дозволяє досягати високої точності та швидкості навіть на складних задачах [8].

Python надає широкий спектр інструментів і бібліотек для розв'язання задач комп'ютерного зору, таких як OpenCV для обробки зображень, PyTorch або TensorFlow для побудови нейронних мереж, та NumPy і Pandas для роботи з даними. Простота синтаксису Python дозволяє швидко створювати прототипи і тестувати різні моделі, а підтримка багатьох платформ забезпечує портативність застосунків [9].

Використання Python та бібліотеки YOLOv8 забезпечує ідеальне поєднання гнучкості, продуктивності та точності. Це дозволяє ефективно створювати моделі для розпізнавання жестів, що будуть працювати швидко та надійно. Завдяки таким інструментам, застосунок зможе обробляти як статичні, так і динамічні жести, підтримуючи високу якість розпізнавання та комфорт для користувача**.**

Google Colab є ідеальним інструментом для реалізації нашого дослідження, спрямованого на розпізнавання української жестової мови. Ця безкоштовна хмарна платформа забезпечує доступ до потужних обчислювальних ресурсів, таких як GPU і TPU, що є ключовим фактором для роботи з обчислювально інтенсивними моделями на кшталт YOLOv8. У вашому випадку, коли аналізу підлягають великі масиви даних, зокрема зображення та відеопослідовності, використання GPU значно скорочує час тренування та підвищує ефективність експериментів [10].

Ще однією перевагою є інтеграція Colab із Python-бібліотеками, такими як PyTorch, яка використовується YOLOv8. Це дає змогу швидко створювати середовище для навчання моделей і працювати з великими датасетами без зайвих налаштувань. Також можливість підключення Google Drive спрощує збереження та управління даними, дозволяючи легко організувати зберігання кадрів, метаданих і результатів тренувань.

Особливості Colab, такі як спільне редагування та наявність уже встановлених бібліотек, дозволяють оптимізувати процес дослідження і

взаємодіяти з різними методами, що важливо для ефективного навчання моделі. З огляду на складність задачі, яка передбачає не лише розпізнавання статичних жестів, але й динамічних рухів калькуючої мови жестів, Colab забезпечує необхідну гнучкість для моделювання та оптимізації алгоритмів.

Таким чином, Google Colab стане важливим інструментом, який допоможе нам не лише реалізувати поставлені завдання, а й зробити це ефективно, забезпечуючи високу якість результатів при мінімальних витратах часу та ресурсів.

2.2 Архітектура

YOLOv8 є сучасною архітектурою для розпізнавання об'єктів, яка значно покращила точність і швидкість роботи порівняно з попередніми версіями. Її робота базується на трьох основних компонентах: Backbone, Neck і Head, які разом забезпечують ефективне виділення ознак, їхнє об'єднання та передбачення координат об'єктів [11].

Backbone є базовою частиною моделі, яка витягує ключові ознаки із зображення. У YOLOv8 використовується архітектура на основі Cross-Stage Partial Networks (CSP), яка оптимізує обчислення, зменшує кількість параметрів і зберігає високу продуктивність. Цей компонент відповідає за те, щоб зображення було розкладене на ключові ознаки, які далі обробляються іншими частинами моделі.

Neck з’єднує Backbone із наступним рівнем і об’єднує інформацію з різних рівнів обробки ознак. Завдяки цьому модель здатна враховувати як дрібні деталі, так і загальну структуру зображення. У YOLOv8 використовується підхід FPN та Path Aggregation Networks, які допомагають з’єднувати ознаки з різних шарів.

Head відповідає за остаточне передбачення — визначення класу об'єкта, його координат та розміру. Цей компонент дозволяє моделі передбачати багато об'єктів одночасно на основі виділених ознак, забезпечуючи високу швидкість і точність навіть для складних задач (рис 2.1).

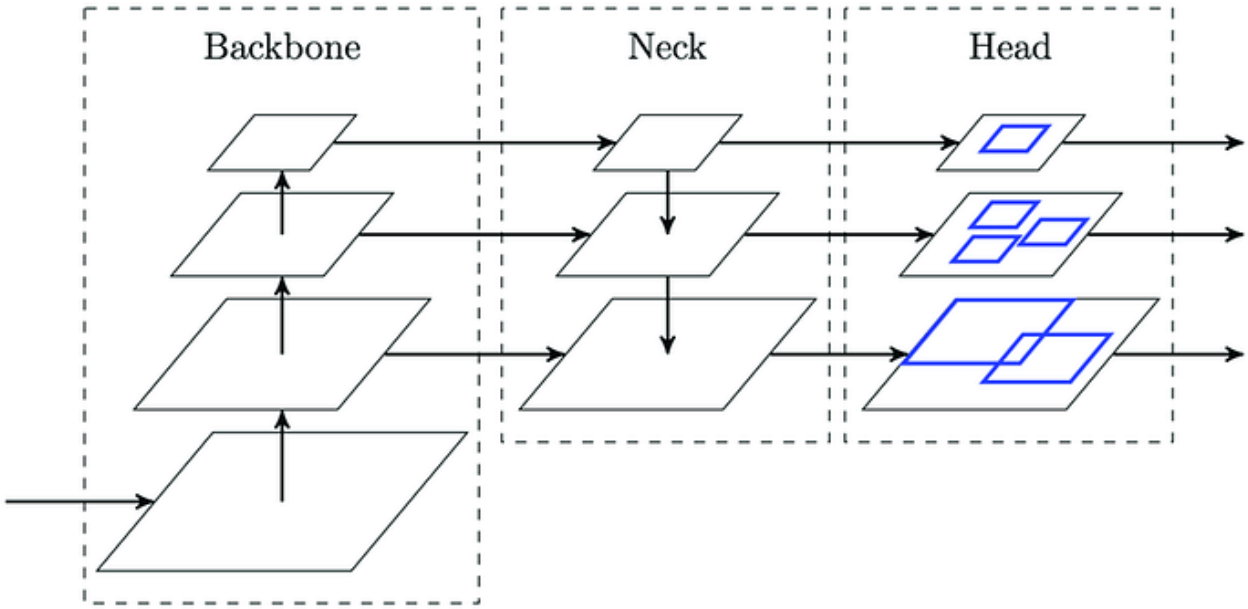

Рисунок 2.1 – Загальна архітектура YOLOv8

Процес аналізу зображення починається з того, що модель розбиває зображення на сітку і для кожної комірки передбачає наявність об'єкта. Далі, використовуючи механізм Non-Maximum Suppression, модель усуває повторювані передбачення та залишає лише найбільш релевантні результати. Завдяки цій архітектурі YOLOv8 може працювати з високою швидкістю навіть на середніх за потужністю пристроях, забезпечуючи точне розпізнавання об'єктів у реальному часі [12].

Але для кращого розуміння розглянемо архітектуру YOLOv8 більш детально і зосередимось на тому що чим вона може допомогти для вирішення нашої задачі (рис. 2.2).

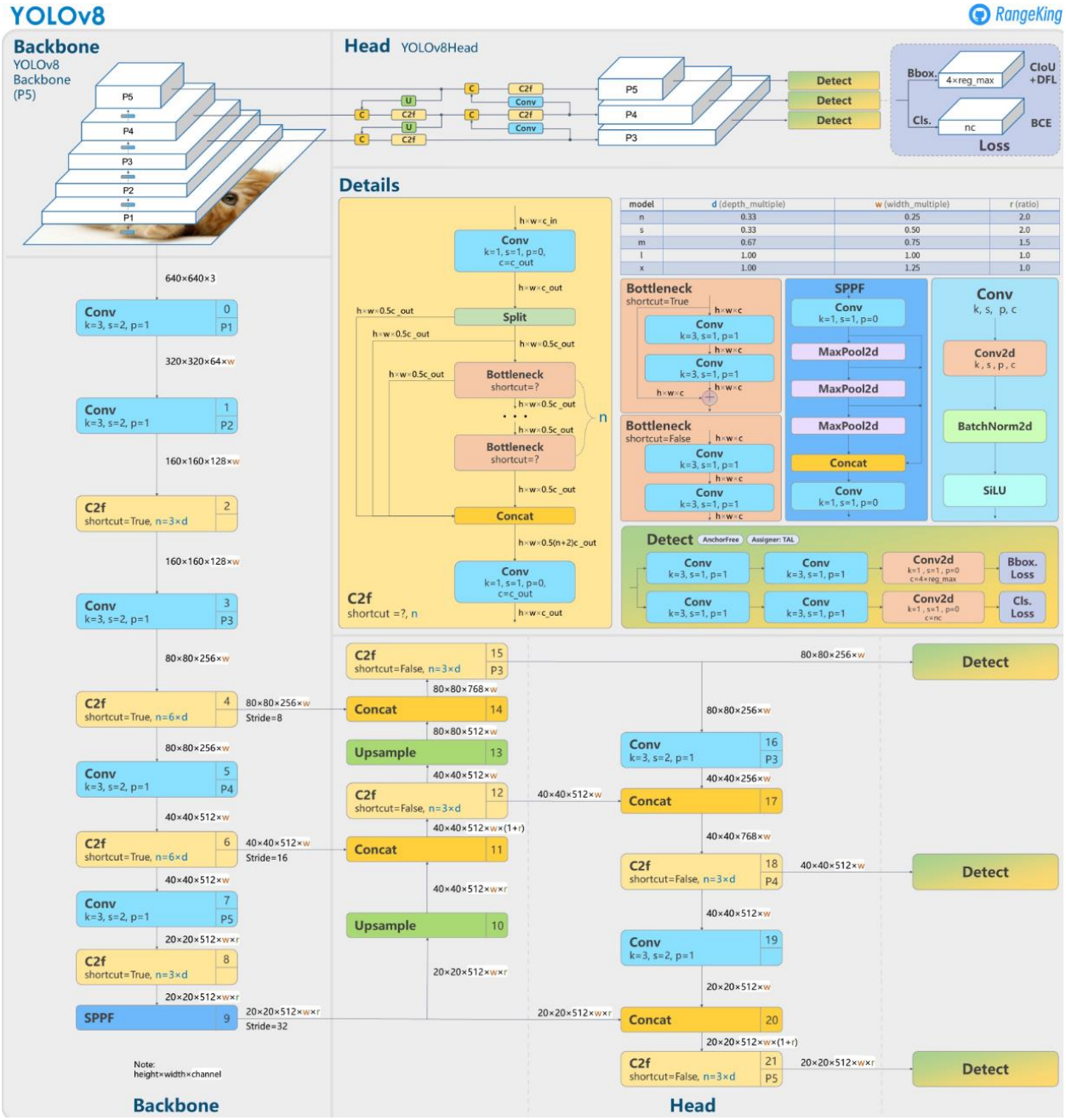

Рисунок 2.2 – Архітектурні особливості YOLOv8

На цьому зображенні детально показана архітектура YOLOv8, яка складається з трьох основних компонентів: Backbone, Neck і Head. Давайте детально розглянемо кожен із цих елементів, а також пояснимо, що відбувається в моделі на кожному етапі.

У моделі YOLOv8-pose ключовими компонентами є Backbone, Neck та Head, кожен із яких виконує специфічну роль у процесі розпізнавання зображень.

Backbone служить для виділення основних ознак із вхідного зображення. Цей компонент використовує згорткові шари та спеціалізовані блоки, такі як Cross-Stage Partial Block і Spatial Pyramid Pooling Fast, для ефективного виділення ознак різного масштабу, що формують базову структуру даних для подальшого аналізу.

Neck об’єднує виділені ознаки з різних рівнів Backbone, враховуючи як дрібні деталі, так і великі об’єкти. Завдяки використанню таких елементів, як Upsample для збільшення розміру ознак і Concat для об'єднання різнорівневої інформації, Neck забезпечує контекстну взаємодію між ознаками, що підвищує точність розпізнавання [13].

Head відповідає за остаточне передбачення об'єктів, їхніх координат і класів. Для цього модель аналізує інформацію, отриману з Neck, і робить точні передбачення для кожної області зображення. Крім того, функції втрат, такі як Bounding Box Loss, Classification Loss і Distributional Focal Loss, оптимізують точність рамок, класифікації та локалізації. Ці етапи дозволяють моделі навчитися ефективно розпізнавати об'єкти навіть у складних умовах.

Також оглянемо процес аналізу зображення в системі YOLOv8n-pose:

– обробка зображення в Backbone, зображення передається через кілька згорткових шарів і блоків CSP, щоб виділити ключові ознаки. На цьому етапі модель формує базову структуру зображення;

– об’єднання ознак у Neck, інформація з різних рівнів масштабу об'єднується, щоб враховувати і дрібні деталі, і великі об'єкти;

– передбачення в Head, використовуючи ознаки, отримані з Neck, модель здійснює остаточне передбачення для кожної комірки сітки зображення. Для цього враховуються координати об'єкта, його розмір і клас.

У результаті, YOLOv8-pose поєднує потужність згорткових мереж, ефективність об’єднання ознак та механізми оптимізації, що забезпечує високу точність і швидкість розпізнавання [14].

YOLOv8 є ідеальним вибором для розпізнавання жестів, оскільки її архітектура ефективно обробляє як статичні, так і динамічні зображення, що є критичним для аналізу жестової мови. Основні переваги YOLOv8 включають (таблиця 2.1):

– висока точність розпізнавання, YOLOv8 має вдосконалені механізми навчання, що дозволяють досягти точності на рівні найкращих моделей у своїй категорії;

– швидкість, YOLOv8 здатна працювати в реальному часі навіть на середніх за потужністю пристроях, що робить її оптимальною для застосунків з обмеженими ресурсами;

– гнучкість, Бібліотека підтримує навчання на користувацьких датасетах, зокрема тих, що містять жестові мови;

– зручність інтеграції, YOLOv8 має простий інтерфейс для роботи з Python і добре документовані інструменти.

Таблиця 2.1 – Порівняння YOLOv8 з іншими фреймворками

Фреймворк	Точність (mAP)	Швидкість (FPS)	Розмір моделі	Легкість інтеграції	Гнучкість навчання
YOLOv8	76%	120	Середній	Висока	Висока
Faster R-CNN	74%	12	Великий	Середня	Висока
SSD	68%	70	Малий	Середня	Середня
RetinaNet	73%	18	Великий	Низька	Середня
YOLOv5	75%	80	Середній	Висока	Висока

З цієї таблиці видно, що YOLOv8 перевершує інші моделі за швидкістю та точністю, що є вирішальним фактором для задач розпізнавання жестів, особливо в реальному часі. У контексті розпізнавання жестової мови ця архітектура є

надзвичайно корисною, оскільки вона дозволяє моделі одночасно аналізувати положення пальців і рухи руки в просторі.

Також необхідно проаналізувати чому використовується саме 8 версія YOLO, а не інші, це буде зображено на наступному рисунку (рис. 2.3).

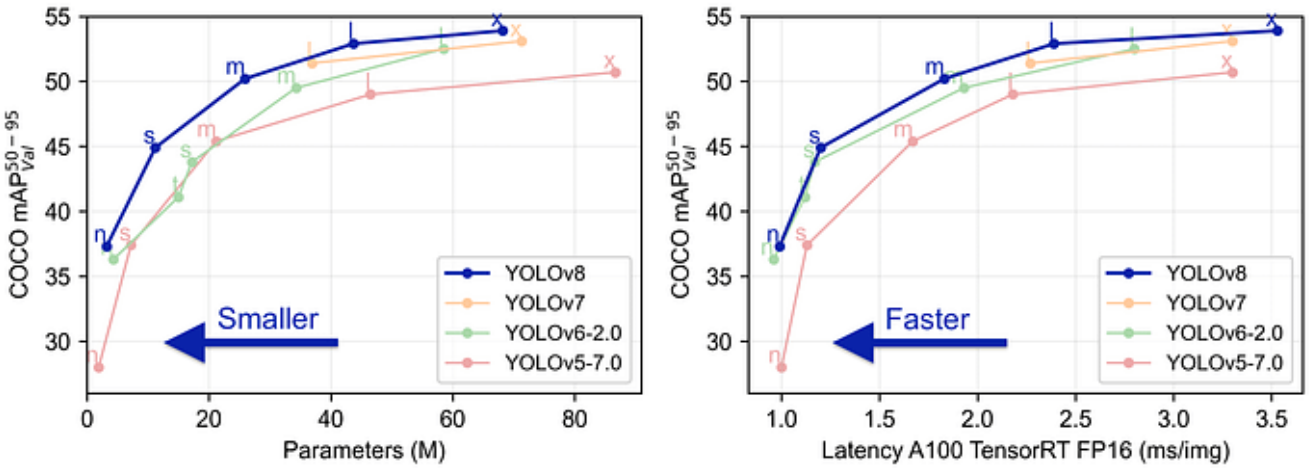

Рисунок 2.3 – Переваги YOLOv8 над попередніми версіями

На представлених графіках порівнюються моделі YOLOv5, YOLOv6, YOLOv7 та YOLOv8 за двома ключовими параметрами: точністю (COCO mAP 50–95) та швидкістю обробки (латентністю). Перший графік демонструє залежність точності від кількості параметрів моделі, що визначає її складність. YOLOv8 має найбільшу точність серед усіх версій, навіть при однаковій кількості параметрів. Це свідчить про те, що вона здатна краще знаходити баланс між складністю моделі та її продуктивністю [15].

Другий графік показує, як змінюється точність залежно від затримки обробки одного зображення. Йдеться про швидкість, з якою модель виконує аналіз. YOLOv8 обробляє зображення значно швидше, ніж попередні версії, зберігаючи при цьому вищу точність. Завдяки цьому YOLOv8 ідеально підходить для задач реального часу, таких як розпізнавання жестів, де важлива швидкість обробки.

Основна перевага YOLOv8 полягає в її здатності забезпечувати високу точність та швидкість завдяки оптимізованій архітектурі. Використання нових

компонентів, таких як C2f, дозволяє досягти ефективності без значного збільшення обчислювальних ресурсів. У контексті розпізнавання жестової мови, ця модель забезпечує точний аналіз як статичних, так і динамічних жестів, що робить її ключовим елементом для створення систем жестового спілкування в реальному часі [16].

Висновки до розділу 2

У процесі опису інструментарію, який буде використовуватись для реалізації дослідження розпізнавання жестової мови, було обґрунтовано вибір ключових технологій, таких як мова програмування Python та архітектура YOLOv8. Ці інструменти обрано з огляду на їх функціональність, продуктивність та адаптивність до вирішення задач комп’ютерного зору.

Python, як основна мова програмування, забезпечує зручне середовище для роботи з бібліотеками, які широко використовуються у сфері машинного навчання та комп’ютерного зору. Його простий синтаксис та величезна кількість відкритих бібліотек (таких як PyTorch, OpenCV і NumPy) значно полегшують процес розробки і тестування моделі. Python дозволяє швидко інтегрувати різноманітні алгоритми та проводити гнучке налаштування моделі, що робить його ідеальним вибором для досліджень у галузі комп'ютерного зору.

YOLOv8 було обрано як основу для моделювання завдяки його вдосконаленій архітектурі, яка є результатом еволюційного розвитку попередніх версій YOLO. У порівнянні з попередниками, YOLOv8 забезпечує значно вищу точність розпізнавання при збереженні низької затримки та високої швидкості обробки зображень. Завдяки його модульній структурі, включаючи компоненти Backbone, Neck та Head, YOLOv8 дозволяє ефективно аналізувати як прості, так і складні зображення. Крім того, оптимізація алгоритмів та впровадження сучасних підходів до обробки зображень гарантують конкурентоспроможність YOLOv8 серед інших нейронних архітектур.

Важливим аспектом є те, що ці технології не лише забезпечують високу точність і швидкість, але й дозволяють масштабувати дослідження. Python дає можливість інтегрувати систему з іншими інструментами, наприклад для створення графічного інтерфейсу користувача чи API, тоді як YOLOv8 легко адаптується до роботи з великими обсягами даних і різними форматами. Це є особливо важливим для розпізнавання жестової мови, адже система має бути здатною працювати в реальному часі, аналізуючи широкий спектр рухів, які можуть мати різні контексти залежно від користувача.

3 КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ ДЛЯ ДАТАСЕТУ

3.1 Датасети і їх роль у розпізнаванні жестів

Датасети відіграють ключову роль у розвитку сучасних технологій, особливо у сфері штучного інтелекту, машинного навчання та комп'ютерного зору. Вони є основою (рис. 3.1-3.2), на якій навчаються та вдосконалюються алгоритми, дозволяючи їм робити прогнози, розпізнавати образи, аналізувати дані і приймати рішення. У кожній галузі, де потрібне автоматизоване оброблення інформації, від точності й обсягу датасетів залежить ефективність кінцевої системи. Чим більший і різноманітніший набір даних, тим краще модель зможе адаптуватися до різних умов і розпізнавати тонкі відмінності в даних. Важливість датасетів особливо яскраво проявляється у таких складних завданнях, як розпізнавання мови жестів, де кожен жест, рух тіла або міміка повинні бути точно зафіксовані та правильно інтерпретовані системою. Відсутність достатніх даних або їхня низька якість призводить до помилок у роботі алгоритмів і знижує точність розпізнавання, що робить датасети центральною ланкою успіху в реалізації будь-якої технології на основі штучного інтелекту [16].

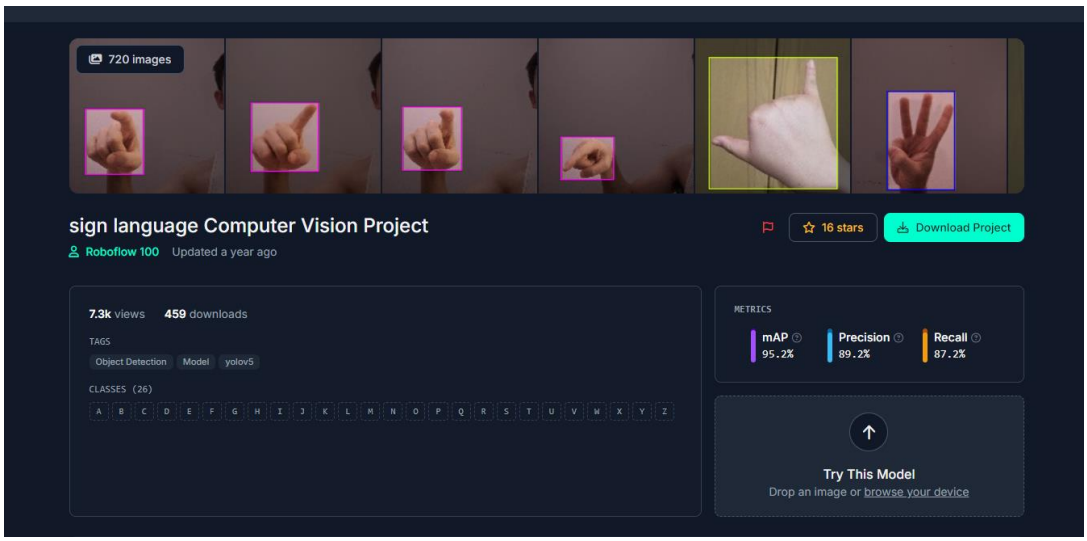

Рисунок 3.1 – Приклад датасету у вільному доступі для американської жестової мови

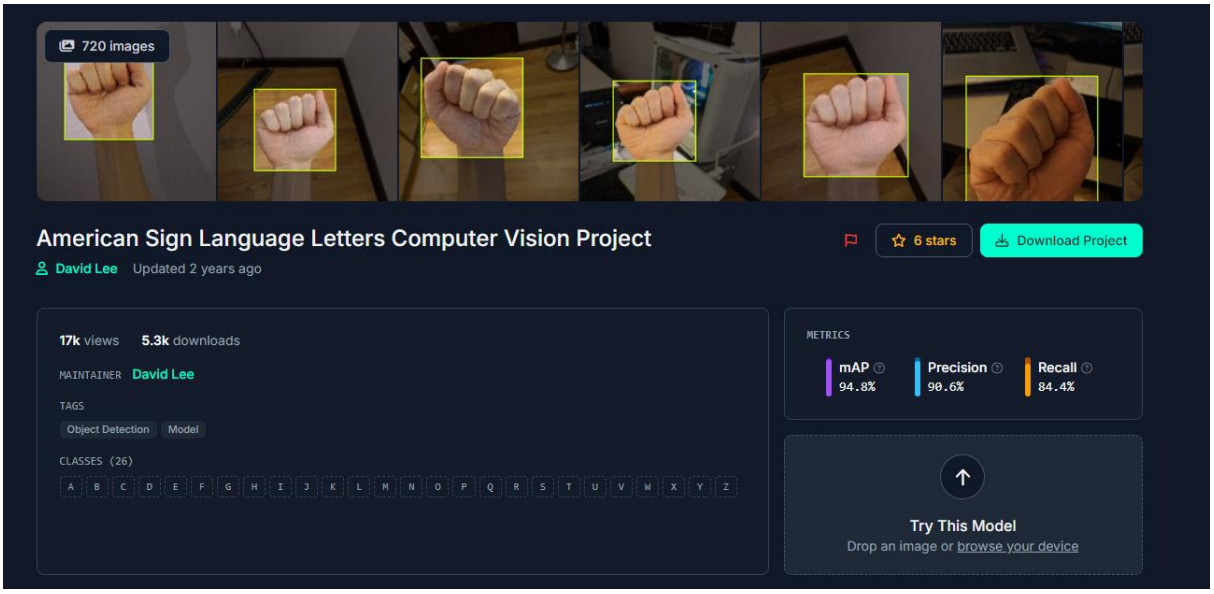

Рисунок 3.2 – Приклад датасету у вільному доступі для американської жестової мови

Датасети надають не тільки базу для навчання моделей, але й допомагають забезпечити їх узгодженість із реальним світом. Наприклад, для побудови системи розпізнавання мовлення або жестів необхідно зібрати великий обсяг реальних прикладів, які відображають різноманітність акцентів, регіональних особливостей, темпів мовлення, міміки, рухів рук і тіла. Ці дані дозволяють алгоритмам машинного навчання виявляти загальні закономірності і одночасно враховувати відхилення та унікальні варіанти виконання жестів. Таким чином, правильно структурований датасет є основою для створення моделі, яка не тільки навчається виконувати завдання, але й робить це з високим рівнем точності та стабільності [17].

Для розпізнавання мови жестів, де комунікація включає динамічні рухи, позиції рук, тіла та вирази обличчя, важливість датасетів не може бути переоцінена. Лише великий і різноманітний набір даних здатен навчити систему штучного інтелекту правильно інтерпретувати складну систему жестової мови, включно з різними регіональними варіантами і особливостями конкретних мов жестів, таких як українська жестова мова (УМЖ). Відсутність таких великих і добре

структурованих наборів даних є головною перешкодою на шляху до створення ефективних систем автоматичного перекладу жестової мови, що ще раз підкреслює критичну важливість датасетів для розвитку цих технологій.

3.2 Датасети в українській мові жестів

Датасети є критично важливими для розвитку технологій розпізнавання української жестової мови (УМЖ), оскільки на сьогодні вони майже повністю відсутні. Без наявності великих і різноманітних наборів даних процес автоматичного розпізнавання жестів не може бути успішно реалізований. Жестові мови, такі як УМЖ, мають складну структуру, що включає не тільки рухи рук, але й вирази обличчя, положення тіла та контекстуальні елементи, які необхідно враховувати. Це робить їх значно складнішими для розпізнавання, ніж просто дактильна абетка. Для того, щоб системи штучного інтелекту могли вивчати та правильно інтерпретувати жести, потрібні великі датасети, що містять тисячі варіантів виконання тих самих жестів у різних контекстах та від різних людей [18].

Відсутність таких датасетів є серйозною перешкодою на шляху до створення точних і ефективних моделей для розпізнавання УМЖ. Без достатньої кількості прикладів жестів штучний інтелект не може навчитися розрізняти нюанси та варіації жестів, що призводить до низької точності розпізнавання. В українській мові жестів, як і в інших жестових мовах, є багато жестів, які можуть змінювати своє значення залежно від контексту або виконання. Наприклад, однакові рухи рук можуть мати різні значення в залежності від міміки або супутніх рухів тіла. Лише великий і добре структурований датасет може забезпечити системі достатньо інформації для розуміння цих тонкощів[19].

Крім того, відсутність датасетів УМЖ ставить під загрозу розробку універсальних систем комунікації для людей із порушеннями слуху в Україні. Без великих наборів даних неможливо створити моделі, які б могли точно перекладати жести в текст або мову, що обмежує можливості глухих людей у спілкуванні з

чуючими. Це створює цифровий розрив, де технології, які могли б допомогти інтеграції людей з вадами слуху в суспільство, залишаються недоступними через відсутність критичних даних для їхнього розвитку.

Наявність датасетів також важлива для збереження й розвитку самої української жестової мови. Оскільки жестові мови є живими системами комунікації, вони постійно змінюються й адаптуються до нових умов. Запис і збереження жестів у датасетах може допомогти у дослідженні та документуванні УМЖ, що дозволить краще розуміти її структуру та забезпечити збереження культурної спадщини. Тому створення й підтримка таких датасетів є важливим не тільки для технічного розвитку, але й для підтримки жестової мови як частини української культури.

Тому було прийнято рішення створити власний датасет за допомогою сервісу Robolow (рис. 3.3).

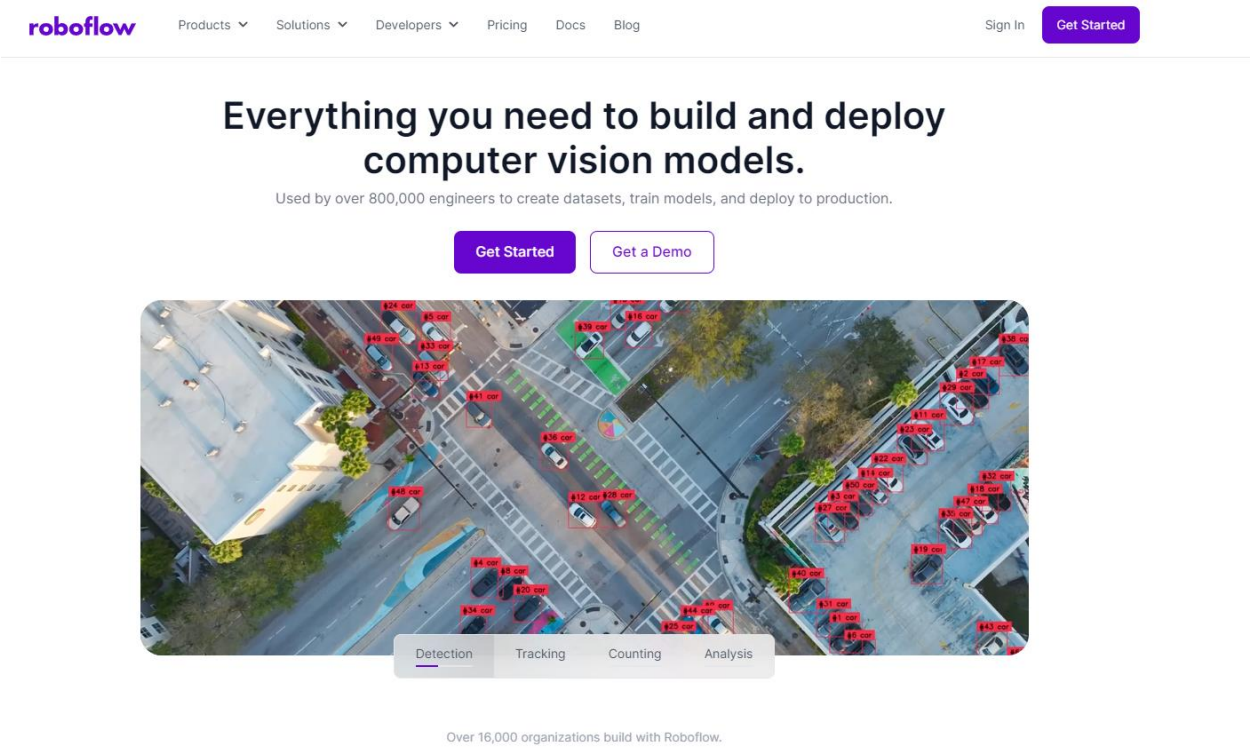

Рисунок 3.3 – сервіс для створення та обробки датасетів

Roboflow має кілька важливих переваг для створення датасетів з розпізнавання жестів, особливо в контексті жестової мови, зокрема української жестової мови. Ось основні переваги цього сервісу [20]:

– простий і зручний інтерфейс, Roboflow надає інтуїтивно зрозумілий інтерфейс для створення, обробки та анотації датасетів. Навіть користувачі без глибоких технічних знань можуть швидко навчитися завантажувати зображення або відео з жестами, а також анотувати їх для подальшого використання в моделях штучного інтелекту;

– автоматизовані інструменти для анотації, робота з датасетами жестів може бути трудомісткою, адже кожен жест повинен бути точно позначений. Roboflow пропонує автоматизовані інструменти для анотації об’єктів на зображеннях або у відео, що дозволяє суттєво зменшити час і зусилля, необхідні для підготовки якісних датасетів;

– гнучка обробка зображень, Roboflow надає інструменти для попередньої обробки даних, включаючи функції нормалізації, зміни розмірів, аугментації (збільшення різноманітності датасету за допомогою модифікацій зображень). Це важливо для жестових датасетів, оскільки дозволяє створювати додаткові варіанти зображень (наприклад, змінювати кут огляду чи освітлення), що сприяє кращому навчанню моделей розпізнавання жестів у різних умовах;

– інтеграція з популярними фреймворками, Roboflow підтримує експорт датасетів у різні формати для машинного навчання (наприклад, TensorFlow, PyTorch, YOLO), що спрощує інтеграцію створених датасетів з моделями штучного інтелекту. Це особливо корисно для розробників, які працюють над створенням моделей розпізнавання жестів;

– можливість спільної роботи та масштабованість, Roboflow дозволяє командно працювати над створенням і вдосконаленням датасетів. Це корисно для масштабних проєктів, таких як створення датасетів для УМЖ, де важливо мати доступ до різних джерел і працювати спільно над підвищенням якості даних.

3.3 Створення класів

Roboflow пропонує зручний і потужний інструментарій для роботи з датасетами, включаючи можливість створення класів і використання кісткової структури для анотації жестів (рис. 3.4 - 3.5). Це особливо корисно для створення складних моделей, які займаються розпізнаванням жестової мови, де важливі як статичні пози рук, так і динамічні рухи [21].

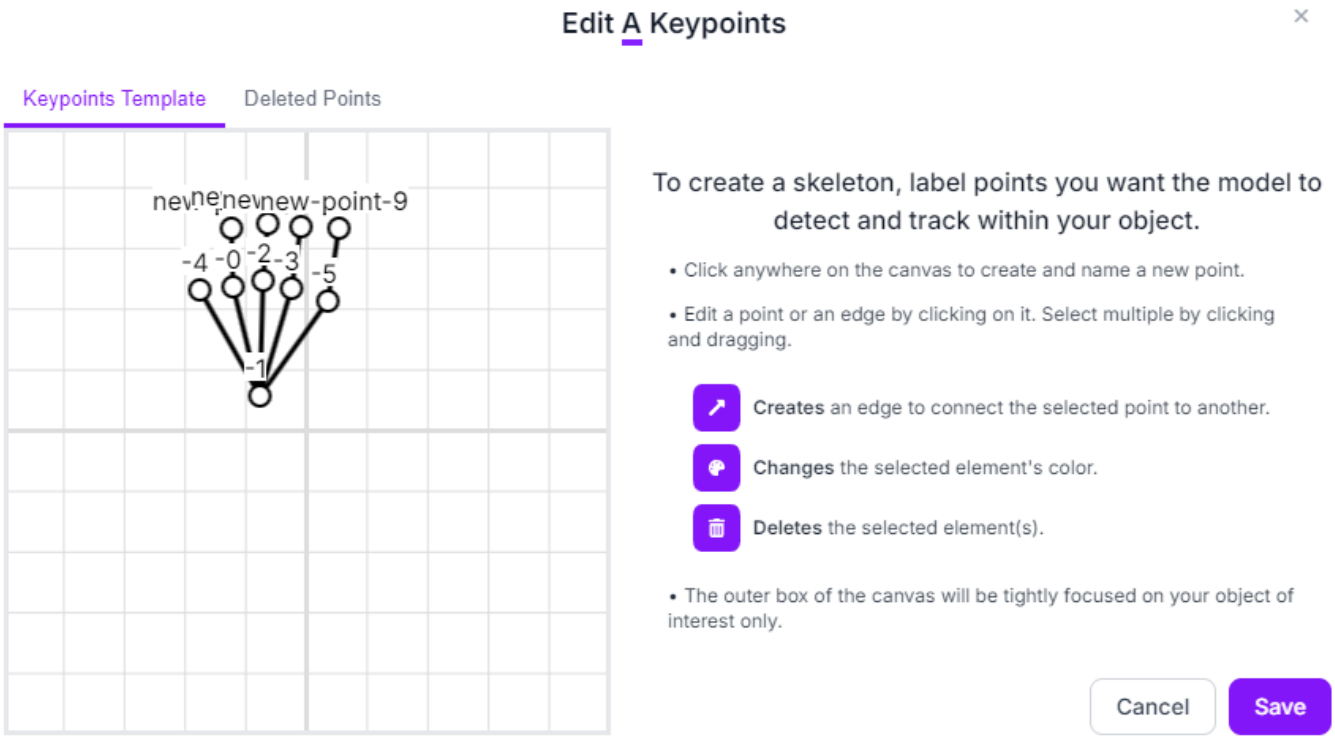

Рисунок 3.4 – Кісткова структура літери А

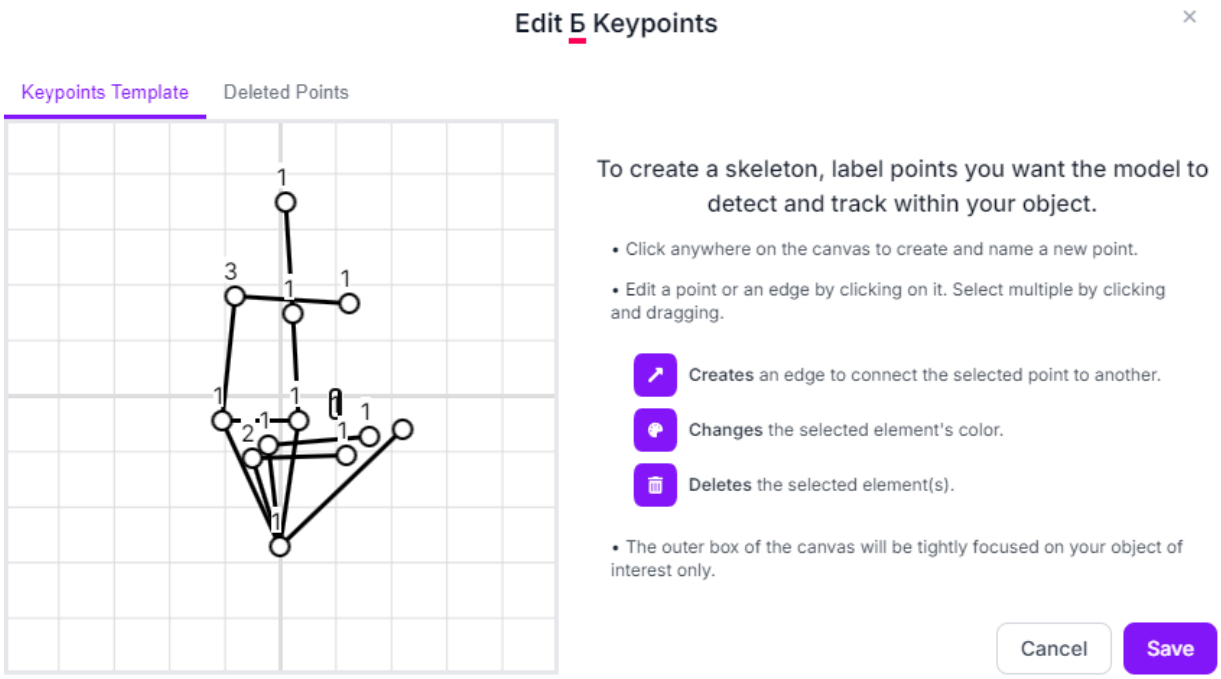

Рисунок 3.5 – Кісткова структура літери А

Класи в контексті машинного навчання дозволяють структурувати різні об’єкти або дії, які необхідно розпізнавати моделлю. У випадку з жестовою мовою, класи можуть бути використані для визначення конкретних жестів або серій жестів, таких як букви дактильної абетки, цифри або складні рухи, що представляють цілі слова [22].

– Додавання зображень до класів, для кожного жесту завантажуєте зображення або відео, які відповідають цьому жесту. Наприклад, ми працюємо з дактильною абеткою української жестової мови, то ми створюєте окремі класи для кожної літери («А», «Б», «В» тощо) і завантажуєте до цих класів відповідні зображення або відеофрагменти, де користувач виконує ці жести.

– Анотація зображень, після завантаження матеріалів, можна використовувати анотаційні інструменти Roboflow для позначення ключових точок або контурів жесту (рис. 3.6).

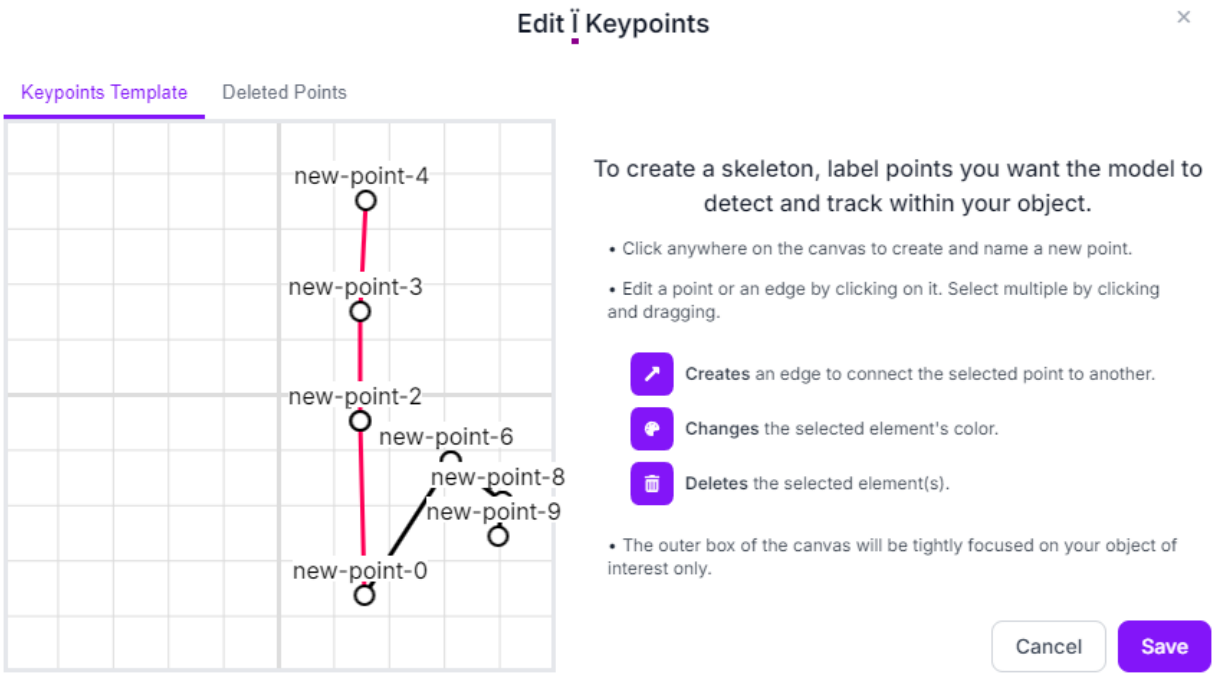

Рисунок 3.6 – Елементи жесту що повині рухатись позначимо іншим кольором

Кісткова структура (skeleton) є важливим підходом для розпізнавання жестів у динаміці. Замість того, щоб просто розпізнавати фіксоване зображення руки, кісткова структура дозволяє визначати ключові точки на тілі, що представляють суглоби (наприклад, зап’ястя, лікті, плечі, пальці) і взаємозв'язки між ними:

– ключові точки, для анотації кісткової структури, користувачі можуть вказати ключові точки на зображенні, наприклад, на кожному суглобі пальців, зап’ястях, ліктях і плечах. Ці точки потім об’єднуються лініями, утворюючи кісткову структуру;

– анотація руху, використовуючи кісткову структуру, можна анотувати не тільки фіксовані позиції рук, але й траєкторії руху рук і пальців у просторі. Це є ключовим для розпізнавання динамічних жестів або жестових фраз, де один жест змінює інший у швидкій послідовності;

– модель розпізнавання динаміки, робота з кістковою структурою дозволяє створювати моделі, здатні розпізнавати не тільки статичні жести, а й

послідовні рухи. Це важливо для калькуючої жестової мови, де значення часто змінюється залежно від контексту рухів рук та їх взаємодії з іншими частинами тіла [40].

Переваги кісткової структури для жестової мови:

– точність і гнучкість, кісткова структура забезпечує моделі можливість краще розуміти складну динаміку жестів. Замість простого виявлення контурів рук, система може враховувати весь комплекс рухів і положення суглобів. Це особливо корисно для жестової мови, де кожен жест може мати кілька варіацій залежно від положення рук і взаємодії з іншими частинами тіла;

– розпізнавання у динаміці, робота з кістковою структурою робить систему більш стійкою до різних варіантів виконання одного і того ж жесту, оскільки вона враховує рухи в реальному часі. Це забезпечує модель здатністю розпізнавати динамічні жести, де важливим є не тільки статичне положення рук, але й рух і його напрямок.

– Інтеграція аугментації з кістковою структурою, аугментація даних — це процес створення додаткових зразків з наявних даних шляхом їх модифікації, наприклад, зміни кута огляду, масштабу або яскравості зображення. Roboflow підтримує аугментацію для датасетів з кістковою структурою, що дозволяє створювати більше варіантів одного і того ж жесту для підвищення якості навчання моделі:

– зміна куту огляду, аугментація з використанням кісткової структури дозволяє змінювати кут огляду або нахил зображень, не порушуючи анатомічних зв'язків між точками. Це дозволяє моделі навчатись розпізнавати жести під різними кутами і умовами освітлення;

– варіації жестів, за допомогою аугментації можна додати до датасету одні й ті ж самі фото жестів але з зміненою кольоровою схемою що система буде сприймати як інше фото одного й того ж самого жесту.

– Експорт і інтеграція в моделі, Roboflow підтримує експорт анотованих датасетів у різні формати, що дозволяє легко інтегрувати їх з популярними фреймворками для машинного навчання, такими як TensorFlow, PyTorch та інші. Це означає, що після створення датасету з кістковою структурою ви можете експортувати його у формат, який легко підходить для використання у нашій моделі[22].

Для дактильної абетки були сформовані класи з використанням кісткової структури (рис. 3.7). Це означає, що кожен жест, який відповідає окремій літері української жестової мови, отримав свою унікальну анотацію з точками, що відповідають суглобам і положенням пальців, зап’ястя та інших частин руки. Такий підхід дозволяє не лише розпізнавати форму руки в статичній позиції, але й відслідковувати точне положення кожного пальця у просторі. Це підвищує точність навчання моделі для розпізнавання дактильної абетки, оскільки модель отримує детальну інформацію про конфігурацію пальців для кожної літери.

COLOR	COUNT	CLASS NAME	KEYPOINTS
	4	A	11 Edit Keypoints
	11	Address	29 Edit Keypoints
	6	B	12 Edit Keypoints
	11	Ch	6 Edit Keypoints
	5	D	10 Edit Keypoints
	7	E	6 Edit Keypoints
	8	F	7 Edit Keypoints
	9	Father	16 Edit Keypoints
	5	G	10 Edit Keypoints
	11	H	9 Edit Keypoints
	11	I	9 Edit Keypoints
	7	K	11 Edit Keypoints
	10	Kh	7 Edit Keypoints
	9	L	8 Edit Keypoints
	9	M	12 Edit Keypoints
	10	N	14 Edit Keypoints
	10	O	14 Edit Keypoints

Рисунок 3.7 – Класи дактильної абетки з кістковою структурою

Створення класів для калькуючої жестової мови з використанням кейпоінтів має важливе значення через динамічну природу цієї мови (рис. 3.8). Калькуюча жестова мова включає не тільки статичні пози рук, але й складні рухи, що змінюються в часі. Кейпоінти, які фіксують ключові точки на руках (пальці, долоні, зап'ястя), допомагають моделі розпізнавати не лише форму рук, а й відстежувати їх рухи під час жесту. Це важливо, тому що жести можуть змінюватися в процесі виконання, і саме ці зміни несуть сенс [23].

Кейпоінти дозволяють відстежувати кожен етап руху рук, фіксуючи послідовність жесту, що дає змогу моделі краще зрозуміти, як саме виконується жест. Кожен жест може складатися з кількох фаз, і для кожної з них необхідно зафіксувати точне положення рук та пальців. Це допомагає моделі навчитися розпізнавати динаміку жестів, що особливо важливо для калькуючої мови, де рухи рук є важливими для правильного розуміння значення. Кейпоінти також дозволяють враховувати індивідуальні особливості виконання жестів і адаптувати модель до різних варіантів виконання [24].

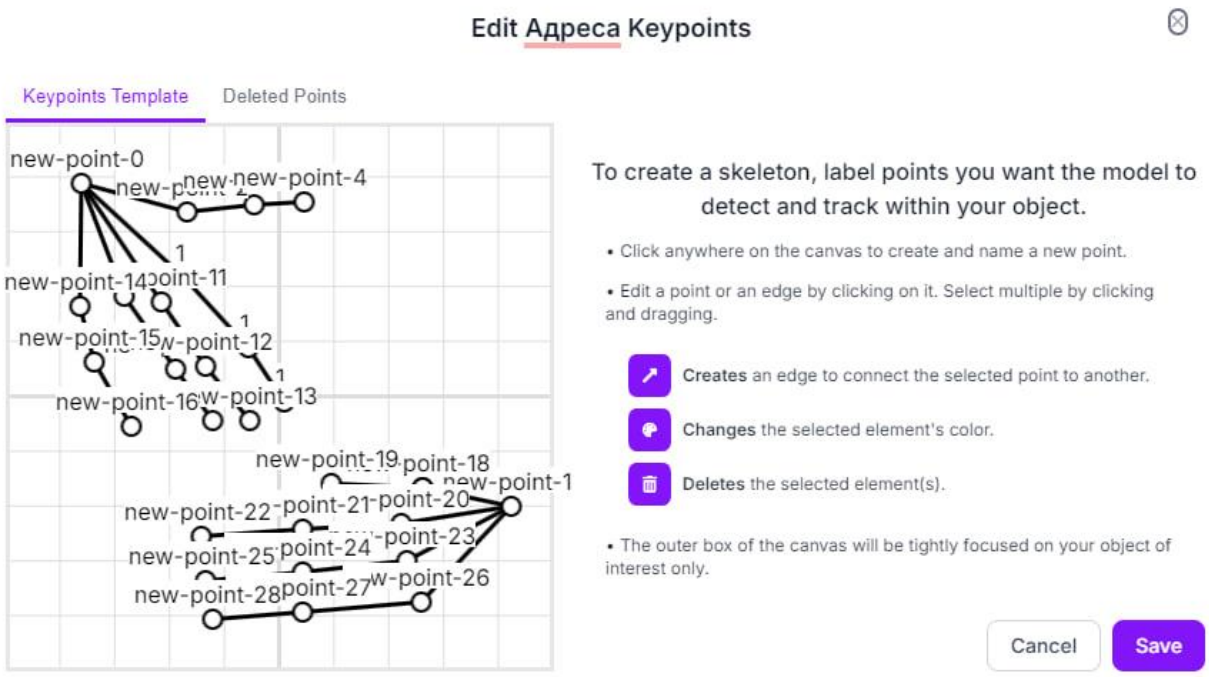

Рисунок 3.8 – Кісткова структура слова адреса

3.3 Анотація зображень

Анотація зображень і розподіл по класах є важливими кроками у створенні датасетів для навчання моделей, особливо для розпізнавання жестової мови. Анотація полягає у позначенні на зображеннях або відео важливих частин, таких як руки та пальці, які беруть участь у жестах. Це допомагає навчальним алгоритмам «розуміти», що саме вони мають розпізнавати на зображенні. Для жестової мови

важливо позначати положення рук і пальців, щоб модель могла навчитися правильно ідентифікувати жести. Розподіл по класах дозволяє відносити кожен жест до певної категорії або класу, наприклад, до конкретної букви дактильної абетки (рис. 3.9 - 3.10). Це допомагає моделі навчитися розрізняти різні жести та правильно їх класифікувати. Без цих процесів модель не зможе ефективно розпізнавати жести або визначати, який жест до якого класу належить, що є ключовим для успішного розпізнавання жестової мови. Анотація даних з використанням кейпоінтів є важливим етапом підготовки даних для тренування моделі, яка розпізнає жести. У процесі анотації кожному жесту присвоюються ключові точки — координати, які позначають певні положення частин тіла, таких як пальці, зап’ястя чи інші важливі суглоби. Кожна з цих точок визначається у вигляді числових значень (x, y) на зображенні. Ці координати дозволяють нейронній мережі зчитувати інформацію про положення й відносний рух частин тіла, забезпечуючи розуміння як статичних поз, так і динамічних рухів. Завдяки цьому система може ефективніше і точніше аналізувати жести, розпізнаючи їх на основі положення частин тіла у просторі та їх взаємодії [25].

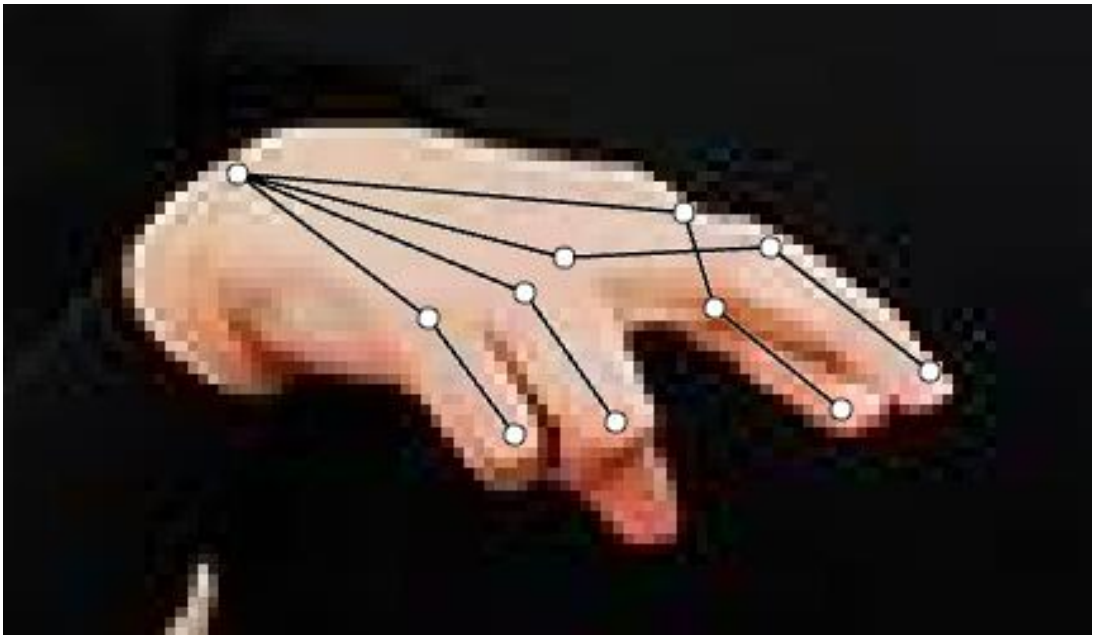

Рисунок 3.9 – Анотація кісткової структури для літери Я

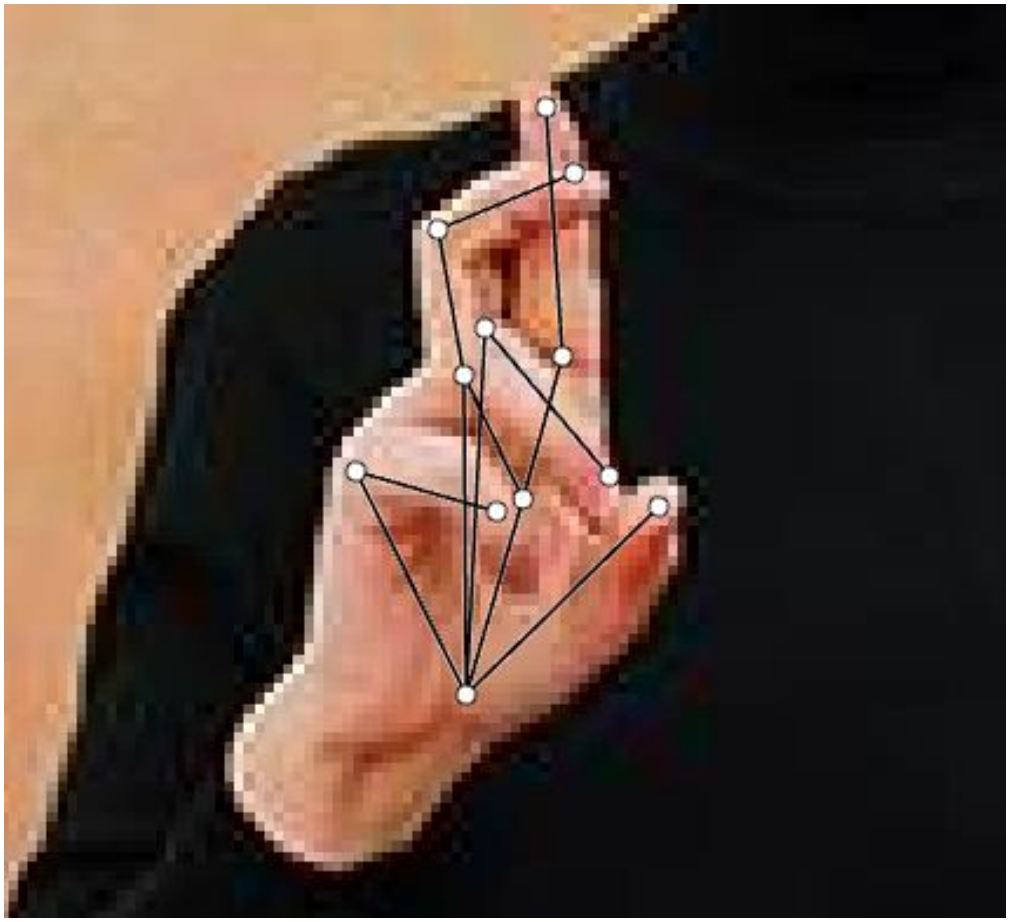

Рисунок 3.10 – Анотація кісткової структури для літери Б

Проте не всі жести мають нерухому структури, а є й такі де важливе як положення пальців так і рухів, для того щоб правильно занотувати їх буде необхідно використати розбиття відео з рухами на кадри і занотувати їх окремо (рис. 3.11). Завантаження відео та його розбиття на окремі кадри є важливим процесом у створенні датасетів для розпізнавання жестової мови, особливо коли потрібно захоплювати не тільки статичні положення пальців, але й динамічні рухи рук. Жестова мова включає не лише окремі позиції рук, але й складні послідовності рухів, що формують слова та вирази. Тому використання відео забезпечує можливість детально відслідковувати кожен етап виконання жесту, зберігаючи контекст руху [26].

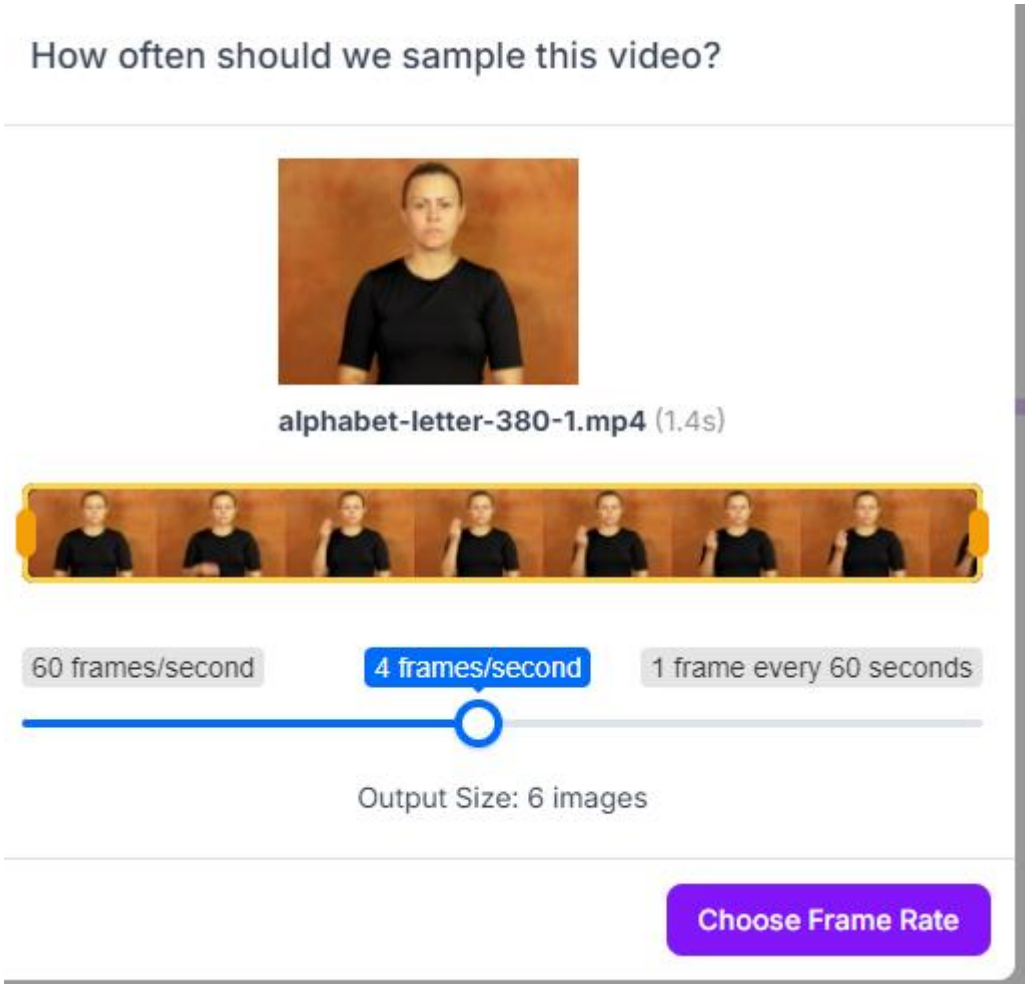

Рисунок 3.11 – Розбиття відео на кадри

Коли відео розбивається на окремі кадри, це дозволяє зафіксувати всі проміжні положення рук і пальців під час виконання жесту. Таким чином, модель отримує більш повну інформацію про динаміку жесту, що дає їй можливість не лише розпізнавати статичні пози, але й правильно інтерпретувати рухи. Відстеження траєкторії рухів є критичним для розуміння складних жестів, де значення може змінюватися залежно від швидкості або напрямку руху [27].

Цей підхід дозволяє моделям навчатися розпізнавати серії послідовних кадрів і коректно ідентифікувати жести, які включають в себе рухи. Завдяки цьому, системи розпізнавання можуть не тільки правильно визначати окремі жести, але й виявляти жести в динамічних умовах, де рухи рук змінюються з часом. Це забезпечує більш точне та природне сприйняття жестової мови моделями

машинного навчання, оскільки відео дозволяє зберігати весь контекст виконання жесту.

Розбиття відео на кадри має особливе значення для розпізнавання калькуючої жестової мови, оскільки ця система жестів часто включає не лише статичні позиції рук, але й важливі рухи та зміни у положенні рук у просторі. На відміну від дактильної абетки, яка здебільшого складається зі статичних жестів, калькуюча жестова мова відображає структуру слів усної мови жестами, що передають послідовності рухів та змін положення рук, які можуть нести сенс [28].

Калькуюча жестова мова є більш динамічною, оскільки кожен жест у ній може складатися з кількох фаз, які відображають послідовність звуків або слів усної мови. Наприклад, один жест може починатися з однієї позиції рук, потім рухатися в певному напрямку або змінювати форму в процесі виконання. Ці зміни важливі для правильного розуміння жесту, оскільки навіть незначні рухи можуть впливати на його значення або інтерпретацію [29].

Коли відео розбивається на кадри, це дозволяє захоплювати кожну фазу виконання жесту (рис. 3.13), фіксуючи всі ключові моменти руху. Для калькуючої жестової мови це особливо важливо, оскільки кожен кадр дає можливість вивчати не тільки кінцеве положення рук, але й шлях, який вони проходять під час жесту. Рухи рук та зміни їх форми є складовими частинами передачі інформації у КЖМ, тому без детальної розбивки на кадри модель не зможе в повній мірі зрозуміти контекст жесту [29].

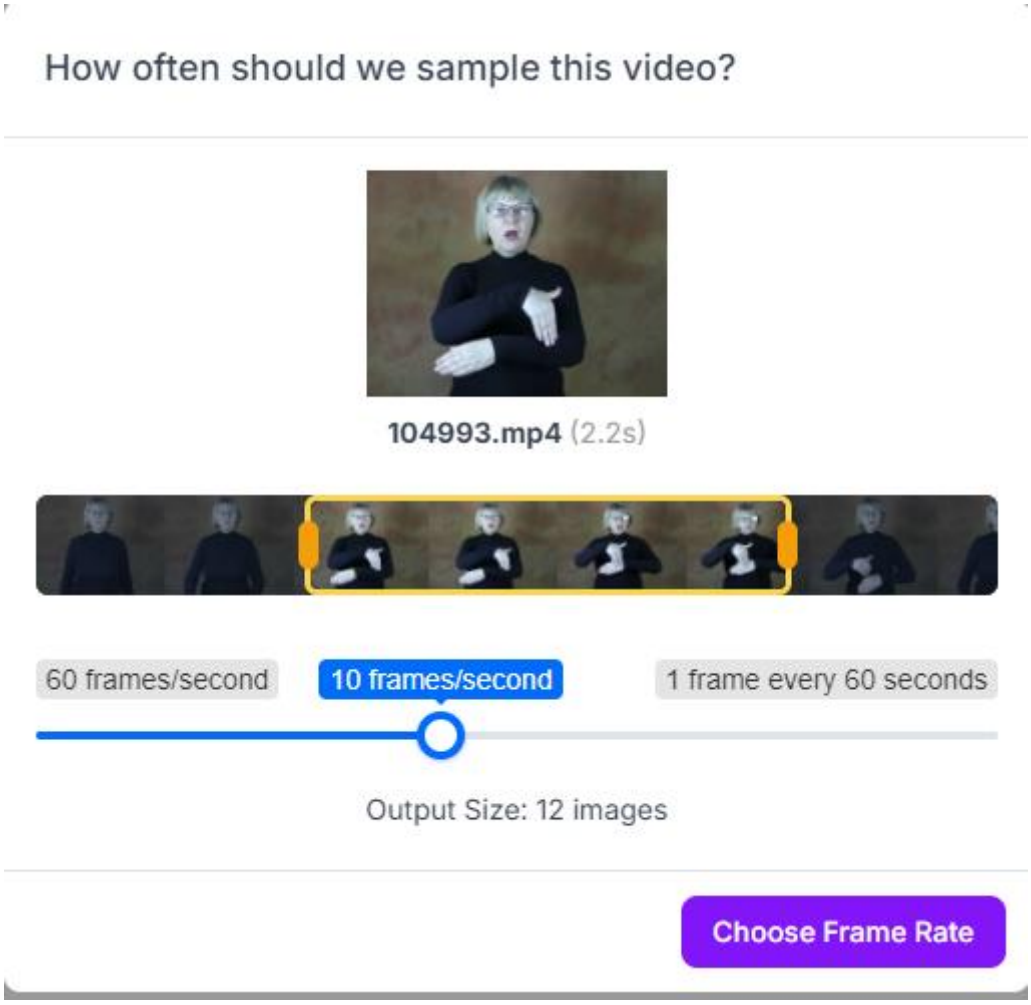

Рисунок 3.12 – Розбиття на фрагменти слова адреса

Це має ще більше значення для навчання моделей машинного навчання, оскільки кожен кадр представляє окремий етап руху (рис. 3.13-3.15). Без такої розбивки на кадри модель могла б втрачати важливі деталі, не розуміючи, як рухається рука під час жесту. У випадку калькуючої жестової мови, де рухи рук можуть відображати мовні структури, фіксування кожного кроку руху є критично важливим. Така деталізація дозволяє моделі вчитися розрізняти різні фази жестів і точно ідентифікувати, який саме жест виконується, навіть якщо його значення змінюється залежно від динаміки [30].

Рисунок 3.13 – Анотування слова адреса

Рисунок 3.14 – Анотування слова адреса

Рисунок 3.15 – Анотування слова адреса

Отже, розбиття відео на кадри дає змогу фіксувати та аналізувати всі тонкощі жестів у калькуючій жестовій мові, що робить цей підхід необхідним для точного розпізнавання жестів і навчання систем машинного зору. Завдяки цьому моделі можуть вчитися не тільки розпізнавати статичні пози рук, але й відслідковувати рухи, які критично важливі для правильного сприйняття та розуміння динамічних жестів у КЖМ [39].

3.4 Отримані результати для навчання

Загалом було створено спеціалізований датасет, що містить 734 зображення жестів (рис. 3.16), призначений для навчання моделі розпізнавання української жестової мови. Датасет був структуровано на три ключові частини, кожна з яких виконувала свою роль у процесі тренування нейронної мережі. Основна частина, тренувальний набір, охоплює 86% зображень і використовується для навчання моделі, дозволяючи їй формувати основні закономірності [38]. Валідаційний набір, який становить 8%, застосовується для моніторингу прогресу під час навчання та

корекції потенційних похибок. Решта 6% була виділена на тестовий набір, що забезпечує незалежну оцінку моделі на нових даних, перевіряючи її здатність до

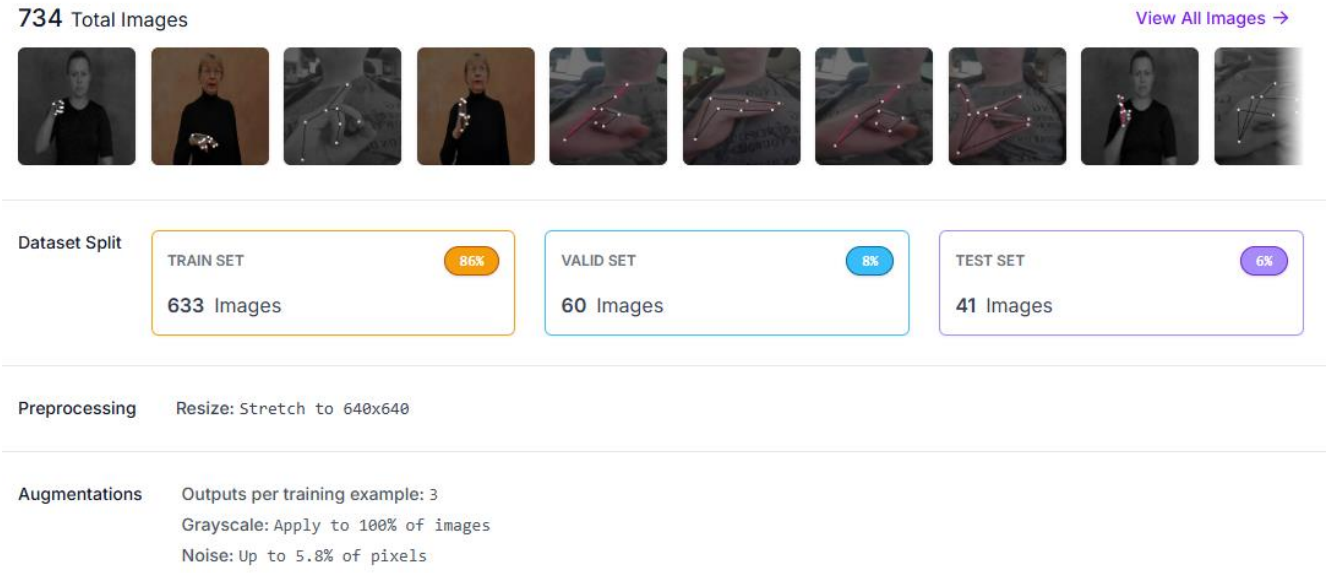

Рисунок 3.16 – Склад датасету

Структура датасету створена з чітким поділом на три основні секції: train, valid і test (рис. 3.17). Кожна з цих секцій містить два підкаталоги — images і labels.

Рисунок 3.17 – Склад датасету експортованого у форматі YOLOv8

У каталозі images зберігаються відповідні зображення, що представляють жести, а у labels — текстові файли з анотаціями, які включають координати кейпоінтів, необхідні для навчання моделі.

– Train основна частина, яка використовується для навчання моделі, дозволяючи алгоритму вивчати залежності в даних.

– Valid призначена для перевірки точності моделі на проміжних етапах, запобігаючи перенавчанню.

– Test використовується для оцінки остаточних показників моделі на даних, які вона раніше не бачила.

Файли з анотаціями у датасеті містять структуровану інформацію про ключові точки, що відповідають положенням рук у жестах. Ці файли прив’язані до відповідних зображень і зберігають дані у вигляді текстових рядків. Кожен рядок включає ідентифікатор класу жесту та набір координат ключових точок, які вказують на певні анатомічні елементи, наприклад, кінчики пальців, суглоби чи зап'ястя. Координати нормалізовані відносно ширини та висоти зображення, що дозволяє універсально працювати з різними розмірами зображень. Логіка опису координат полягає в точному визначенні взаємного розташування кейпоінтів, що є важливим для коректного розпізнавання як статичних поз, так і динамічних рухів жестів. Це забезпечує модель необхідною інформацією для аналізу просторових відносин між точками, дозволяючи ефективно навчатися і розпізнавати жести з високою точністю [31].

Висновки до розділу 3

Було створено класи для дактильної абетки та класи слів для калькуючої жестової мови. Класи дактильної абетки дозволяють моделі розпізнавати окремі букви, що є важливим елементом для точного передачі інформації через жестову мову, особливо в контексті написання слів. Для калькуючої жестової мови були створені класи з урахуванням необхідності більшої кількість кейпоінтів, що

фіксують динамічні зміни положення рук і пальців, важливі для передачі рухів, які часто несуть основний зміст жесту.

Ця робота є основою для навчання моделей розпізнавання жестів, оскільки моделі тепер можуть опрацьовувати як статичні жести дактильної абетки, так і складні рухи, властиві калькуючій жестовій мові. Завдяки цьому система зможе точніше розуміти та ідентифікувати жести, що сприятиме розвитку технологій автоматичного перекладу української жестової мови.

4 СИСТЕМА РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ

При плануванні система розпізнавання жестів має структуру, представлену на блок-схемі (рис. 4.1).

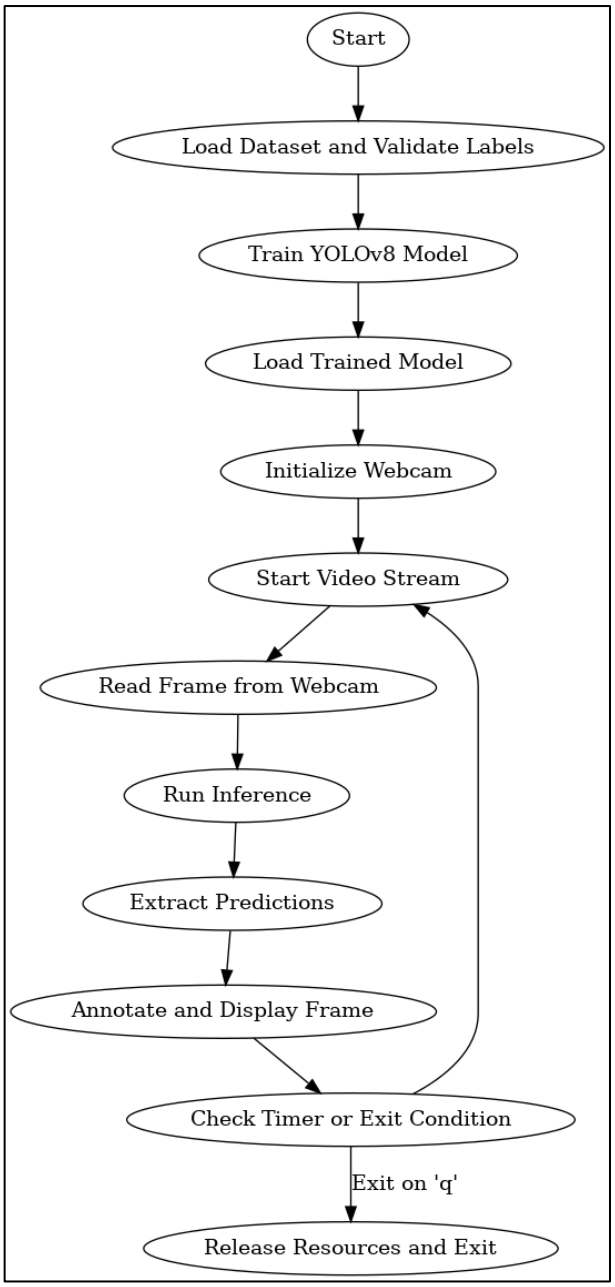

Рисунок 4.1 – Блок-схема інформаційної системи

Спочатку завантажується попередньо навчена модель YOLO, яка спеціалізується на розпізнаванні жестів. Перед початком роботи проводиться

перевірка даних анотацій, щоб забезпечити їхню коректність і узгодженість із кількістю класів.

Далі система підключається до камери, отримує відеопотік і розбиває його на окремі кадри для подальшого аналізу. Кожен кадр передається до моделі, яка виконує інференс, розпізнає жести, повертаючи класи об'єктів і координати ключових точок, таких як положення пальців чи кисті.

Результати обробки кадрів візуалізуються. На кожному кадрі відображаються ідентифіковані жести разом із відповідними підписами, що полегшує верифікацію роботи системи

1.1 Шари, функції втрат і оптимізатори в YOLO8n -pose

У процесі навчання моделі для розпізнавання жестової мови українського жесту кожне зображення, яке є частиною датасету, проходить кілька ключових етапів аналізу. Спершу модель отримує вхідні зображення разом із текстовими файлами анотацій, які містять координати кейпоінтів для кожного жесту. Ці координати включають положення ключових точок рук, таких як пальці, долоня, і навіть загальне розташування рук у просторі. Завдяки цьому модель отримує точну геометричну інформацію про кожен жест.

Протягом кожної епохи навчання модель виконує передбачення на основі отриманих зображень, порівнює ці передбачення із справжніми анотаціями, а потім коригує свої параметри для зменшення функції втрат. Кейпоінти використовуються для навчання моделі розумінню структурного розташування рук, дозволяючи їй враховувати як статичні позиції пальців, так і їхній взаємний рух. Це особливо важливо, оскільки українська жестова мова вимагає точного розуміння як положень, так і напрямків рухів, що визначають значення жесту.

Координати, які включають видимість точки, дозволяють моделі справлятися зі складними випадками, коли частина руки може бути перекрита. Ця інформація критична для забезпечення надійності моделі під час реального розпізнавання

жестів. Завдяки ретельно структурованим анотаціям модель досягає високого рівня точності у розумінні жестів, створюючи потужний інструмент для підтримки спілкування через жестову мову.

На зображенні (рис. 4.2) представлена загальна структура процесу розпізнавання в обчислювальних моделях, включаючи методології, що використовуються в YOLOv8, використанні до задачі розпізнавання жестової мови, зокрема української дактильної абетки.

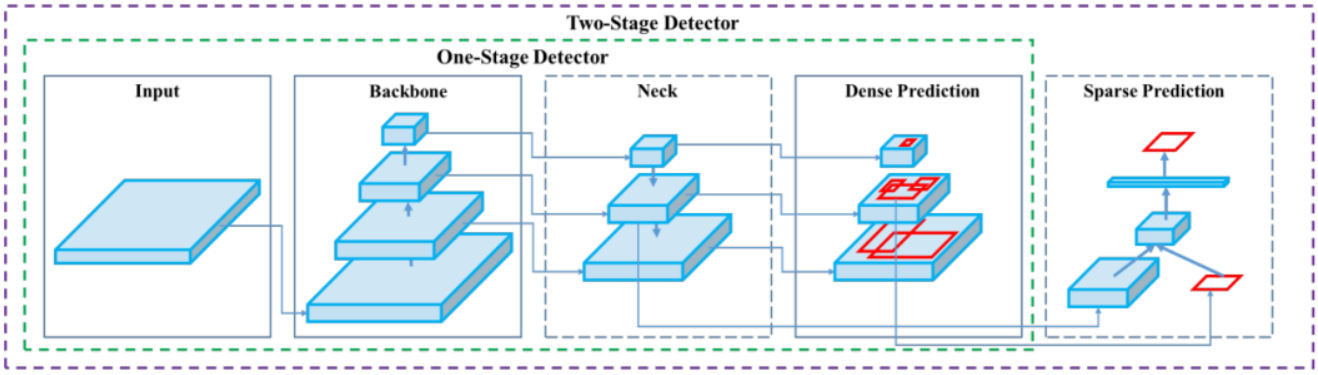

Рисунок 4.2 – Структура процесу розпізнавання

Перший етап — це обробка вхідних даних, де модель приймає на вхід зображення, що містять жести. Зображення нормалізуються й масштабуються до визначених розмірів для узгодженості даних.

Далі відбувається робота основного блоку (, який витягує ключові ознаки з вхідного зображення. Це багаторівнева нейронна мережа, яка будує представлення особливостей, зокрема локальних і глобальних патернів жестів.

Neck виконує функцію побудови багатомасштабних представлень ознак. Цей блок важливий для забезпечення точності виявлення жестів на зображеннях із різними розмірами об’єктів, забезпечуючи більш детальне й цілісне сприйняття.

На етапі щільного прогнозування модель визначає всі можливі області, де може знаходитися ключова точка або рука. Цей процес включає розрахунок координат кейпоінтів і визначення найбільш ймовірного місцезнаходження.

Заключний етап, розріджене прогнозування, дозволяє фільтрувати результати, залишаючи лише ті, що відповідають визначеним критеріям точності. Це гарантує, що на виході залишаються лише релевантні дані [32].

У контексті розпізнавання української жестової мови така структура дозволяє ефективно адаптувати модель до завдань локалізації та класифікації ключових елементів жестів, забезпечуючи високу точність і продуктивність.

Функції втрат у YOLOv8-Pose є ключовим компонентом процесу навчання, забезпечуючи адаптацію моделі до специфічних завдань, таких як локалізація, класифікація та визначення координат ключових точок. Під час навчання модель намагається мінімізувати втрати для кожного з цих аспектів, поступово вдосконалюючи свої передбачення.

Наприклад, функція втрат для рамок оцінює різницю між передбаченими рамками об’єктів і їхніми реальними координатами. Це допомагає моделі правильно обмежувати область, де знаходяться руки чи інші важливі частини жесту. Втрата по ключових точках фокусується на тому, щоб координати кейпоінтів, які передбачає модель, були максимально наближеними до реальних координат. Для цього зазвичай використовується функція, що обчислює відстань між двома наборами точок, як-от L2-норма. Чим менше помилка, тим точніше модель може виявляти позиції пальців у жесті [33].

Передбачення рамок включає визначення координат центральної точки об'єкта (x, y), розмірів та обчислення ймовірності приналежності об'єкта до певного класу. Всі ці значення розраховуються одночасно, завдяки щільному розподілу осередків на зображенні, де кожен осередок відповідає за виявлення об'єктів у своїй зоні.

Функції втрат, такі як CIoU Loss (для регресії рамок) та Classification Loss (для передбачення класів), коригують прогнозовані рамки, навчаючи модель розрізняти справжні рамки та фонові помилки. У результаті модель поступово

вдосконалює свої передбачення, досягаючи точності в ідентифікації та локалізації об'єктів [37].

Також модель оптимізує втрату класифікації, яка визначає, наскільки правильно модель ідентифікує конкретний жест серед доступних класів. Це критично важливо для забезпечення того, щоб кожен жест був правильно розпізнаний.

Під час кожної епохи навчання функції втрат дають моделі зворотний зв’язок (рис. 4.3) про те, наскільки добре вона справляється із завданнями, на основі чого модель оновлює свої параметри. Таким чином, всі втрати працюють разом, щоб зробити систему більш точною та надійною у розпізнаванні жестів.

box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances
1.362	9.752	0.2306	2.753	2.247	16

Рисунок 4.3 – Результати зворотного зв’язку функції втрати при навчанні

Оновлення параметрів у моделі YOLOv8 відбувається через механізм зворотного поширення помилки (backpropagation) і використання оптимізатора (рис. 4.4). Коли модель отримує вхідні дані та робить прогноз, вона порівнює свої результати з реальними значеннями, визначаючи помилку за допомогою функції втрат. Ця помилка надсилається назад через мережу, де оптимізатор коригує ваги та параметри, щоб мінімізувати помилку на наступних ітераціях.

Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances
48/60	0G	1.713	9.753	0.2746	3.279	2.305	41
Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances
57/60	0G	1.363	9.838	0.2346	2.763	2.275	16
Epoch	GPU_mem	box_loss	pose_loss	kobj_loss	cls_loss	dfl_loss	Instances
58/60	0G	1.362	9.752	0.2306	2.753	2.247	16

Рисунок 4.4 – Результат роботи оптимізатора на прикладі результатів 3 епох навчання

Це дозволяє моделі поступово поліпшувати свої прогнози, знижуючи похибку. Але з цими функціями також мають працювати оптимізатори задля покращення навчання та для того щоб не виникла проблема перенавчання. Оптимізатори відіграють важливу роль у навчанні моделей глибинного навчання, зокрема нейронних мереж. Вони керують процесом оновлення параметрів (ваг) мережі для мінімізації функції втрат. Метою оптимізатора є вибір найкращих ваг, які дозволяють моделі ефективно навчатися та досягати найкращих результатів. Різні оптимізатори використовують різні методи коригування швидкості навчання та адаптації до умов даних, що дозволяє покращити точність та швидкість навчання моделі, а також зменшити ймовірність перенавчання[34].

В нашому випадку оптимізатор називається AdamW. AdamW — це варіант оптимізатора Adam, який включає модифікацію для кращого управління регуляризацією ваг. Основною відмінністю від стандартного Adam є те, що AdamW застосовує штраф за величину ваг (L2-регуляризацію) незалежно від того, як оновлюються параметри за допомогою градієнтного спуску. Це дозволяє краще контролювати величину параметрів і зменшувати їх надмірне зростання, що може покращити загальну ефективність моделі, зменшуючи перенавчання [35].

1.2 Процес навчання

Для навчання моделі використовувались вхідні дані, які були визначені у налаштуваннях процесу тренування. Зокрема, модель YOLOv8n-pose налаштовувалась за допомогою конфігураційного файлу yolov8n-pose.yaml, що задає архітектуру мережі. Навчання тривало 60 епох, при цьому використовувались зображення розміром 640 пікселів. Датасет для навчання був визначений через файл data_yaml_path (рис. 4.5), який містив інформацію про шлях до тренувальних, валідаційних та тестових наборів даних, а також категорії і координати ключових точок. Ці налаштування забезпечували відповідний підхід до навчання та адаптації моделі під задачу.

```yaml
train: ../train/images
val: ../valid/images
test: ../test/images

kpt_shape: [29, 3]
flip_idx: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]

nc: 35
names: ['A', 'Address', 'B', 'Ch', 'D', 'E', 'F', 'Father', 'G', 'H', 'I', 'K', 'Kh', 'L', 'M', 'N', 'O', 'P', 'R', 'S', 'Sh', 'Shch', 'T', 'Ts

roboflow:
  workspace: sign-languages
  project: sign-languages-hz11g
  version: 16
  license: CC BY 4.0
  url: https://universe.roboflow.com/sign-languages/sign-languages-hz11g/dataset/16
```

Рисунок 4.5 – Файл data.yaml

Файл data.yaml містить ключову інформацію для навчання моделі YOLOv8-pose, забезпечуючи її правильну інтеграцію з даними. У секції train, val і test вказані шляхи до папок, які містять відповідні набори зображень: для тренування, валідації та тестування моделі. Ці параметри визначають, на яких даних буде навчатися модель, які дані вона використовуватиме для перевірки під час навчання, та які для фінальної оцінки продуктивності.

Параметр kpt_shape задає форму ключових точок, що визначаються як [29, 3]: кількість точок (29) та їхні координати у просторі (x, y) разом з видимістю. Це необхідно для точної ідентифікації жестів.

flip_idx визначає індекси для симетричних точок при дзеркальному відображенні зображень, що є важливим для аугментації даних та підвищення стійкості моделі до змін у положенні жестів.

Параметр nc задає кількість класів (35), а список names визначає їхні імена. Це включає різні літери та жести української жестової мови, необхідні для класифікації.

Секція roboflow містить метаінформацію про джерело датасету, включно з робочим простором, проектом, версією, ліцензією та URL-адресою для завантаження. Це забезпечує відповідність даних ліцензійним вимогам і зберігає датасет у вільному доступі для подальшого використання.

Усі ці параметри формують основу для налаштування навчання моделі, роблячи процес організованим, адаптованим до тренування для розпізнавання мови жестів та подальшого масштабування якщо це буде потрібно [36].

Тепер перейдемо до реалізації самого навчання, почнемо з підготовки залежностей та вказання шляхів (рис. 4.6).

```
from ultralytics import YOLO
import os
import yaml


dataset_path = "C:/Users/Bohdan/Desktop/mag_work/dataset"

data_yaml_path = os.path.join(dataset_path, "data.yaml")
with open(data_yaml_path, "r") as f:
    data_yaml_content = yaml.safe_load(f)

# Витягуємо кількість класів та їх назви з data.yaml
num_classes = data_yaml_content["nc"]
class_names = data_yaml_content["names"]
```

Рисунок 4.6 – Вказання залежностей та передача початкових даних

Цей код виконує кілька ключових дій, пов’язаних із підготовкою до навчання моделі за допомогою датасету. Спочатку, використовуючи модуль os, визначається шлях до файлу data.yaml, який містить конфігурацію датасету. Далі файл відкривається і зчитується за допомогою модуля yaml, який перетворює вміст файлу у формат Python-словника.

Після цього з конфігурації витягується кількість класів та їхні назви. Ці параметри будуть потрібні для налаштування моделі: num_classes вказує на кількість категорій для класифікації, а class_names містить назви жестів або літер, які відповідатимуть кожному класу. Це забезпечує відповідність між даними та алгоритмом розпізнавання. Також через особливості імпорту датасету з roboflow було потрібно перевірити наскільки дані що знаходяться в анотаціях збігаються з тим які мають бути в них згідно опису класів (рис. 4.7).

```python
def validate_labels(labels_path, num_classes):
    """Перевіряє коректність анотацій у файлах."""
    invalid_files = []
    for file in os.listdir(labels_path):
        if file.endswith(".txt"):
            file_path = os.path.join(labels_path, file)
            with open(file_path, "r") as f:
                lines = f.readlines()

            for line in lines:
                parts = line.strip().split()
                if len(parts) < 5:  # Мінімум: class_id + bbox
                    print(f"Некоректний рядок у файлі: {file_path}")
                    invalid_files.append(file_path)
                    break

                # Перевіряємо class_id
                try:
                    class_id = int(parts[0])
                except ValueError:
                    print(f"Некоректний class_id у файлі: {file_path}")
                    invalid_files.append(file_path)
                    break

                if not (0 <= class_id < num_classes):
                    print(f"Некоректний class_id ({class_id}) у файлі: {file_path}")
                    invalid_files.append(file_path)
                    break

    return invalid_files
```

Рисунок 4.7 – Перевірка файлів анотації

Ця функція призначена для перевірки коректності анотацій у текстових файлах, які містяться в папці labels_path. Кожен файл аналізується на відповідність базовим вимогам структури анотацій. Для кожного рядка функція перевіряє, чи вказано коректну кількість параметрів (мінімум клас об'єкта та координати рамки), а також чи належить class_id до допустимого діапазону (від 0 до num_classes-1).

Якщо знаходяться помилки, такі як неправильний формат або недопустимий ідентифікатор класу, шлях до проблемного файлу додається до списку invalid_files. Цей список повертається після перевірки, щоб полегшити виправлення помилок.

Наступним кроком є перевірка файлів та початок навчання за допомогою обраною конфігурації (рис. 4.8).

```
train_labels_path = os.path.join(dataset_path, "train", "labels")
valid_labels_path = os.path.join(dataset_path, "valid", "labels")
test_labels_path = os.path.join(dataset_path, "test", "labels")


invalid_train = validate_labels(train_labels_path, num_classes)
invalid_valid = validate_labels(valid_labels_path, num_classes)
invalid_test = validate_labels(test_labels_path, num_classes)

if invalid_train:
    print("Некоректні файли у train:", invalid_train)
if invalid_valid:
    print("Некоректні файли у valid:", invalid_valid)
if invalid_test:
    print("Некоректні файли у test:", invalid_test)

model = YOLO("yolov8n-pose.yaml")

results = model.train(
    data=data_yaml_path,
    epochs=60,
    imgsz=640,
)
```

Рисунок 4.8 – Початок навчання на обраній конфігурації

Цей код перевіряє коректність анотацій у трьох підмножинах датасета: для тренування, валідації та тестування. Для кожної з них він використовує функцію validate_labels, що шукає некоректні файли анотацій. Якщо такі файли знаходяться, вони виводяться на екран разом із повідомленням про їхню належність до певного набору (train, valid, або test).

Далі ініціалізується модель YOLO з конфігураційним файлом yolov8n-pose.yaml. Після цього виконується тренування моделі, використовуючи вказані параметри: шлях до конфігурації data_yaml_path, 60 епох та розмір зображення 640x640.

Під час навчання нейронної мережі за виведеним логом можна спостерігати кілька ключових метрик та процесів, які показують, як модель поступово адаптується до навчального набору даних (рис. 4.9).

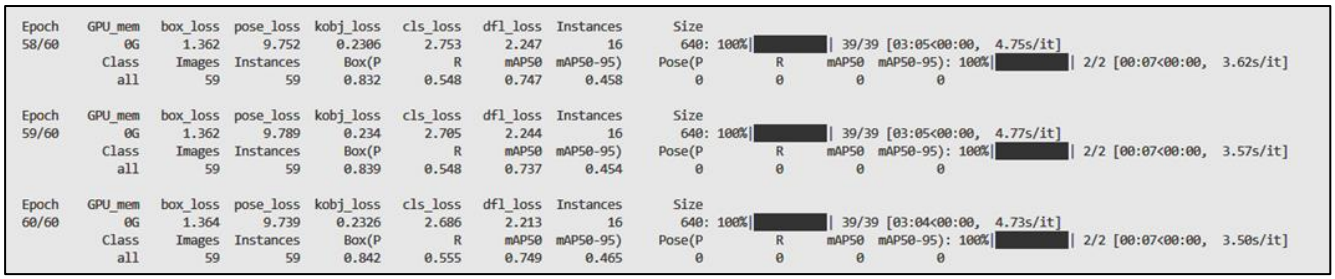

Рисунок 4.9 – Процес навчання моделі по епохам

Під час навчання моделі після кожної епохи виводиться детальна статистика результатів навчання для епохи, для того щоб можна було порівнювати результати та визначити наскільки збільшується результати моделі за кожну ітерацію. Далі буде описано за що відповідає кожен параметр та чому він важливий у контексті розпізнавання жестів нашої мови жестів.

– Box_loss ця функція втрат відповідає за корекцію координат обмежувальних рамок. Менше значення вказує, що модель стає точнішою у визначенні місця об'єктів. Важливість цієї метрики полягає у здатності моделі точно локалізувати жест у кадрі.

– Pose_loss це функція втрат для ключових точок, які використовуються для прогнозування положення руки чи частин тіла. Вона оцінює, наскільки добре модель вміє передбачати координати кожного ключового елемента. У випадку розпізнавання жестів, точність ключових точок є основою для коректного розпізнавання.

– Kobj_loss це функція втрат для об'єктності (objectness), тобто визначення, чи є в певній частині зображення об'єкт. Чим менше це значення, тим краще модель розрізняє корисні області від фону.

– Cls_loss це втрати класифікації, які відповідають за правильне визначення класу об'єкта. У контексті жестів ці втрати оцінюють, наскільки правильно модель визначає, який жест було показано.

– Dfl_loss це специфічна функція втрат для детальної корекції локалізації рамок. Вона дозволяє моделі точніше передбачати положення і форму обмежувальної рамки. Це критично для задач, де потрібно максимально точне визначення позицій.

– Instances це кількість прикладів, які обробляються за одну ітерацію. Збільшення кількості оброблених прикладів може покращити загальну стабільність і точність моделі.

– mAP це основна метрика для оцінки якості моделі. Вона обчислюється на основі точності і повноти для різних значень IoU. mAP50 показує результати для IoU=0.5, а mAP50-95 — середнє значення для кількох порогів IoU. Чим вищі значення mAP, тим краще модель справляється з розпізнаванням.

– Час обробки це важливий параметр, який показує ефективність навчання. Оптимізація часу обробки дозволяє швидше отримувати результати і перевіряти якість моделі.

Поступове зменшення значень втрат і зростання mAP свідчить про те, що модель стає точнішою у прогнозах.

4.3 Результати навчання

Після закінчення ми отримаєм метрику ефективності навчання яка містить деталізовану інформацію про продуктивність моделі для кожного класу жестів окремо. У верхньому рядку наведено загальні метрики для всіх класів, а в наступних рядках — індивідуальні метрики для кожного класу (рис. 4.10).

YOLOv8n-pose summary (fused): 187 layers, 3,591,110 parameters, 0 gradients, 10.5 GFLOPs

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	59	59	0.842	0.555	0.749	0.462
A	1	1	1	0	0	0
Address	2	2	0.65	1	0.995	0.279
B	1	1	1	0	0.497	0.348
Ch	2	2	0.32	0.5	0.662	0.389
D	1	1	1	0	0.497	0.211
E	2	2	1	0	0.498	0.448
F	1	1	1	0	0.995	0.338
Father	2	2	0.461	0.5	0.578	0.323
G	1	1	0.816	1	0.995	0.232
H	2	2	0.84	1	0.995	0.697
I	2	2	1	0.656	0.995	0.647
K	1	1	1	0	0.995	0.697
Kh	2	2	1	0.859	0.995	0.61
L	2	2	0.741	0.5	0.499	0.299
M	1	1	0.698	1	0.995	0.895
N	2	2	0.916	1	0.995	0.696
O	2	2	0.751	0.5	0.745	0.571
P	2	2	0.596	1	0.995	0.552
R	1	1	0.745	1	0.995	0.597

Рисунок 4. 10 – Таблиця результатів розпізнавання жестів в тренувальному наборі даних

У колонці Images показано кількість зображень, використаних для кожного класу під час навчання. Колонка Instances демонструє кількість об’єктів конкретного класу в цих зображеннях.

Метрики Box і R характеризують якість визначення меж об’єктів. Precision показує частку правильно класифікованих об’єктів із тих, що були передбачені моделлю, а Recall відображає, наскільки добре модель знаходить усі можливі об’єкти [42].

mAP50 — середнє значення точності при IoU=50%, яке визначає, наскільки добре модель може локалізувати об’єкти. mAP50-95 розраховує середню точність для кількох порогових значень IoU (від 50% до 95%), що є більш складною і всеохоплюючою метрикою [43].

Параметри, такі як кількість шарів 187, кількість параметрів моделі 3591110 і GFLOPs 10.5, характеризують складність та обчислювальну інтенсивність моделі.

Відсутність градієнтів вказує, що навчання завершено, і модель перебуває у стані інференсу.

Детальна статистика допомагає оцінити якість навчання моделі для кожного жесту та загалом, дозволяючи визначити класи, які потребують додаткової уваги для покращення.

Модель демонструє різну точність для різних жестів. Деякі жести, такі як "H", "I", "K", "Kh", "N", розпізнаються з високою точністю (mAP50 більше 0.6). Інші жести, такі як "A", "Address", "Ch", "Father", "L", розпізнаються гірше (mAP50 менше 0.4).

Загальна точність моделі (mAP50) становить 0.749, що є непоганим результатом, враховуючи власноруч створений датасет який має дуже обмежену кількість зображень. Однак, mAP50-95 становить 0.462, що вказує на зниження точності при більш строгих критеріях перекриття.

Також було отримано числені графіки що описують ефективність моделі по метрикам які мають бути проаналізовані для усвідомлення ефективності моделі.

Графік відображає зміну значення функції втрат для завдання детекції об'єктів під час тренування моделі (рис. 4.11). Функція втрат - це показник того, наскільки добре модель виконує завдання прогнозування.

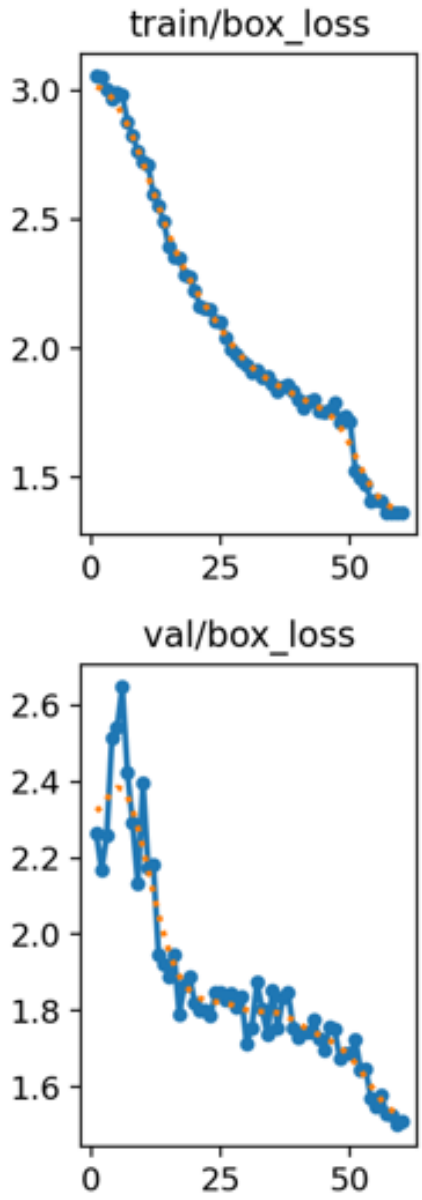

Рисунок 4.11 – Графік функції втрат обмежувальних рамок

Загалом, графік показує, що зі збільшенням кількості епох тренування значення втрат поступово зменшуються. Це свідчить про те, що модель стає точнішою у визначенні об'єктів на зображеннях. Однак після певної кількості епох, приблизно після 50-ї, крива втрат стабілізується, що означає, що модель досягла свого локального мінімуму і подальше тренування не призведе до значного покращення. Необхідно зазначити, що невеликі коливання значення втрат є нормальними, і вони зумовлені характеристиками навчального набору даних.

На наступних графіках (рис. 4.12) зображено зміну значення функції втрат (loss) під час навчання моделі для завдання оцінки пози об'єктів. Функція втрат відображає, як добре під час навчання моделі вона справляється з захопленням пози для оцінки для тренувальної частини датасету та для валідації.

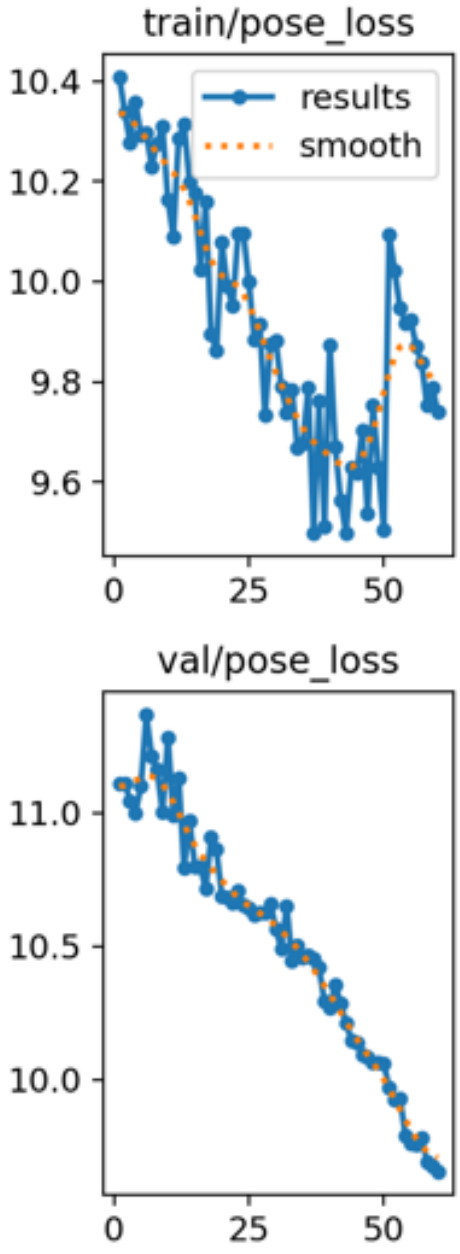

Рисунок 4.12 – Графік функції втрат для пози під час тренування та валідації

Графік показує, що в процесі навчання втрати пози загалом зменшуються, що вказує на поступове покращення моделі. Однак є помітні флуктуації, які можуть свідчити про складність завдання і чутливість моделі до навчальних даних.

Наступними оглянемо графіки що показують зміну значень втрат (loss) під час навчання моделі, яка виконує завдання класифікації (рис. 4.13). Графік kobj_loss відображає точність визначення наявності об’єкта у певній області зображення. Зменшення цього показника вказує на те, що модель стає більш точною в цьому завданні. Графік cls_loss демонструє, як модель класифікує об’єкти. Чим менше значення втрат, тим краще модель визначає категорії об’єктів [44].

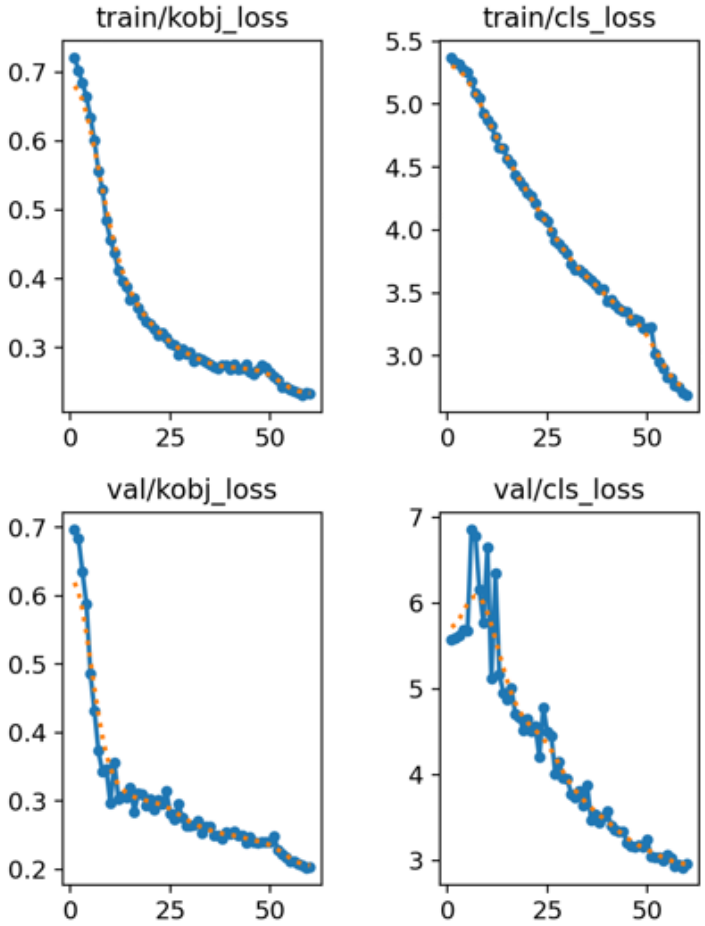

Рисунок 4.13 – Графік функції втрат для знаходження наявності об’єкта під час тренування та валідації

Графіки будуються для двох наборів даних: тренувального (train) і валідаційного (val). Це дозволяє побачити, як модель навчається на вже відомих даних і наскільки добре узагальнює свої знання на нові, не бачені раніше дані.

На графіках видно, що втрати поступово зменшуються, що свідчить про ефективне навчання моделі. Після певної кількості епох значення втрат стабілізується, що означає досягнення оптимізації. Однак у процесі навчання спостерігаються флуктуації, особливо на валідаційних даних, що є нормальним явищем з умовою обмеженого датасету. Це може бути пов’язане з особливостями даних або параметрами навчання. Аналіз графіків показує, що модель навчається добре, але за кривою втрат на валідаційному наборі потрібно слідкувати, щоб уникнути перенавчання [45].

4.4 Тестування

Тепер коли були опрацьовано результати навчання перейдемо до етапу тестування моделі. Для цього реалізуємо систему розпізнавання жестів у реальному часі за допомогою вже навченої нами моделі. Для початку імпортуємо залежності та ініціалізуємо веб-камеру (рис. 4.14)

```python
import cv2
import torch
from ultralytics import YOLO
import time
import numpy as np

model = YOLO("C:/Users/Bohdan/Desktop/mag_work/runs/pose/train2/weights/best.pt")

cap = cv2.VideoCapture(0)

if not cap.isOpened():
    print("Не вдалося відкрити камеру.")
    exit()

frame_count = 0
start_time = time.time()
```

Рисунок 4.14 – Імпорт залежностей та ініціалізація камери для розпізнавання образів

При ініціалізації камери та зображення з неї програма перевіряє, чи вдалося відкрити камеру. Якщо доступ до неї неможливий, програма повідомляє про це і завершується. Також вводяться змінні для підрахунку оброблених кадрів і відстеження часу. Це дає можливість оцінювати продуктивність системи, наприклад, обчислювати кількість кадрів, оброблених за секунду, або аналізувати, як ефективно модель працює у реальному часі. Це підготовчий етап, який забезпечує базову функціональність програми для подальшого розпізнавання жестів.

Тепер опрацюємо захоплення камерою інформації та її налаштування для наступного розпізнавання жестів (рис. 4.15)

```
while True:
    ret, frame = cap.read()

    if not ret:
        print("Не вдалося захопити кадр з камери")
        break

    frame_count += 1

    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Запуск інференсу на кадрі без виведення додаткової інформації
    try:
        results = model(rgb_frame, verbose=False)  # Вимкнення детального виведення
    except Exception as e:
        print(f"Помилка під час інференсу: {e}")
        break

    # Отримання класів і координат для кожного розпізнаного об'єкта
    pred_labels = results[0].names
    pred_classes = results[0].boxes.cls.numpy()
    pred_coords = results[0].keypoints


    print("Розпізнані класи:", pred_classes)
    print("Координати ключових точок:", pred_coords)
```

Рисунок 4.15 – Обробка зображення

На початку, з кожного кадру камери отримується зображення, яке зберігається у змінній. Якщо з якоїсь причини кадр не вдалося отримати (наприклад, камера несправна), цикл припиняється. Це гарантує стабільність роботи програми, навіть якщо виникають проблеми з обладнанням.

Кожен отриманий кадр конвертується з кольорового простору BGR, який використовує OpenCV за замовчуванням, у RGB, оскільки модель нейронної мережі потребує саме такого формату.

Після підготовки кадру виконується інференс, тобто процес, під час якого модель аналізує зображення, шукаючи об’єкти. Використовується спеціальний параметр verbose=False, який вимикає детальне виведення, щоб уникнути зайвих повідомлень під час виконання.

Результати обробки зображення містять передбачені класи об'єктів (тобто типи жестів, якщо вони розпізнані) та ключові точки, які представляють координати частин тіла або руху. Ці дані виводяться для перевірки, дозволяючи оцінити, які саме жести були розпізнані та з якою точністю модель визначила ключові позиції.

Наступним кроком буде візуалізації розпізнавання жестів, виведення результатів на екран та завершення роботи програми (рис. 4.16).

```
    annotated_frame = results[0].plot()


    if len(pred_classes) > 0:
        for cls in pred_classes:

            gesture_name = pred_labels[int(cls)]
            cv2.putText(annotated_frame, f"Gesture: {gesture_name}", (10, 30),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    cv2.imshow("Gesture Recognition", annotated_frame)

    if time.time() - start_time > 10:
        print(f"Виконано {frame_count} кадрів за {int(time.time() - start_time)} секунд.")
        start_time = time.time()  # Скидаємо таймер


    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

ap.release()
v2.destroyAllWindows()
```

Рисунок 4.16 – Візуалізація результатів розпізнавання

Цей блок коду відповідає за візуалізацію та інтерактивність програми. Спочатку створюється annotated_frame — це кадр із зображенням результатів, які модель отримала після обробки. До цього кадру додаються накладені ключові точки та області які визначила модель.

Далі відбувається перевірка, чи є серед результатів розпізнані класи жестів. Якщо класи знайдені, код проходить через кожен із них і отримує відповідну назву

жесту з таблиці класів моделі. Назва жесту виводиться на екрані прямо поверх відеопотоку у визначеній позиції (координати (10, 30)) зеленим кольором, використовуючи шрифт FONT_HERSHEY_SIMPLEX.

Після додавання тексту кадр із розпізнаними жестами відображається у вікні "Gesture Recognition". Таким чином, користувач може в реальному часі бачити і відеопотік, і результат роботи моделі.

Кожні 10 секунд програма виводить кількість кадрів, які вона обробила за цей час. Це використовується для перевірки продуктивності й стабільності. Таймер обнуляється, щоби розпочати відлік заново.

Останній фрагмент забезпечує можливість завершити роботу програми. Якщо користувач натискає клавішу 'q', цикл завершується. Вікно з відеопотоком закривається, і програма коректно завершується.

Тепер коли було описано механізм розпізнавання зображень перейдемо до використання програми для розпізнавання жестів. Що і буде продемонстровано на наступних зображеннях. Першим було перевірено жест що відповідає літері ”H” (рис. 4.17).

Рисунок 4.17 – Візуалізація результатів розпізнавання жесту

На цьому зображенні модель продемонструвала результат розпізнавання жесту руки ”H”. Рука обведена фіолетовою рамкою, яка позначає область, де модель виявила об'єкт. Всередині цієї рамки модель визначила ключові точки на руці. Ці точки, відмічені різними кольорами, відповідають важливим частинам руки, таким як кінчики пальців, суглоби та центр долоні. Ї

У верхньому куті рамки вказано клас об'єкта разом із значенням ймовірності, яке дорівнює 0.7. Це число свідчить про те, що модель з упевненістю у 70% визначила виявлений об'єкт як певний жест. Хоча це досить високе значення, але як ми можемо бачити система розпізнавання розмістила багато точок в хибному місці через те що втрата пози є слабким місцем системи так як і необхідність жорсткого зберігання правильного положення руки.

Наступним жестом що буде проходити перевірку буде жест що відповідає літері ”П” (рис. 4.18).

Рисунок 4.18 – Візуалізація результатів розпізнавання жесту ”П”

Під час розпізнавання цього жесту видно що впевненість в розпізнавання жесту становить 80 відсотків, але видно що під час розпізнавання було хибно виставлено частину точок, а одна взагалі виходить за межі поля розпізнавання. Також протестуємо розпізнавання жесту в умові зміни позиції пальців через що системі потрібно буде по іншому визначати ключові точки для розпізнавання (рис. 4.19).

Рисунок 4.19 – Візуалізація результатів розпізнавання жесту ”П” в іншій позиції

Наступним жестом що буде проходити перевірку буде жест що відповідає літері ”Т” (рис. 4.20).

Рисунок 4.20 – Візуалізація результатів розпізнавання жесту ”T”

Як ми бачимо під час цього розпізнавання цього жесту були труднощі через його подібність до іншого жесту і через це впевненість у розпізнаванні лише 50 відсотків.

Також для перевірки розглянемо більш складні жести для розпізнавання такий як жест “O” з різними позиціями пальців (рис. 4.21-4.22)

Рисунок 4.21 – Візуалізація результатів розпізнавання жесту ”O”

Як ми бачимо через складність жесту для визначення ключових точок рівень розпізнавання складає 50 відсотків, що не є поганих результатом, але значно поступається жестам з більш простими позами.

Рисунок 4.22 – Візуалізація результатів розпізнавання жесту ”O” зі зміною пози

Під час зміни пози відбулось значне зменшення розпізнавання що говорить нам про те що в датасеті недостатньо зображень для навчання з значно зміненим положенням пальців для одного жесту.

Висновки до розділу 4

У цьому розділі було розглянуто розробку та реалізацію програми для розпізнавання жестів у реальному часі, що базується на використанні моделі YOLOv8. Детальний аналіз включав три ключові етапи: створення коду програми, навчання моделі на підготовленому наборі даних, а також тестування її роботи в режимі реального часу.

Етап навчання моделі включав аналіз графіків втрат (loss) та оцінку якості моделі. Спостерігалося зменшення значення втрат на тренувальному і валідаційному наборах, що свідчить про успішне навчання. У стабільний момент навчання модель досягла прийнятного рівня узагальнення на нових даних, але також були помітні невеликі флуктуації, що є нормальним для складних задач.

Під час тестування в реальному часі модель успішно виявляла жести, що підтверджувалося коректним візуальним представленням рамок і ключових точок на зображенні. Програма демонструвала стабільну роботу з хорошою частотою кадрів, але якість результатів залежала від зовнішніх факторів, таких як освітлення та фон.

Таким чином, створена система розпізнавання жестів у реальному часі довела свою ефективність, показавши високий потенціал для використання у завданнях аналізу жестової мови, комп’ютерного зору та взаємодії з користувачем. Однак залишаються можливості для покращення точності моделі та оптимізації її роботи, зокрема шляхом збільшення навчального набору даних або покращення моделі навчання та збільшення кількості епох.

ВИСНОВКИ

В результаті проведеного дослідження була вирішена комплексна задача, спрямована на розробку системи розпізнавання жестової мови для підтримки української жестової мови (УЖМ). Робота охопила всі ключові етапи створення технології: від аналізу особливостей жестової мови до реалізації та тестування прототипу в реальному часі.

Першим важливим кроком було виявлення проблем, пов’язаних із дослідженням і використанням української мови жестів. Було встановлено, що жестова мова має значний лінгвістичний і соціальний потенціал, але її автоматизація залишається недостатньо розвинутою через брак спеціалізованих інструментів та обмеженість доступних наборів даних. Аналіз наявних рішень і технологій у цій галузі виявив необхідність створення локалізованої моделі, яка враховує специфіку УЖМ, зокрема її дактильну абетку та унікальні жести.

Другим важливим етапом стало створення датасету для навчання моделі. Було розроблено методику збору й анотування зображень, яка включала залучення жестів із різних ракурсів та умов освітлення. У ході роботи був створений збалансований і якісно підготовлений набір даних, який охоплює основні жести, включаючи букви дактильної абетки. Це стало основою для подальшого навчання системи.

На етапі розробки системи використовувалися сучасні підходи до аналізу зображень і глибокого навчання. Основою моделі став алгоритм YOLOv8, який був адаптований для завдань розпізнавання жестів та пози ключових точок рук. Навчання моделі на підготовленому датасеті дало змогу досягти високої точності в ідентифікації жестів, а також їх класифікації в реальному часі. Аналіз графіків втрат і метрик моделі підтвердив ефективність навчання, хоча були помічені потенційні напрямки для подальшого вдосконалення, такі як покращення узагальнення моделі.

Фінальним етапом стало тестування системи в реальних умовах. Розроблена програма дозволяє зчитувати жести з відеопотоку, обробляти кадри в реальному часі й візуалізувати результати. Система успішно продемонструвала здатність розпізнавати жести, що підтвердило практичну цінність розробленого рішення. Проте тестування також виявило залежність результатів від умов освітлення та фону, що вказує на потребу в подальшій оптимізації.

Таким чином, виконана робота зробила вагомий внесок у розв’язання проблеми автоматизації української жестової мови. Було створено повноцінну систему розпізнавання жестів, яка може слугувати основою для майбутніх розробок, таких як інтерактивні інтерфейси, освітні платформи чи інклюзивні рішення для осіб із порушеннями слуху. Подальший розвиток цього проєкту може включати масштабування датасету, інтеграцію віртуальних помічників та покращення точності й продуктивності моделі, що дозволить забезпечити ще ширшу підтримку користувачів УЖМ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Адамюк Н. Б. Синтаксичні особливості УЖМ: на прикладі простого речення. жестова мова й сучасність. 4-те вид. 250 с.

2. Arabnia, H. R., Deligiannidis, L., & Tinetti, F. G. Image processing, computer vision, and pattern recognition. C.S.R.E.A., 2020. (168 p.)

3. Papastratis, I., et al. Artificial intelligence technologies for sign language. Sensors, 21(17), 5843, 2021. https://doi.org/10.3390/s21175843 (дата звернення: 06.12.2024).

4. Cooper, H., & Bowden, R. Sign language recognition systems: Towards large-scale deployment. 2015. (400 p.)

5. Lee, H. J. (Ed.). Deep learning-based action recognition. MDPI, 2022. https://doi.org/10.3390/books978-3-0365-5200-2 (дата звернення: 06.12.2024).

6. Toomsen, S. Sign language: American sign language alphabet for beginners: American sign language. Independently Published, 2021. (P. 112)

7. Woll, B., Steinbach, M., & Pfau, R. Sign language. De Gruyter, Inc., 2012. (1140 p.)

8. Goodfellow, I., Bengio, Y., & Courville, A. Deep learning. MIT Press, 2016. (P. 345)

9. Chollet, F. Deep learning with Python. Manning Publications Co., 2017. (P. 210)

10. Aggarwal, C. C. Neural networks and deep learning. Springer, 2018. (P. 450)

11. Rosebrock, A. Deep learning for computer vision with Python. PyImageSearch, 2019. (P. 580)

12. Géron, A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019. (P. 670)

13. Flach, P. Machine learning: The art and science of algorithms that make sense of data. Cambridge University Press, 2012. (P. 390)

14. Bishop, C. M. Pattern recognition and machine learning. Springer, 2006. (P. 730)

15. Sutton, R. S., & Barto, A. G. Reinforcement learning: An introduction. MIT Press, 2018. (P. 285)

16. Haykin, S. S. Neural networks and learning machines. Pearson Education, 2009. (P. 890)

17. Nielsen, M. A. Neural networks and deep learning. Determination Press, 2015. (P. 150)

18. Russell, S. J., & Norvig, P. Artificial intelligence: A modern approach. Pearson Education Limited, 2020. (P. 1020)

19. Murphy, K. P. Machine learning: A probabilistic perspective. MIT Press, 2012. (P. 1100)

20. Gulli, A., & Kapoor, A. Deep learning with Keras. Packt Publishing Ltd, 2017. (P. 310)

21. Stevens, E., Antiga, L., & Viehmann, T. Deep learning with PyTorch. Manning Publications Co., 2020. (P. 460)

22. Bird, S., Klein, E., & Loper, E. Natural language processing with Python. O'Reilly Media, Inc., 2009. (P. 500)

23. Szeliski, R. Computer vision: Algorithms and applications. Springer Science & Business Media, 2010. (P. 810)

24. VanderPlas, J. Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc., 2016. (P. 620)

25. Goodfellow, I. NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160, 2017. (P. 15)

26. Lecun, Y., Bengio, Y., & Hinton, G. Deep learning. Nature, 521(7553), 436-444, 2015. (P. 444)

27. He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778), 2016. (P. 778)

28. Bishop, C. M. Pattern Recognition and Machine Learning. Springer, 2006. (P. 730)

29. Duda, R. O., Hart, P. E., & Stork, D. G. Pattern classification. John Wiley & Sons, 2012. (P. 680)

30. Szeliski, R. Computer Vision: Algorithms and Applications. Springer Science & Business Media, 2010. (P. 810)

31. Bradski, G., & Kaehler, A. Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc., 2008. (P. 570)

32. Gonzalez, R. C., & Woods, R. E. Digital image processing. Pearson Education, 2017. (P. 920)

33. Krohn, J., Beyleveld, G., & Bassens, A. Deep learning illustrated: A visual, interactive guide to artificial intelligence. 2023. (P. 480)

34. Trask, A. W. Grokking deep learning. 2019. (P. 350)

35. Kinsley, H., & Kukieła, D. Neural networks from scratch in Python. 2021. (P. 290)

36. Burkov, A. The hundred-page machine learning book. 2019. (P. 100)

37. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. Dive into deep learning. 2021. (P. 830)

38. Boehmke, B., & Greenwell, B. Hands-on machine learning with R. 2019. (650 p.)

39. Lapan, M. Deep reinforcement learning hands-on. 2020. (550 p.)

40. Ng, A. Machine learning yearning. 2018. (240 p.)

41. Molnar, C. Interpretable machine learning. 2020. (420 p.)

42. Ameisen, E. Building machine learning powered applications. 2023. (380 p.)

43. Lakshmanan, V., Robinson, S., & Munn, M. Machine learning design patterns. 2020. (350 p.)

44. Raschka, S., & Mirjalili, V. Python machine learning. 2019. (800 p.)

45. Hastie, T., Tibshirani, R., & Friedman, J. The elements of statistical learning. 2009. (760 p.)

ДОДАТОК А

Матеріали апробації роботи

Робота пройшла апробацію під час проведення Всеукраїнської науково-практичної конференції молодих вчених, аспірантів і студентів «Інтелектуальні інформаційні системи» (Миколаїв, 2-4 грудня, 2024 р.).

Міністерство освіти і науки України
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Інформаційний лист

Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів

Інтелектуальні інформаційні системи

2 – 4 грудня 2024 року

Миколаїв

ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

03.12.2024 р.
15-30
Посилання: https://meet.google.com/omd-omzx-cxz

Голови секції: к.т.н., доц. Сіденко Є.В.,
д.т.н., проф. Козлов О.В.

Секретар секції: Димо В.

Somriakov B. Logistics and Supply Chain Optimization Software for Ukraine's National Infrastructure.

Мельничук М. С. Інтелектуальна система моделювання та прогнозування на основі методів комбінування.

Валюшок Б. І. Інтелектуальна система для розпізнавання жестової мови з використанням нейроної мережі.

УДК 004.42

Валюшок Б. І.
студент
Чорноморський національний університет
імені Петра Могили
м. Миколаїв, Україна

Інтелектуальна система для розпізнавання жестової мови з використанням нейроної мережі

В Україні, за даними досліджень міністерства соціальної політики за 2021 рік, близько 40 тисяч людей використовують жестову мову як основний засіб комунікації, а кількість людей що постійно використовують жестову мову сягає 200 тисяч[1]. Ця цифра охоплює як глухих та слабочуючих людей, так і тих, хто взаємодіє з ними в повсякденному житті, зокрема членів їхніх родин. Однак через триваючу війну та бойові дії ця кількість має тенденцію до щорічного зростання(рис. 1).

Військові дії мають значний вплив на здоров'я солдатів та цивільного населення, зокрема спричиняючи втрату слуху внаслідок контузій, вибухів та інших травматичних факторів. Сучасна статистика свідчить, що акустичні травми, пов'язані з перебуванням у зоні активних бойових дій, є однією з найпоширеніших проблем серед військових. Це означає, що після завершення служби значна частина ветеранів матиме потребу в адаптації до нових умов життя, зокрема у використанні жестової мови.