Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра інформатики

Курсова робота

освітній ступінь – бакалавр

на тему: «**Знаходження об’єктів у відеопотоці**»

Виконав: студент 3-го року навчання,

Спеціальності 121 «Інженерія Програмного Забезпечення»

Снігір Андрій

Керівник Бучко Олена Андріївна

Старший викладач, кандидат технічних наук

Київ – 2023

Тема: Знаходження об’єктів у відеопотоці

Календарний план виконання роботи:

№	Назва етапу курсової роботи	Термін виконання	Примітка
1.	Отримання завдання на курсову роботу	листопад	
2.	Огляд літератури за темою роботи	листопад-грудень	
3.	Створення практичної частини роботи	січень	
4.	Написання текстової частини роботи	лютий-квітень	
6.	Надання роботи керівнику для перевірки	кінець квітня	
8.	Коригування роботи за результатами перевірки	кінець квітня-початок травня	
9	Подання роботи на кафедру для перевірки на плагіат	тиждень до захисту	
10.	Захист курсової роботи	кінець травня	

Студент Снігір А. Б., Керівник Бучко О. А. “____” _____

1. Зміст

ВСТУП

1.1. Актуальність теми.

У сучасному світі, де комп'ютерний зір і штучний інтелект посідають ключові позиції у розвитку технологій та індустрії, знаходження об'єктів у відеопотоці стає важливою складовою ефективних систем спостереження, контролю та моніторингу.

Актуальність знаходження об'єктів у відеопотоці обумовлена рядом факторів, які вказують на важливість дослідження цієї проблеми.

- Безпека та моніторинг: Системи відеоспостереження активно використовуються для забезпечення безпеки на промислових об'єктах, у сфері транспорту, в торгових центрах та урбаністичному середовищі. Автоматичне знаходження об'єктів у відеопотоці дає змогу оперативно реагувати на події, що відбуваються, та забезпечує ефективний контроль та аналіз ситуації.
- Розвиток технологій комп'ютерного зору: В останні роки спостерігається стрімкий розвиток технологій комп'ютерного зору та штучного інтелекту, що призводить до появи нових методів та алгоритмів для знаходження об'єктів у відеопотоці. Вивчення актуальних методів сприяє підвищенню конкурентоспроможності на ринку, оскільки дозволяє орієнтуватися в продуктах та сервісах з високим рівнем точності та продуктивності.
- Широке застосування в різних галузях: Технології знаходження об'єктів у відеопотоці можуть бути використані в різних галузях, таких як реклама, маркетинг, медицина, робототехніка та автоматизація виробництва. Вони дозволяють оптимізувати процеси, забезпечити більш точне розпізнавання об'єктів та підвищити ефективність роботи в цілому.

1.2. Мета і завдання дослідження.

Основною метою даної курсової роботи є дослідження методів знаходження об'єктів у відеопотоці з використанням нейронних мереж. Мета полягає в розумінні теоретичних основ знаходження об'єктів у відео, вивченні нейронних мереж та їх застосуванні для розпізнавання об'єктів, а також у розробці програми, що використовує натреновану нейронну мережу для знаходження об'єктів у відеопотоці.

Для досягнення вищезазначеної мети, передбачається вирішення наступних завдань:

1. Вивчити та проаналізувати теоретичні основи знаходження об'єктів у відеопотоці. Дослідити основні поняття та термінологію, пов'язані з розпізнаванням об'єктів у відео, включаючи методи та підходи до цього процесу.
2. Ознайомитись з різними методами та підходами до розпізнавання об'єктів у відео. Розглянути різноманітні алгоритми та підходи, включаючи традиційні методи комп'ютерного зору, машинного навчання та глибоке навчання з використанням нейронних мереж.
3. Оглянути сучасні алгоритми та інструменти, що використовуються для знаходження об'єктів у відеопотоці. Дослідити різноманітні алгоритми та популярні відкриті бібліотеки та фреймворки, що використовуються для розробки програм для розпізнавання об’єктів у відеопотоці.
4. Використати натреновану нейронну мережу для створення програми знаходження об'єктів у відеопотоці.

2. Теоретичні основи знаходження об'єктів у відеопотоці.

2.1. Основні поняття та термінологія.

У цьому підрозділі хотів би зазначити основні поняття, пов’язані зі знаходженням об’єктів у відеопотоці. Не всіх з наступних термінів буде безпосередньо торкатися моя курсова робота, але вважаю, що їх згадка була б доцільна.

- Комп'ютерний зір: Галузь науки, що займається аналізом та інтерпретацією зображень і відео за допомогою алгоритмів та програмного забезпечення з метою розпізнавання, класифікації та відслідковування об'єктів.
- Відеопотік: Послідовність зображень, які зазвичай зчитуються з відеокамери або зберігаються у відеофайлі. Відеопотік може бути в режимі реального часу або відтворюватися з запасу.
- Об'єкт: Сутність, яка може бути визначена та розпізнана на зображенні або відео. Об'єкти можуть бути різного типу, наприклад: люди, транспортні засоби, тварини, предмети тощо.
- Розпізнавання об'єктів: Процес виявлення, класифікації та відслідковування об'єктів на зображеннях або відеопотоці за допомогою алгоритмів комп'ютерного зору та/або штучного інтелекту.
- Нейронна мережа: Математична модель, яка навчається з даних та імітує роботу людського мозку. Нейронні мережі здатні виконувати складні задачі розпізнавання образів, включаючи знаходження об'єктів у відеопотоці.
- Глибоке навчання (Deep Learning): Розділ машинного навчання, що базується на нейронних мережах з великою кількістю шарів. Глибоке навчання використовується для вирішення складних задач

розпізнавання образів, в тому числі знаходження об'єктів у відеопотоці.

- Згорткова нейронна мережа (Convolutional Neural Network, CNN): Тип нейронної мережі, особливо ефективний для аналізу зображень і відео. Вони містять згорткові шари, які виявляють локальні особливості зображень, що допомагає забезпечити інваріантність до масштабу, перекручення та інших видів деформацій.
- Відслідковування об'єктів: Процес визначення траєкторії об'єкта у відеопотоці шляхом аналізу зміни його положення від кадру до кадру.
- База даних для навчання: Набір даних, який використовується для навчання нейронної мережі. Він містить зображення або відео з різними об'єктами та їх мітками, які вказують на тип об'єкта, його розташування та інші атрибути.
- Аугментація даних: Техніка збільшення кількості навчальних даних шляхом створення нових зразків з використанням вихідних даних за допомогою різних трансформацій, таких як обертання, масштабування, відображення тощо.
- Функція втрат або Loss function: Математична функція, яка вимірює різницю між передбаченнями нейронної мережі та дійсними мітками. Функція втрат використовується під час навчання для оцінки ефективності моделі та коригування її ваг.
- Оптимізатор: Алгоритм, що використовується для мінімізації значення функції втрат шляхом оновлення ваг нейронної мережі. Оптимізатори, такі як стохастичний градієнтний спуск (SGD), Адам (Adam) та інші, допомагають забезпечити ефективне навчання та зменшення помилок моделі.

- Валідація та тестування: Процес перевірки ефективності нейронної мережі на незалежних наборах даних після навчання. Валідація використовується для налаштування гіперпараметрів моделі, в той час як тестування допомагає оцінити загальну ефективність моделі та її здатність узагальнювати результати на нових даних.
- Гіперпараметри: Параметри, які задаються перед початком навчання нейронної мережі і не змінюються під час навчання. Гіперпараметри включають кількість шарів та нейронів у мережі, розмір партії (batch size), крок навчання (learning rate) та інші параметри, що впливають на процес навчання та ефективність моделі.
- Transfer learning: Техніка, яка передбачає використання вже навченої нейронної мережі як вихідної точки для навчання нової моделі. Transfer learning дозволяє скоротити час навчання та використовувати накопичений досвід для розв'язання схожих задач з розпізнавання об'єктів у відеопотоці.

2.2. Методи та підходи до розпізнавання об'єктів в відео.

Розпізнавання об'єктів у відео є важливою задачею в галузі комп'ютерного зору, і для її вирішення розроблено ряд методів та підходів. Нижче я навів огляд основних методів розпізнавання об'єктів у відеопотоці:

- Методи засновані на класичному комп'ютерному зорі, наприклад, такий як виявлення об'єктів на основі особливостей: використовування алгоритмів, таких як SIFT, SURF, ORB та інші, для виявлення особливостей, щоб виявити та співставити об'єкти на різних зображеннях або відео.
- Глибокі нейронні мережі для виявлення об'єктів, наприклад, згорткові нейронні мережі (CNN): Використовують CNN для автоматичного виявлення та класифікації об'єктів у відеопотоці, використовуючи навчання з учителем та великі набори даних.
- Алгоритми відслідковування на основі кореляції: Використовують алгоритми, такі як MeanShift, CAMShift та інші, для стеження за об'єктами у відеопотоці, враховуючи зміни їх положення, масштабу та орієнтації.
- Методи семантичної сегментації, наприклад, за допомогою мереж на основі згорток та декодування (Encoder-Decoder Networks): Застосовують глибокі архітектури згорткових мереж для виявлення особливостей об'єктів на різних масштабах, а потім відновлюють просторову інформацію з використанням декодуючих шарів, що дозволяє точно сегментувати об'єкти на зображеннях або відео.
- Методи 3D-розпізнавання як об'ємна реконструкція: Використовують стерео зображення, глибинні карти або інші джерела 3D-інформації для реконструкції об'ємних моделей об'єктів, що можуть полегшити їх розпізнавання та відслідковування.

З урахуванням швидкого розвитку технологій машинного та глибокого навчання, методи розпізнавання об'єктів у відео продовжують покращуватися, що дає можливість створювати все більш ефективні та надійні системи для автоматичного аналізу відеоданих.

2.3. Огляд сучасних алгоритмів та інструментів.

Сучасні алгоритми та інструменти для знаходження об'єктів у відеопотоці базуються на різних техніках комп'ютерного зору та машинного навчання. Розглянемо деякі ключові алгоритми та інструменти, які використовуються в цій області:

- YOLO (You Only Look Once): Швидкий алгоритм виявлення об'єктів у реальному часі, який використовує єдину згорткову нейронну мережу для одночасного виявлення та класифікації об'єктів на зображенні.
- Faster R-CNN: Цей алгоритм комбінує згорткові нейронні мережі (CNN) та мережі з пропозиціями регіонів (RPN) для швидкого та точного виявлення об'єктів на зображенні.
- SSD (Single Shot MultiBox Detector): Цей метод також використовує згорткову нейронну мережу для виявлення та класифікації об'єктів. Він може працювати швидше, ніж Faster R-CNN, з незначним зниженням точності.
- Mask R-CNN: Розширення Faster R-CNN, яке додає гілку для передбачення масок об'єктів, дозволяючи одночасно виконувати сегментацію та виявлення об'єктів.
- OpenCV (Open Source Computer Vision Library): Бібліотека комп'ютерного зору, яка містить імплементації різних алгоритмів знаходження об'єктів, таких як Haar Cascade, SIFT, SURF та інші.

Ці п'ять методів широко використовуються у дослідженнях та практичних застосуваннях знаходження об'єктів у відеопотоці. Вони відрізняються своїми архітектурами, точністю та швидкістю, і вибір підходящого методу залежить від вимог до конкретної задачі.

3. Нейронні мережі для знаходження об'єктів у відеопотоці.

3.1. Основні принципи нейронних мереж.

Нейронні мережі є обчислювальними моделями, навченими виконувати завдання на основі навчальних даних. Вони створені з метою імітації способу, яким людський мозок обробляє інформацію. Основою нейронних мереж є штучні нейрони, які імітують біологічні нейрони в мозку. Ці штучні нейрони отримують вхідні дані, обробляють їх за допомогою ваг та генерують вихідне значення.

Архітектура нейронних мереж складається з взаємопов'язаних шарів штучних нейронів. Найпростіша архітектура включає вхідний, прихований та вихідний шари. Складніші архітектури можуть мати кілька прихованих шарів, які допомагають навчитися глибоким патернам у даних.

Важливим компонентом нейронних мереж є зважені з'єднання між нейронами. Ваги представляють силу зв'язків і оптимізуються в процесі навчання для покращення точності мережі.

Ще одним важливим аспектом нейронних мереж є функція активації, яка використовується в нейроні для обробки сумарного входу, який отримується від попередніх нейронів. Функція активації може бути лінійною або не лінійною, залежно від вимог до завдання та архітектури мережі.

Завдяки цим основним принципам нейронні мережі здатні навчатися виконувати різноманітні завдання, такі як класифікація, регресія, сегментація зображень, виявлення об'єктів та багато інших.

3.2. Архітектури нейронних мереж для обробки зображень та відео.

Архітектури нейронних мереж для обробки зображень та відео можуть бути досить різноманітними, оскільки вони мають бути адаптовані до конкретних завдань. Однак існує декілька загальних категорій архітектур, які підходять для обробки зображень та відео.

Згорткові нейронні мережі (CNN) є однією з найпопулярніших архітектур для обробки зображень. Вони використовують згорткові шари для виявлення локальних особливостей зображення. Згорткові шари фільтрують зображення за допомогою наборів фільтрів, які "згортають" зображення з метою виявлення різних властивостей, таких як кольори, текстури та контури. Згорткові мережі часто містять пулінг-шари, які зменшують просторові розміри зображення та сприяють узагальненню.

Для обробки відео архітектури нейронних мереж можуть бути розширені за допомогою рекурентних нейронних мереж (RNN) або 3D згорток. Рекурентні нейронні мережі, такі як LSTM (Long Short-Term Memory) або GRU (Gated Recurrent Unit), можуть обробляти послідовності кадрів у відео для моделювання часових залежностей. Згорткові LSTM-мережі, наприклад, можуть використовуватися для передбачення наступних кадрів відео або аналізу відео.

3D згорткові нейронні мережі є ще одним підходом до обробки відео. Вони використовують 3D згортки, які одночасно проходять через просторові та часові вісі, що дозволяє моделювати і просторові, і часові властивості відео.

Комбінації різних архітектур, таких як згорткові, рекурентні та 3D згорткові мережі, можуть створювати гнучкі та потужні моделі для обробки зображень та відео. Ці архітектури можуть бути адаптовані до

широкого спектру завдань, включаючи класифікацію, сегментацію, виявлення об'єктів, розпізнавання дій та інше.

Автоенкодери та генеративно-змагальні мережі (GAN) є іншими видами архітектур, які можуть бути використані для обробки зображень та відео. Автоенкодери складаються з двох частин: енкодера, який стискає вхідне зображення в компактне кодування, та декодера, який відтворює зображення з цього кодування. Вони можуть бути використані для завдань, таких як стиснення зображень, відтворення та доповнення зображень.

Генеративно-змагальні мережі (GAN) мають дві частини: генератор, який створює нові зображення або кадри відео, та дискримінатор, який намагається відрізнити справжні зображення від тих, що створені генератором. GAN можуть генерувати високоякісні зображення та відео, а також знаходити застосування у завданнях, таких як перенос стилю, супер-розширення зображень, синтез зображень та інше.

У процесі вибору архітектури нейронної мережі для обробки зображень та відео, розробники повинні враховувати специфіку завдання, наявність даних, а також обчислювальні можливості, доступні для навчання та виконання моделі.

3.3. Процес тренування нейронних мереж для розпізнавання об'єктів.

Процес тренування нейронних мереж для розпізнавання об'єктів може бути розбитий на ключові етапи. У цьому підрозділі докладно розглянемо їх.

1. Збір даних: Першим кроком є збір набору даних зі зображеннями або відео, на яких представлені різні об'єкти, які необхідно розпізнати. Набір даних повинен бути репрезентативним та містити достатню кількість прикладів для кожного класу об'єктів.
2. Анотація даних: Для кожного зображення чи кадру відео анотатори мають позначити класи об'єктів та їх розташування на зображенні, зазвичай у вигляді прямокутників обрамлення (bounding boxes) або масок сегментації. Ця інформація буде використана як "істина" під час тренування мережі.
3. Розбиття даних: Зібрані та анотовані дані розбиваються на навчальний, валідаційний та тестовий набори. Навчальний набір використовується для тренування моделі, валідаційний - для налаштування гіперпараметрів та перевірки переобучення, а тестовий - для оцінювання загальної ефективності моделі.
4. Попередня обробка даних: Перед тренуванням моделі дані можуть бути піддані попередній обробці для покращення результатів та прискорення тренування. Це може включати масштабування зображень, аугментацію даних (повороти, зсуви, зміна контрастності та інше) та нормалізацію пікселів.
5. Вибір архітектури: В залежності від завдання та наявних даних, можна вибрати підходящу архітектуру нейронної мережі, таку як згорткова нейронна мережа (CNN), рекурентна нейронна мережа (RNN) або трансформери. Для задач розпізнавання об'єктів зазвичай

використовуються згорткові нейронні мережі, такі як Faster R-CNN, YOLO або SSD, через їх здатність ефективно обробляти зображення.

6. Ініціалізація моделі: Перед початком тренування модель ініціалізується випадковими вагами або вагами з передтренованої моделі. Передтреновані моделі можуть прискорити процес навчання та покращити результати, особливо якщо ваш набір даних є відносно невеликим.
7. Тренування моделі: Модель тренується на навчальному наборі даних, використовуючи алгоритм оптимізації (наприклад, стохастичний градієнтний спуск) для мінімізації функції втрат. Протягом тренування модель намагається передбачити мітки об'єктів на зображеннях та порівнює свої передбачення з анотованими мітками. Вона постійно оновлює свої ваги, щоб зменшити помилки передбачення.
8. Налаштування гіперпараметрів: Протягом тренування моделі необхідно відрегулювати гіперпараметри, такі як швидкість навчання, розмір партії та кількість епох. Це може бути зроблено за допомогою валідаційного набору даних, щоб забезпечити оптимальні результати та уникнути переобучення.
9. Оцінка моделі: Після закінчення тренування модель оцінюється на тестовому наборі даних, щоб визначити її загальну ефективність у розпізнаванні об'єктів. Це включає в себе використання метрик, таких як точність (precision), повнота (recall), F1-міра та метрика Intersection over Union (IoU). Оцінка моделі на тестовому наборі даних допомагає зрозуміти, наскільки добре модель узагальнюється на нових даних, та виявити можливі проблеми, такі як переобучення або недообучення.
10. Вивчення результатів та ітерації: Після оцінки моделі можна проаналізувати її результати, виявити слабкі місця та зробити

висновки щодо подальшого вдосконалення. Це може включати зміну архітектури мережі, додавання регуляризації, збільшення розміру набору даних або використання інших технік для підвищення ефективності моделі.

11. Реалізація та впровадження: Останнім кроком є інтеграція натренованої моделі у реальний застосунок або систему для розпізнавання об'єктів у відеопотоці. Це може включати оптимізацію моделі для прискорення часу обробки, адаптацію моделі для роботи на різних пристроях (наприклад, мобільних пристроях або вбудованих системах) та інтеграцію з іншими компонентами системи, такими як бази даних, інтерфейси користувача та системи сповіщення.

Усі ці кроки можуть бути повторені та оптимізовані в процесі розробки та впровадження системи розпізнавання об'єктів, щоб досягти найкращих можливих результатів та відповідати вимогам конкретного застосування або викликів проблеми.

4. Використання натренованої нейронної мережі для знаходження об'єктів у відеопотоці.

4.1. Вибір та аналіз натренованої моделі.

Ретельно проаналізувавши всі доступні натреновані моделі, для своєї курсової роботи я вирішив зупинитися на Google Teachable Machine.

Google Teachable Machine - це інтерактивний інструмент від Google, який дозволяє користувачам створювати і навчати моделі машинного навчання безпосередньо у веб-браузері без необхідності написання коду.

Teachable Machine використовує передавання навчання (transfer learning) для швидкого навчання моделей на основі користувацьких даних.

Передавання навчання полягає в тому, що замість навчання моделі з нуля на великому наборі даних, ми використовуємо вже натреновану модель (наприклад, MobileNet або Inception) на великому наборі даних (зазвичай ImageNet). Ці базові моделі вже навчені розпізнавати загальні особливості зображень, такі як кольори, текстури, форми тощо.

Ось як Google Teachable Machine використовує передавання навчання:

1. Вибір базової моделі: Teachable Machine вибирає натреновану базову модель, таку як MobileNet або Inception. Ці моделі вже навчені розпізнавати загальні особливості зображень на великому наборі даних ImageNet.
2. Відокремлення останнього шару: Останній шар базової моделі, який відповідає за класифікацію, відокремлюється. Замість нього додається новий шар або декілька шарів, які відповідають за класифікацію користувацьких даних.
3. Навчання на користувацьких даних: Teachable Machine навчає тільки новий шар (або шари) на користувацьких даних. Це означає, що ми

адаптуємо базову модель до нашої конкретної задачі, зберігаючи вже навчені особливості зображень.

4. Експортування моделі: Після успішного навчання моделі на користувацьких даних, Teachable Machine дозволяє експортувати модель для подальшого використання у власних проектах або використовувати її безпосередньо через веб-інтерфейс або API TensorFlow.js.

Використання передавання навчання в Teachable Machine дозволяє користувачам створювати досить точні моделі з невеликими наборами даних та значно скорочувати час навчання моделі. Оскільки більшість загальних особливостей зображень вже вивчена базовою моделлю, нашій адаптованій моделі потрібно тільки "відточити" ці особливості для нашої конкретної задачі.

Крім того, передавання навчання дозволяє нам використовувати натреновані моделі на різних пристроях, таких як смартфони або веб-браузери, з меншими обмеженнями на обчислювальні ресурси. Це робить Teachable Machine особливо привабливою для розробників, дизайнерів та художників, які хочуть легко створювати власні машинні моделі навчання без необхідності глибокого розуміння машинного навчання чи доступу до великих обчислювальних ресурсів.

4.2. Опис теми програми та аналіз її побудови.

Програма, представлена мною в рамках курсової роботи, демонструє застосунок TensorFlow та OpenCV для розпізнавання об'єктів на відео у режимі реального часу. В моєму випадку, програма визначає чи вдягнена людиною медична маска. Під час пандемії COVID-19, системи розпізнавання об'єктів у відеопотоці стали актуальними і корисними інструментами для багатьох застосунків, особливо для визначення того, чи носять люди медичні маски. Враховуючи важливість дотримання протоколів медичної безпеки, такі програми мали значний вплив на забезпечення безпеки громадських місць, таких як магазини, ресторани, аеропорти та офісні приміщення.

Я почав роботу над програмою зі створення моделі Google Teachable Machine. Як я вже зазначив у минулому підрозділі, Teachable Machine дозволяє доволі швидко та зручно створити модель для потреб користувача завдяки використанню вже натренованої базової моделі.

Спочатку, користувачу пропонується вибрати тип проекту (Рис. 1).

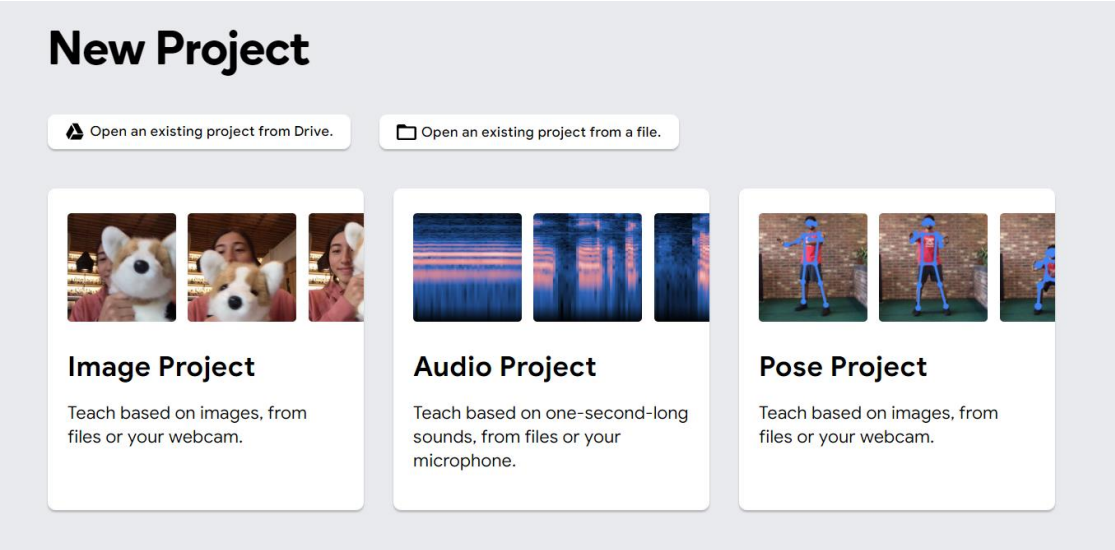


Рис. 1

Google Teachable Machine надає різні типи проектів, включаючи проект зображень (Image Project), проект аудіо (Audio Project) і проект поз (Pose Project). Кожен тип проекту призначений для роботи з певними типами даних та завдань.

У контексті роботи з відео, якщо потрібно навчити модель розпізнавати та класифікувати певні об'єкти або дії на кадрах відео, слід обрати проект зображень (Image Project).

Це робиться насамперед через аналіз на рівні кадрів: за допомогою проекту зображень можна аналізувати кожен кадр відео окремо. Можливо також виділяти та обробляти зображення з кадрів відео, розглядаючи їх як окремі екземпляри для навчання та прогнозування. Це дозволяє зосередитися на візуальному змісті кожного кадру.

Після вибору проекту зображень, нас переносить на сторінку тренування моделі (Рис. 2)

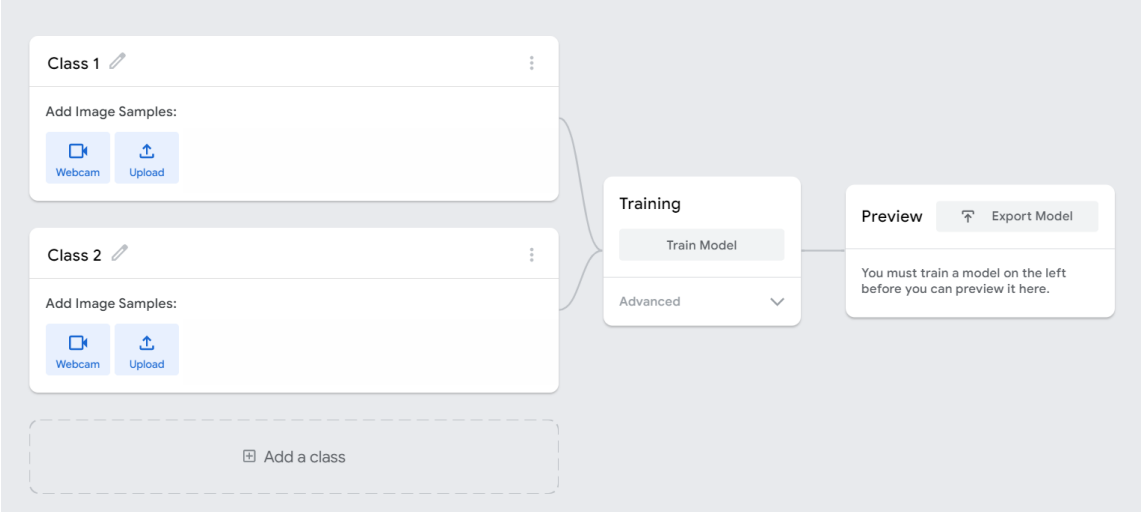


Рис. 2

Процес тренування включає кілька етапів.

Перший етап це збір та підготовка даних. На ньому ми збираємо набір зображень для навчання моделі. Кожне зображення повинно бути призначене для певного класу або категорії, яку потрібно навчити модель розпізнавати. Важливо забезпечувати різноманітність зображень у наборі та баланс між кількістю зображень для кожного класу. Для свого проекту, я знайшов приблизно по 30-40 фотографій людей в масках та без масок, і розсортував їх до двох класів: Mask та No Mask.

Другий етап це навчання моделі: після завантаження та розмітки даних натискаємо на “Train model” та запускаємо процес тренування моделі. Як вже було зазначено, Teachable Machine використовує попередньо навчені моделі, які побудовані на основі нейронних мереж, щоб пристосувати їх до нашого набору даних.

Третій етап це оцінка та налаштування моделі, бо після тренування важливо оцінити її ефективність. Можна використовувати набір тестових зображень, які не брали участь у тренуванні, для вимірювання точності моделі. В моєму випадку, я протестив модель через свою веб-камеру, з’являючись у кадрі по черзі з та без маски на обличчі. У випадку якщо б модель не досягла очікуваної ефективності, можна налаштувати параметри навчання або внести зміни в набір даних. Моя модель мала певні недоліки, але про них ми поговоримо у наступному підрозділі.

Після третього етапу тренування модель готова до застосування. Google Teachable Machine дозволяє нам завантажити модель для використання у своєму проекті.

Свій проект я вирішив писати на мові Python, що, до речі, є вибором номер один для багатьох спеціалістів, працюючих з обробкою зображень та машинним навчанням, у тому числі знаходженням об'єктів у відеопотоці, завдяки ряду його переваг.

Перше, що привертає увагу до Python, це його простота і читабельність. Ця мова знаменита своєю лаконічністю та чистим синтаксисом, що значно полегшує розробку і підтримку коду. Python особливо корисний для більш швидкого прототипування, оскільки він дозволяє розробникам ефективно перевіряти ідеї без багатої підготовки і кодування.

Друга важлива особливість Python полягає в тому, що він має потужну екосистему бібліотек і фреймворків, спеціально призначених для обробки зображень, машинного навчання та наукових обчислень. Наприклад, бібліотеки, як-от OpenCV, TensorFlow, Keras, PyTorch та Scikit-learn, надають готові до використання рішення для роботи з зображеннями та моделями машинного навчання, що значно спрощує розробку.

Пропоную детально розглянути код програми.

Спочатку, імпортуємо до Python-проекту потрібні бібліотеки:

import tensorflow as tf: Імпорт TensorFlow для роботи з нейронними мережами.

import numpy as np: Імпорт бібліотеки NumPy для роботи з масивами даних.

import cv2: Імпорт бібліотеки OpenCV для роботи з обробкою зображень та відео.

from process_labels import generate_labels: Імпорт функції generate_labels зі створеного мною класа process_labels для отримання міток класів.

capture = cv2.VideoCapture(0): Ініціалізація камери та отримання відеопотоку з неї.

loaded_model = tf.keras.models.load_model(...): Завантаження натренованої через Google Teachable Machine моделі TensorFlow.

input_data = np.ndarray(...): Створення масиву для зберігання вхідних даних зображення.

label_dict = generate_labels(): Створення словника міток класів.

while True:: Цикл для безперервного читання кадрів з камери.

font_style = cv2.FONT_HERSHEY_COMPLEX: Вибір стилю шрифту для відображення тексту на відео.

ret_val, video_frame = capture.read(): Читання кадра з камери.

video_frame = cv2.flip(video_frame, 1): Зеркальне відображення кадра, щоб користувач бачив себе у відео, як у дзеркалі.

if not ret_val: continue: Якщо кадр не прочитано правильно, пропустити наступні кроки та повернутися до початку циклу.

video_frame = cv2.rectangle(...): Намалювати прямокутник на кадрі відео.

cropped_frame = video_frame[80:360, 220:530]: Обрізати область кадра, яка відповідає прямокутнику.

resized_frame = cv2.resize(cropped_frame, (224, 224)): Змінити розмір обрізаного кадра до 224x224, що є розміром, необхідним для моделі.

frame_array = np.asarray(resized_frame): Конвертація обрізаного та зміненого зображення у масив NumPy.

norm_frame_array = (frame_array.astype(np.float32) / 127.0) - 1: Нормалізація зображення, перетворення значень пікселів з діапазону [0, 255] до діапазону [-1, 1].

input_data[0] = norm_frame_array: Збереження нормалізованого зображення у масиві **input_data**.

predictions = loaded_model.predict(input_data): Виконання передбачення за допомогою моделі на основі нормалізованого вхідного зображення.

result_idx = np.argmax(predictions[0]): Отримання індексу класу з найвищою ймовірністю.

cv2.putText(...): Відображення мітки класу з найвищою ймовірністю на кадрі відео.

if cv2.waitKey(1) & 0xFF == ord('q'): break: Вихід з циклу, якщо користувач натисне клавішу "q".

cv2.imshow("Object Detector", video_frame): Відображення кадра відео з результатами розпізнавання.

Після завершення циклу - **capture.release()**: Звільнення ресурсів камери.

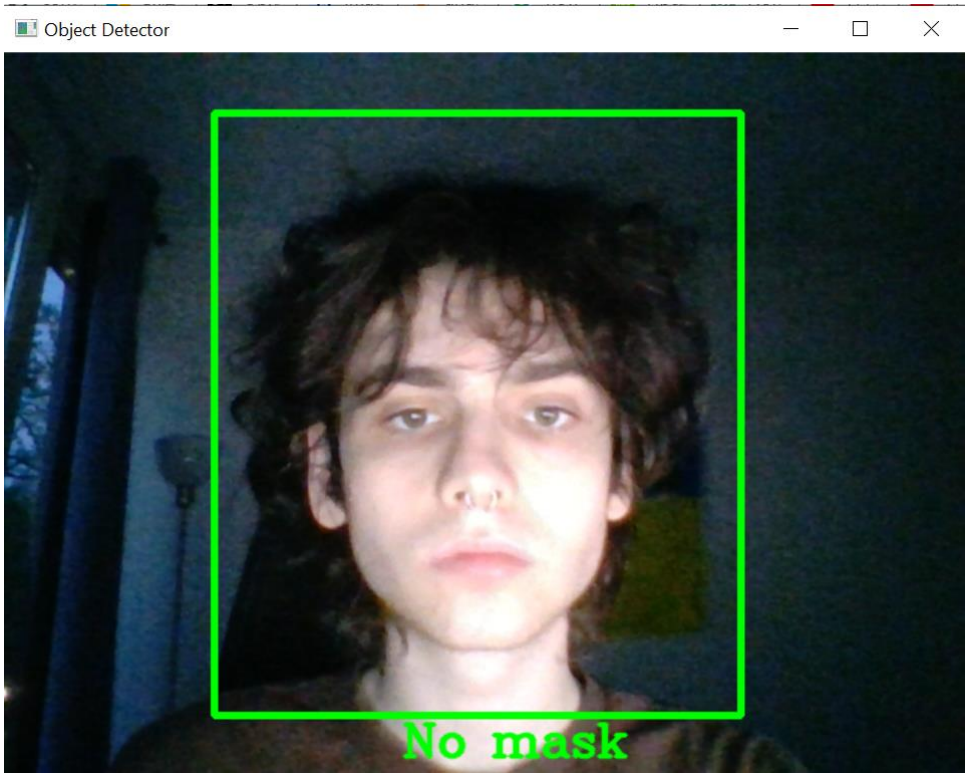
cv2.destroyAllWindows(): Закриття всіх відкритих вікон OpenCV.

Загалом, можна підсумувати що цей код створює систему розпізнавання об'єктів у режимі реального часу, використовуючи TensorFlow для класифікації зображень та OpenCV для обробки відеопотоку.

4.3. Тестування та оцінка результатів.

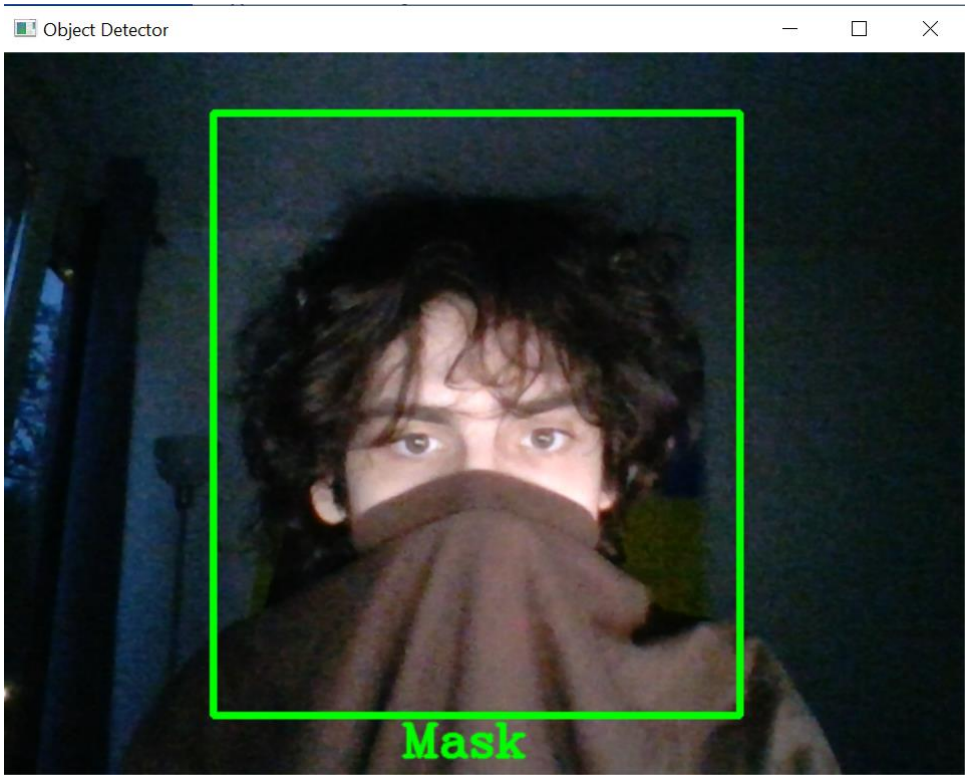
Для того щоб протестувати свою програму, я без довгих міркувань вирішив застосувати самого себе.

Після запуску програми я спочатку показався у кадрі без маски:



Як ми бачимо, програма коректно визначила клас, до якого належить видимий нею кадр, а саме - клас No mask.

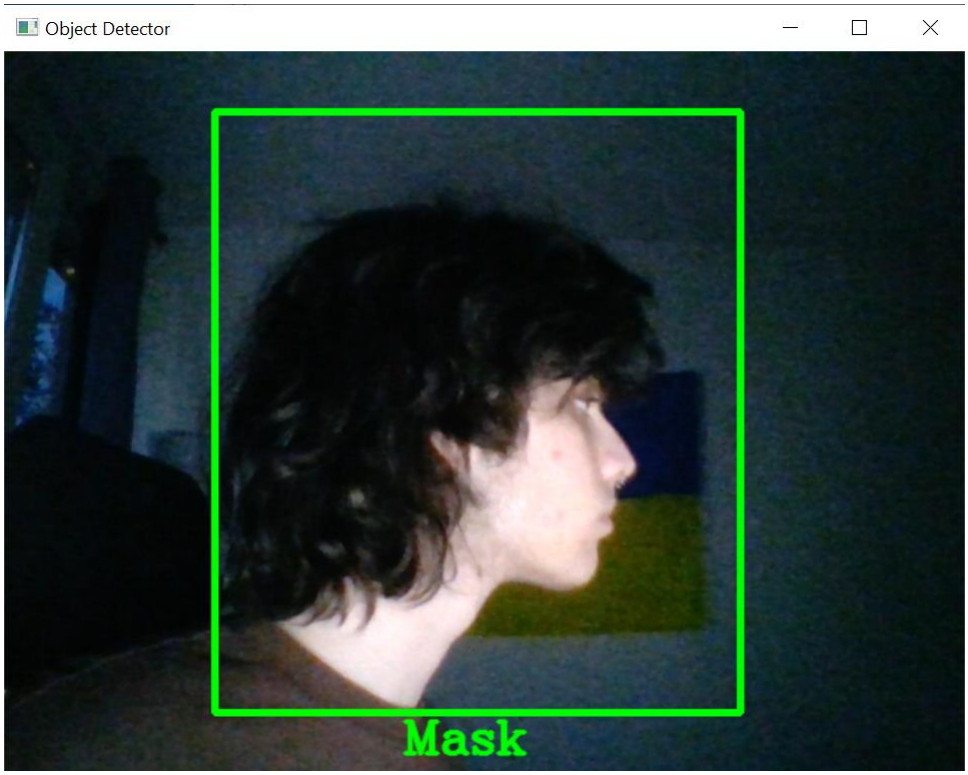
Після тривалих пошуків маски вдома та розуміння що я, на жаль, не маю жодної, я вирішив використати свою сорочку замість маски. В принципі, так як сорочка теж може виконувати головну функцію маски - закривати нижню половину обличчя, та моя натренована модель не має ніяких обмежень по кольору, це задовільна заміна.



Як ми можемо бачити, програма цього разу також правильно визначила клас, в цей раз - Mask.

На жаль, програма також має свої недоліки, про наявність яких я згадував раніше.

Наприклад, у разі повертання голови у профіль, програма у абсолютній більшості випадків визначить це у клас “Mask”, хоча я не вдягнув маску:



Це пов’язано з тим, що наданий мною у Google Teachable Machine ще на стадії тренування моделі матеріал не включав фотографії людей у профіль.

В принципі, так як подібні програми зазвичай використовуються таким чином, що людина буде дивитися прямо в камеру, це не надто серйозна проблема.

У підсумку, результат роботи програми мене задовольнив. Тим не менше, я б точно додав ще як мінімум по стільки ж фотографій в кожен клас на стадії тренування моделі. Це б значно покращило роботу програми та зробило б вихідний результат більш точним та надійним.

5. Висновки.

5.1. Основні результати дослідження.

У результаті дослідження теми знаходження об'єктів у відеопотоці було виявлено значну кількість важливих аспектів.

Спочатку були розглянуті теоретичні основи знаходження об'єктів у відеопотоці, включаючи основні поняття та термінологію, методи та підходи до розпізнавання об'єктів в відео, а також було проведено огляд сучасних алгоритмів та інструментів.

Після того, як були встановлені теоретичні засади, увага була зосереджена на нейронних мережах як інструменті на знаходження об'єктів у відеопотоці. У цьому контексті були досліджені основні принципи нейронних мереж, різні архітектури нейронних мереж для обробки зображень та відео, а також процес тренування нейронних мереж для розпізнавання об'єктів.

На заключному етапі дослідження основна увага була приділена використанню натренованої нейронної мережі для знаходження об'єктів у відеопотоці. Було проведено вибір та аналіз натренованої моделі, описано тему програми та проведено детальне роз’яснення процесу її побудови та коду. Також було виконано тестування та оцінка результатів.

Загалом, результати цього дослідження показали, що нейронні мережі можуть бути ефективно використані для знаходження об'єктів у відеопотоці, та процес вибору, тренування та використання відповідної моделі не завжди вимагає значного розуміння теорії та практики в цій області, завдяки таким ресурсам як Google Teachable Machine.

5.2. Аналіз майбутнього теми.

Аналізуючи майбутнє теми знаходження об'єктів у відеопотоці, можна стверджувати, що ця область залишатиметься надзвичайно актуальною. Це пов'язано з тим, що розпізнавання об'єктів в відео дедалі більше використовується в широкому спектрі застосувань, від автономних транспортних засобів і систем безпеки до аналітики споживачів і розваг.

В автономних транспортних засобах, наприклад, здатність розпізнавати і відстежувати об'єкти в реальному часі є ключовою для безпечної і ефективної навігації. Подібні технології також стають все важливішими в галузі безпеки, де вони дозволяють автоматично виявляти підозрілу поведінку або небезпечні ситуації.

Але можливості знаходження об'єктів у відеопотоці далеко виходять за рамки цих застосувань. В аналітиці споживачів, наприклад, вони можуть бути використані для відстеження поведінки покупців у магазинах і навіть для аналізу емоцій. В розвагах вони можуть бути використані для створення більш реалістичних і занурювальних віртуальних світів.

Враховуючи ці різноманітні застосування, здається, що знаходження об'єктів у відеопотоці залишатиметься важливою областю дослідження і розвитку. При цьому, як технології стають все більш потужними і доступними, можливо, ми побачимо ще більше інноваційних застосувань в цій області.

6. Список використаної літератури.

Електронний ресурс: https://www.fritz.ai/object-detection/#:~:text=Object%20detection%20is%20a%20computer%20vision%20technique%20that%20works%20to,move%20through)%20a%20given%20scene.

Електронний ресурс: https://www.mathworks.com/discovery/object-detection.html

Електронний ресурс: https://www.hackster.io/mjdargen/easy-object-detection-with-teachable-machine-python-d4063b

Книга: Szeliski, R. (2010) - Computer Vision: Algorithms and Applications.

Електронний ресурс: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Електронний ресурс: https://medium.com/@meetjethwa3/object-detection-in-5-mins-58d322008af2

Електронний ресурс: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

Електронний ресурс: https://towardsdatascience.com/object-detection-with-neural-networks-a4e2c46b4491