ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка системи розпізнавання облич людей на вході в домівку»

на здобуття освітнього ступеня бакалавр
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні науки
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Сергій ДОВГУН
(підпис) *(Ім'я, ПРІЗВИЩЕ здобувача)*

Виконав:
здобувач вищої освіти групи КНД-41 Сергій ДОВГУН

Керівник:
науковий ступінь, вчене звання Юлія БЕРЕЗОВСЬКА
доктор філософії

Рецензент:
науковий ступінь, вчене звання _____
(Ім'я, ПРІЗВИЩЕ)

Київ – 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Комп'ютерних наук
Ступінь вищої освіти Бакалавр
Спеціальність 122 Комп'ютерні науки
Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук

_____ В.В. Вишнівський

"_____" _____ 20__ року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ

Довгун Сергій Вікторович
(прізвище, ім'я, по батькові)

1. Тема роботи: « Розробка системи розпізнавання облич людей на вході в домівку»
Керівник роботи Юлія БЕРЕЗОВСЬКА, доктор філософії
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «27» лютого 2024 року № 36
2. Строк подання кваліфікаційної роботи 31 травня 2024р.

3. Вхідні дані до роботи:
Методи визначення рухомих об'єктів у відеопослідовності.
Методи визначення об'єктів на зображенні.
Науково-технічна література. Стандарти. Рекомендації.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
4.1 Огляд сучасних методів розпізнавання об'єктів у відеопослідовності.
4.2 Постановка задачі і розробка моделі розпізнавання рухів об'єктів на вході в домівку.
4.3 Розробка програмного забезпечення алгоритму розпізнавання облич людей на вході в домівку.

5. Перелік демонстраційного матеріалу: презентація

6. Дата видачі завдання 23 лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз науково-технічної літератури	22.03.2024 р.	
2.	Модель системи розпізнавання об’єктів на вході в домівку.	05.04.2024 р.	
3.	Модель розпізнавання облич людей на вході в домівку.	26.04.2024 р.	
4.	Розробка алгоритму розпізнавання облич людей на вході в домівку.	03.05.2024 р.	
5.	Розробка програмного забезпечення алгоритму розпізнавання облич людей на вході в домівку.	10.05.2024 р.	
6.	Реферат, вступ, висновки.	15.05.2024 р.	
7.	Підготовка презентації до захисту.	20.05.2024 р.	

Здобувач вищої освіти _____ Сергій ДОВГУН
(підпис) (Ім’я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи _____ Юлія БЕРЕЗОВСЬКА
(підпис) (Ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 60 сторінок, 47 рисунків, 12 джерел.

Мета роботи – підвищити ефективність систем охорони за рахунок розпізнавання облич людей на вході в домівку.

Об’єкт дослідження – процес розпізнавання і класифікації рухомих об’єктів.

Предмет дослідження – методи і алгоритми розпізнавання рухомих об’єктів.

Методи дослідження – методи розпізнавання руху на кадрі, методи розпізнавання об’єктів за контуром, обробка та аналіз отриманих результатів.

Досліджено множину методів для вирішення задачі оптимізації класифікації рухомих об’єктів.

Обрано нейронні мережі та використано навчену відкриту нейронну мережу MobileNet_SSD.

За вибраним методом розпізнавання розроблено модель системи та алгоритм роботи програми.

На основі результатів виконаних досліджень розроблено програмне забезпечення, що розпізнає обличчя людей на вході в домівку, а також класифікує їх.

Простежено, що при використанні нейронних мереж система показує задовільні результати, але має незначні погрішності.

Упровадження розробленої схеми дозволяє без великих затрат використовувати систему розпізнавання облич людей на вході в домівку та модифікувати цю систему під конкретні цілі споживача.

ТЕХНОЛОГІЯ НЕЙРОННИХ МЕРЕЖ, КОМП’ЮТЕРНИЙ ЗІР, ОПТИМАЛЬНІ МЕТОДИ, РОЗПІЗНАВАННЯ ОБЛИЧ, PYTHON, OPENCV, КЛАСИФІКАЦІЯ ОБ’ЄКТІВ.

ЗМІСТ

ВСТУП

У міру того, як штучний інтелект проникає в життя людей, виявлення руху у відеопотоках стає поширеною проблемою. Автоматизовані системи відеоспостереження з класифікацією облич можуть усунути людський фактор з цього процесу, не змушуючи спостерігачів дивитися на кілька камер одночасно.

Ця тенденція особливо актуальна для охорони здоров'я, безпеки та комерційних рішень. Доктор Адріан Розброк провів велику роботу з розпізнавання відео та зображень. Для цього він використовував електронний ресурс PyImageSearch, який всебічно охоплює всі аспекти комп'ютерного зору.

Об'єктом досліджень є процес розпізнавання і класифікації рухомих об'єктів.

Предметом дослідження є методи і алгоритми розпізнавання рухомих об'єктів.

Метою роботи є підвищення ефективності систем безпеки за рахунок розпізнавання облич людей на вході в домівку.

Методи дослідження – методи розпізнавання руху на кадрі, методи розпізнавання об'єктів за контуром, обробка та аналіз отриманих результатів.

Задачами дослідження є:

проаналізувати методи розпізнавання руху;

вибір найкращого варіанту на основі аналізу;

розробка моделі системи розпізнавання та програмного алгоритму;

розробити та протестувати програмне забезпечення на основі алгоритму.

Наукова новизна роботи полягає у спрощеному застосуванні рішень комп'ютерного зору.

Практичне значення отриманих результатів полягає у подальших дослідженнях та розробках з класифікації об'єктів та використання нейронних мереж.

1 АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ВІДЕОПОСЛІДОВНОСТЯХ

1.1 Огляд проблеми

Проблема розпізнавання об'єктів у відеопослідовностях є дуже важливою і поширеною, особливо в умовах змінної освітленості, часткового затемнення або шуму в зображенні. На сьогоднішній день існують різноманітні методи розпізнавання облич, такі як метод віднімання фону, метод стеження за унікальними точками, фазова кореляція та методи розпізнавання за контуром, які використовуються для виявлення та аналізу об'єктів у відеопослідовностях. Однак, кожен з цих методів має свої переваги та обмеження, і досягнення точності та надійності в розпізнаванні облич залишається важкою задачею. Також є недоліки у випадках, коли потрібно розрізняти, наприклад, рух людей від руху тварин, або рух різних типів тварин між собою. Тому важливо постійно розвивати та вдосконалювати методи розпізнавання облич, щоб забезпечити їх ефективність у різноманітних умовах застосування.

Існують важливі застосування в безпеці, криміналістиці, штучному інтелекті, психології та інших галузях, де важливо розпізнавати обличчя на відео, щоб ідентифікувати об'єкт, який виконує рух, або запобігти поведінці, яку виконує цей об'єкт. Приклади включають системи безпеки для виявлення грабіжників і зловмисників, що проникають в будівлі, системи розпізнавання облич і жестів для ідентифікації людської поведінки як засобу спілкування з машинами, а також розпізнавання мікрообличчя для розуміння намірів людини. Існують також інші системи спостереження, такі як камери для спостереження за громадянами на вулицях і виявлення злочинних нападів, а також системи домашнього відеоспостереження для підтримки будинку, коли власник відсутній.

Системи виявлення руху можуть допомогти запобігти хибним тривогам на надсекретних об'єктах, де доступ до інформації та розробок потрібен лише певним

людям, усуваючи необхідність мобілізації всіх сил безпеки для усунення загрози випадкового потрапляння тварин на об’єкт.

Це також важливо для захисту військових баз і прикордонних територій ворожих країн. Система розпізнає рухомі об’єкти та визначає їхню класифікацію. Залежно від ситуації, система передає управління військовим і виконує дію, зазначену в програмному забезпеченні системи, якщо вона може контролювати ситуацію.

У цивільному житті система може вмикати світло та електроприлади, регулювати температуру в кімнатах та інших приміщеннях, відтворювати новини та вмикати розважальну апаратуру, якщо присутній господар. Вона також може стежити за будинком, коли господаря немає вдома, і подавати сигнал тривоги, щоб повідомити господаря, що хтось є вдома. Цю систему можна використовувати в системах "розумного будинку", які сповіщають власника про те, що хтось є вдома.

Такі системи можуть розпізнавати рухи людського тіла і зв'язуватися з комп’ютером, щоб отримати доступ до потрібного об’єкта.

Люди з обмеженими можливостями можуть спілкуватися зі здоровими людьми завдяки інтерфейсам, які розпізнають жести та міміку.

У сфері спорту система зможе розпізнавати, як швидко біжить спортсмен або які рухи неправильні під час тренування, і фіксувати кількість необхідних точок руху.

У кожній з цих функцій ключовими характеристиками системи повинні бути виявлення руху, позиціонування об'єкта і класифікація об’єктів. Це основні та найважливіші параметри для додатків розпізнавання руху. Для вирішення цієї проблеми розглядаються методи виявлення руху для виявлення руху і позиціонування об’єктів, а також методи розпізнавання форми для класифікації об’єктів.

Після того, як всі методи розглянуті, вибирається один або декілька методів в залежності від поставленого завдання. Обирається найефективніший метод, який відповідає всім вимогам і може бути реалізований в додатку або системі розпізнавання. Тому важливо підійти до вирішення проблеми, вибравши найбільш

підходящий варіант з представлених нижче методів на основі швидкості обробки, використання пам’яті, простоти реалізації та найкращого наближення до ідеального результату реалізації. Після вибору найбільш підходящого методу необхідно розробити алгоритми і створити програмне забезпечення для вирішення проблеми розпізнавання поведінки в залежності від поставленої задачі.

1.2 Системи безпеки

Системи безпеки – це пристрої, що складаються з датчиків, які запобігають або попереджають різні негативні події. Основна сфера застосування – системи відеоспостереження. Такі системи можна розділити на дротові та бездротові.

Дротові системи зазвичай встановлюються на етапі ремонту або будівництва, щоб запобігти пошкодженню внутрішніх приміщень будівлі. Такі системи довговічні і не страждають від проблем з електроживленням, тому збоїв і помилкових спрацьовувань через обмеження потужності не відбувається. Недоліком є висока вартість встановлення системи. На рисунку 1.1 представлені камери стеження, бренду Hikvision.

Рисунок 1.1 – Камери стеження

Бездротові мережі мають значні переваги з точки зору встановлення. Вони

вимагають дуже мало праці та матеріалів. Залежно від моделі, все, що потрібно – це клейка стрічка і викрутка. Такі системи можна легко транспортувати з одного місця на інше. Сам датчик зазвичай підключений до мережі, куди записуються і завантажуються відеозображення. Тому такі системи більш мобільні, ніж дротові. Основний недолік – проблема з батареєю. Такі системи не встановлюють, якщо приміщення не опалюється, оскільки батареї можуть вийти з ладу. Крім того, батареї доводиться постійно міняти, що тисне на гаманець" [5]. На рисунку 1.2 представлена бездротова система охорони, бренду Ajax.

Рисунок 1.2 – Бездротова система охорони

Більшість постачальників систем безпеки зараз пропонують комплекти відеоспостереження, що складаються з набору камер, інфрачервоного датчика та відеореєстратора, який записує зображення з камер у режимі реального часу. Можуть бути й інші компоненти, залежно від потреб користувача. Більшість таких систем призначені для "ручного" відеоспостереження, що дозволяє користувачеві бачити, що відбувається перед камерою в будь-який час. Цей тип систем є дуже економічно вигідним і простим у використанні.

У ритмі сучасного життя не так багато можливостей переглядати записи з камер відеоспостереження. Але це не єдина проблема. Навіть якщо камери регулярно перевіряються, вони можуть не вловити момент, коли зловмисник

проникає в будинок і зникає з поля зору камери. Також незручно наймати охоронців для постійного спостереження.

Автоматизовані системи відеоспостереження (рис. 1.3) є новим рішенням цієї проблеми. Автоматизовані системи відеоспостереження можуть отримувати зображення з камер і делегувати завдання виявлення руху об’єктів програмі, яка розпізнає рухомі об’єкти в будівлі або приміщенні. Це усуває необхідність постійного моніторингу з боку власника житла або охоронців. Якщо домовласник не може відреагувати на вторгнення, система приймає рішення замість нього, або ж домовласник отримує сповіщення про вторгнення і приймає рішення на основі даних, отриманих від системи.

Рисунок 1.3 – Камера з автоматичним стеженням пересування

Такі системи еволюціонують у бездротові моделі розміром з маленьку іграшку, але мають потенціал звичайної камери.

Виробники автоматичних камер випускають різні версії камер безпеки. Деякі камери відстежують кожен рух, інші мають вузьке поле зору. Деякі можуть виявляти людей у кадрі та автоматично позначати їх як зловмисників. Такі камери можуть відрізняти випадкові рухи об'єктів від різних штучних рухів, а також ідентифікувати людей за формою, кольором тощо.

Проблема такого підходу полягає в тому, що камера може розпізнати об’єкт як людину, коли він не має нічого спільного з людиною. Ця помилка називається первинною. Інший варіант – система не розпізнає людину в кадрі або розпізнає її

як дружній об’єкт. Цей тип помилки називається вторинною помилкою. Зазвичай тривалість таких помилок не перевищує однієї секунди, але однієї секунди може бути достатньо, щоб система була задоволена своїм вибором і вжила певних заходів для запобігання вторгненню.

Тому, важливо внести зміни до програмного забезпечення розпізнавання. Це означає навчити систему розпізнавати не лише людей, а й об’єкти, які часто мають невідомі рухи. Таким чином, система зможе розпізнавати, наприклад, тварин як тварин і зменшити ймовірність помилок системи. Крім того, системи з таким типом прошивки набагато дешевші, оскільки рішення простіше.

1.3 Методи розпізнавання руху в системах відеоспостереження

Виявлення руху можна використовувати, щоб визначити, чи присутній рух на зображенні або відео, і точно визначити, де саме він відбувається.

У таких методах важливими є ефективність і простота реалізації.

До методів виявлення руху належать:

віднімання фону (виділення об’єктів, які сильніші або слабші за оригінальну модель);

відстеження власних точок (присвоєння точки пікселю та відстеження її руху);

фазова кореляція (зміна фаз);

нейронні мережі.

Ці методи популярні, оскільки вони прості в реалізації та ефективні.

Метод віднімання фону використовується, коли камера нерухома і фон не змінюється. Він працює наступним чином: із заданою частотою, починаючи з першого кадру послідовності, програма вибирає кадр як еталонну модель і віднімає пікселі еталонного кадру від поточного кадру (рис. 1.4). Якщо в результаті обчислень виходить різниця, яку можна виправити, програма виділяє частину поточного кадру, що відрізняється від еталонного.

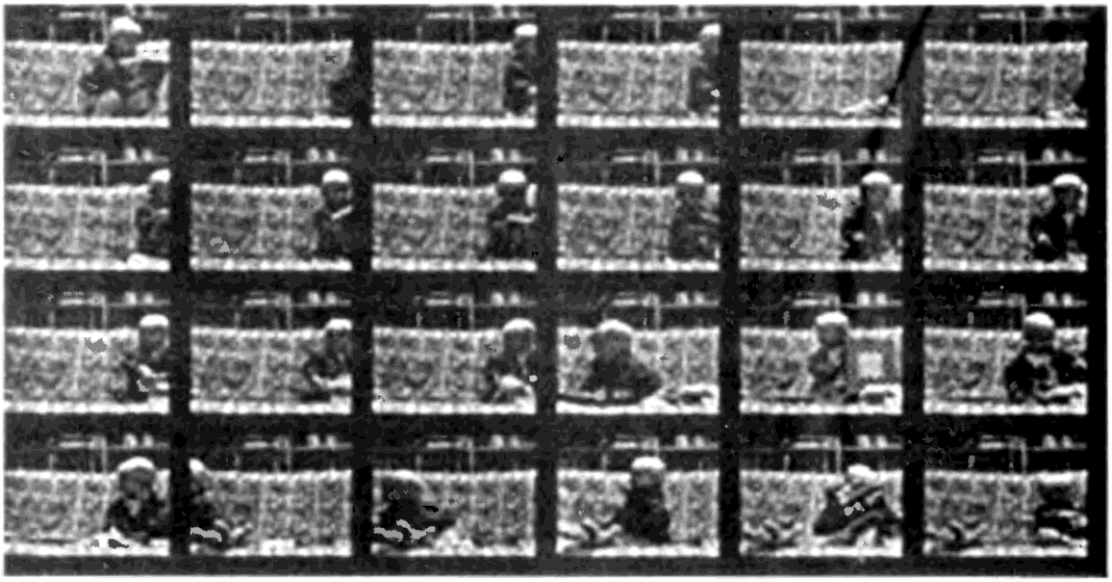


Рисунок 1.4 – Віднімання фону і визначення меж кадрів

Це дозволяє дуже легко автоматизувати розпізнавання рухомих об’єктів на фіксованому фоні. Однак, простота реалізації має свої недоліки, такі як низька якість зображення, відеошум, а також зміни кольору і яскравості через погодні умови та зміну дня і ночі. Тому, цей метод використовується лише з модифікаціями, які згладжують шуми і роблять зміни дня/ночі та погоди незначними [6].

На рисунку 1.5 наведено приклад того, як працює віднімання фону.

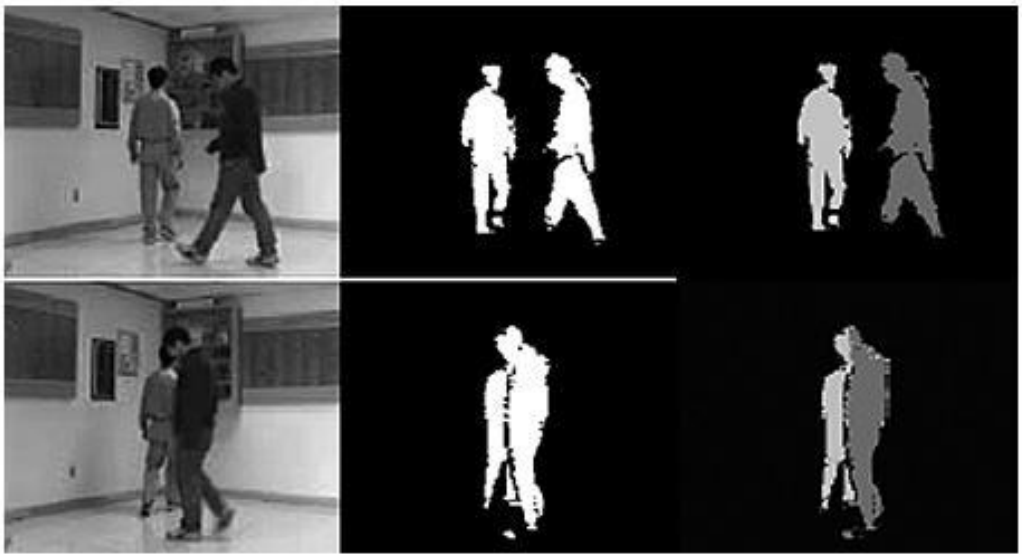


Рисунок 1.5 – Приклад роботи методу віднімання фону

Унікальне точкове відстеження руху використовується для виявлення конкретних об’єктів.

Він ґрунтується на призначенні пікселя або масиву пікселів одній унікальній точці або “трекеру”. Кілька трекерів, призначених об’єкту, рухаються вздовж частини об’єкта, тому призначення трекера на кожну кінцівку не тільки відстежує об’єкт, але й дозволяє певною мірою визначити його положення.

Цей метод широко використовується для розпізнавання міміки і жестів людини, а також може використовуватися машинами для імітації людських рухів.

Цей метод не позбавлений недоліків. Основною проблемою цього алгоритму є оклюзія. Оклюзія – це явище, коли рухомий об’єкт прихований за іншим нерухомим або рухомим об’єктом з точки зору спостерігача. Тому, якщо людина, за якою стежать, ховається в одному місці, а потім з’являється під іншим кутом, пристрій стеження може не знайти масив пікселів для продовження стеження за цим об’єктом.

Приклад роботи пристрою стеження показаний на рисунку 1.6.

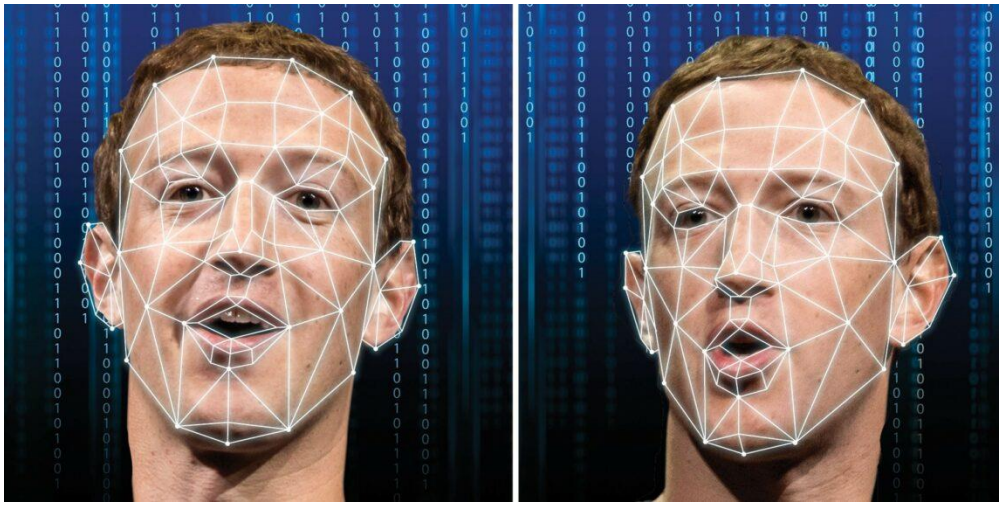


Рисунок 1.6 – Метод стеження за точками

Фазова кореляція – це метод аналізу сигналів, який використовується для вимірювання подібності між двома або більше сигналами або шаблонами на основі їх фазових характеристик. Цей метод часто застосовується в області обробки сигналів, комп’ютерного зору та машинного навчання.

Основна ідея фазової кореляції полягає в порівнянні фазових характеристик двох сигналів, таких як зображення, звукові сигнали або будь-які інші типи сигналів. Фаза сигналу вказує на зсув або зміщення сигналу в часі відносно початку відліку.

У контексті обробки зображень фазова кореляція може використовуватися для пошуку об’єктів на зображенні або для визначення ступеня схожості між двома зображеннями. Цей метод може бути особливо корисним у задачах відстеження об’єктів або в розпізнаванні образів.

Однією з переваг фазової кореляції є те, що вона враховує структурні характеристики сигналу, що може призвести до більш точних результатів порівняння, особливо у випадках, коли сигнали піддані зміщенням або зміні масштабу (рис. 1.7). Однак цей метод може бути вимогливим до обчислювальних ресурсів, особливо при аналізі великої кількості даних.

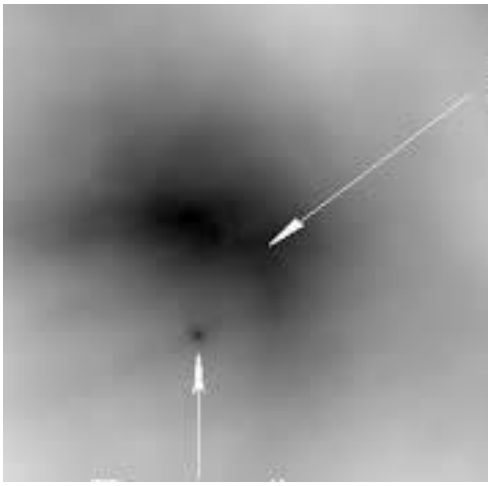


Рисунок 1.7 – Фазова кореляція

Методи розпізнавання за контуром – це техніки обробки зображень, спрямовані на виявлення та аналіз контурів об’єктів на зображеннях. Контур – це зовнішній обрис об’єкта на зображенні, який відокремлює його від тла.

Основна ідея методів розпізнавання за контуром полягає в виявленні ліній, кривих або інших форм, що представляють контур об’єкта. Це може включати в

себе використання алгоритмів обробки зображень для виявлення змін у яскравості, кольорі або текстурі пікселів, які відповідають контуру об’єкта. Далі можуть використовуватися різноманітні методи аналізу форми, такі як виявлення країв, згорткові фільтри, методи границь та інші, для подальшого визначення форми та розмірів об’єктів на зображенні.

Методи розпізнавання за контуром можуть бути використані в багатьох областях, включаючи комп’ютерний зір, медичне зображення, автоматизовану обробку зображень та багато інших. Вони дозволяють автоматично визначати та аналізувати об’єкти на зображеннях, що важливо для багатьох задач, таких як відстеження об’єктів, розпізнавання облич, медична діагностика тощо.

1.4 Нейронні мережі для розпізнавання об’єктів на зображеннях

Нейронні мережі для розпізнавання об’єктів на зображеннях – це техніки машинного навчання, які використовують нейронні мережі для виявлення та класифікації об’єктів на зображеннях. Ці методи використовуються для автоматичного виявлення об’єктів на зображеннях без необхідності в ручному визначенні ознак або шаблонів.

Основна ідея полягає в тому, що нейронна мережа навчається аналізувати вхідні зображення та відтворювати їхні особливості, що відповідають певним класам об’єктів. Для цього можуть використовуватися різноманітні архітектури нейронних мереж, такі як згорткові нейронні мережі (CNN), які спеціалізуються на обробці зображень та здатні виявляти локальні особливості та шаблони у вхідних даних.

Процес навчання нейронної мережі для розпізнавання об’єктів на зображеннях включає в себе подачу великої кількості зображень разом з відповідними мітками або класами об’єктів, які потрібно розпізнати. Під час навчання мережа автоматично виявляє внутрішні зв’язки та шаблони між пікселями зображення та відповідними класами об’єктів.

Після завершення навчання нейронна мережа може бути використана для розпізнавання об’єктів на нових зображеннях, де вона аналізує зображення та видає вихід, що вказує на наявність та положення об’єктів на зображенні. Ці методи широко застосовуються в різних областях, включаючи комп’ютерний зір, медичне зображення, автономні транспортні засоби та багато інших.

Штучні нейронні мережі дозволяють покращити функціональність користувацького додатку без явного програмування. Такий додаток має можливість розпізнавати колір, контур, форму, звук, послідовність і т.д. в залежності від даних, введених під час навчання. Таким чином, користувач може аналізувати окремі події, знаходити інформацію про об’єкти по зображеннях, створювати твори мистецтва.

Нейронним мережам слід приділяти особливу увагу в області систем безпеки. Використовуючи камеру для автоматизації пошуку злочинців, є можливість вчасно відповісти 24/7 без втручання людини. Автоматична система стеження також відіграє важливу роль.

Таким чином, нейронні мережі можуть не тільки розпізнавати досліджуваний об’єкт, але й розпізнавати його стан, коли вони знають, хто що і як робить.

Загальна схема нейронної мережі наведена на рисунку 1.8.

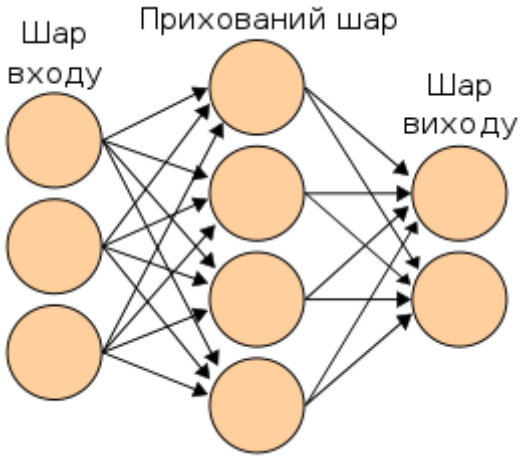


Рисунок 1.8 – Схема нейронної мережі

Для них готова нейронна мережа або бібліотека:

– BrainCL Chung – це невелика програма з бібліотекою, написаною для обчислення трьох або чотирирівневих нейронних мереж з великими вхідними та вихідними даними за допомогою прискорення OpenCL, написаного у швидкій та простій відкритій базі даних;

– Chemical Reaction Prediction – нейронна мережа прогнозування хімічних реакцій, яка призначена для прогнозування хімічних реакцій з відомими реагентами. Метою є створення продукту реакції за допомогою глибокого навчання. Він використовує графічний інтерфейс користувача зі спрощеною системою введення формул;

– Easy Agent Simulation – дана платформа є “плагіном”, і її зв’язок переривається і використовується окремо для зв’язку між механізмом моделювання та симуляцією моделі. Він був розроблений на основі нової архітектури з використанням “прикладного рівня”. Це дозволяє додавати та видаляти незалежні від моделювання функції, такі як візуалізації, карти, діаграми та побудова траєкторій. Дослідження показали, що програма проста у використанні і доступна для початківців, які займаються моделюванням на різних рівнях складності. Дослідницька документація включена в програму;

– AIPac – це клон гри PacMan, в якій нейронна мережа навчилася грати в цю гру;

– Starnet + призначений для збору зірок із фотографій, отримання більш чітких туманностей у зоряних скупченнях та отримання красивих фоторамок, де потрібно відображати зображення без зірок;

– GMDH Shell – це програма, яка використовується для створення та навчання нейронних мереж для таких завдань, як аналіз, прогнозування та використання великих даних. Нейронні мережі використовують більш гнучкий тип передбачення, ніж лінійне або поліноміальне передбачення, тому Ви можете більш точно підійти до правильної відповіді. Крім того, оболонка GMDH Shell не вимагає нормалізації даних і дозволяє виконувати швидше обчислення;

– MobileNet_SSD – це нейронна мережа з набором класів, призначених для розпізнавання об’єктів на фотографіях і в реальному часі. На цьому етапі можна

ігнорувати невикористані об’єкти та налаштувати їх для класів, необхідних для завдання.

1.5 Висновок до розділу 1

Розглянуто сучасні методи та підходи до розпізнавання рухів людей і тварин у відеопослідовностях. Було відзначено, що розвиток технологій відеоспостереження та комп’ютерного зору значно полегшив аналіз поведінки об’єктів у відео. Зокрема, зростаюча роль штучного інтелекту, зокрема нейронних мереж, дозволяє досягати високої точності розпізнавання та ідентифікації об’єктів.

Визначено основні виклики та проблеми, пов’язані з розпізнаванням рухів об’єктів у відеопослідовностях. Серед основних проблем – високий рівень складності обробки великих обсягів відеоданих, варіативність умов зйомки (освітлення, кути огляду тощо), а також складність розпізнавання об’єктів у різних контекстах (наприклад, у натовпі або при частковому перекритті об’єктів).

Розгляд систем охорони показав, що сучасні системи відеоспостереження активно використовують методи розпізнавання рухів для підвищення ефективності безпеки. Впровадження автоматичних систем аналізу відеопотоків дозволяє швидше реагувати на потенційні загрози, забезпечувати моніторинг великих територій та мінімізувати людський фактор в охороні.

Розглянуто різні методи розпізнавання руху, такі як оптичний потік, фонове віднімання та використання нейронних мереж. Кожен з методів має свої переваги та недоліки, і вибір конкретного методу залежить від умов застосування та вимог до точності розпізнавання. Було відзначено, що комбінування методів може значно підвищити точність та надійність системи.

Підрозділ присвячено ролі нейронних мереж у розпізнаванні об’єктів на зображеннях. Було розглянуто різні типи нейронних мереж, такі як згорткові нейронні мережі (CNN), які особливо ефективні для обробки зображень та відео. Використання нейронних мереж дозволяє досягати високої точності розпізнавання

завдяки можливості навчання на великих обсягах даних та здатності до самовдосконалення.

2 СТВОРЕННЯ МОДЕЛІ ДЛЯ РОЗПІЗНАВАННЯ РУХІВ ОБ'ЄКТІВ НА ВХОДІ В ДОМІВКУ

2.1 Вимоги до системи

Програмне забезпечення для підключення до камер відеоспостереження на вході в домівку виконує розпізнавання об'єктів у кадрі за допомогою навченої нейронної мережі, яка передає кадр з відеопотоку.

Щоб зменшити кількість помилок, нейронній мережі необхідно кілька разів знаходити один і той же людський об'єкт, щоб забезпечити розпізнавання людини, яке досягається шляхом підрахунку кількості послідовних кадрів, в яких програма розпізнає обличчя. Отже, якщо система помилково розпізнає об'єкт, що не є особистістю, як особистість, помилка зводиться до мінімуму.

Як тільки система розпізнає людину, необхідно попередити власника про вторгнення. Це забезпечується системою, що відправляє повідомлення мешканцям будинку. Таким чином, система може повідомити людину про неприємну ситуацію.

Крім того, нейронна мережа повинна виділяти й інші об'єкти, такі як домашні тварини. Очікуваний рух в будинку-це саме рух домашньої тварини або людини. Ця функція використовується для забезпечення того, щоб рухомий об'єкт система не переплутала з об'єктом противника.

Дане програмне забезпечення спрямоване на забезпечення високого рівня безпеки в домі, надаючи можливість точно розпізнавати обличчя людей, мінімізуючи помилкові спрацьовування і оперативно інформуючи власників про потенційні загрози.

Основні функції програмного забезпечення:

1. Вибір кадру з відеопотоку: а) захоплення відеопотоку – програма підключається до камер відеоспостереження та отримує відеопотік у реальному часі; б) обробка відео – відеопотік розбивається на окремі кадри для подальшого

аналізу; в) частота вибору кадрів може бути налаштована в залежності від вимог до точності та продуктивності; г) попередня обробка – кожен кадр піддається попередній обробці для видалення шуму, нормалізації освітлення та підготовки до аналізу нейронною мережею;

2. Розпізнавання обличчя людини у відібраному кадрі: а) аналіз кадру – нейронна мережа аналізує кожен кадр, виявляючи наявність людських об’єктів; б) ідентифікація людини, для зменшення кількості помилкових спрацьовувань, програмне забезпечення перевіряє наявність людини в кількох послідовних кадрах. Якщо людина розпізнається у встановленій кількості кадрів поспіль, система вважає це підтвердженням її присутності; в) фільтрація помилкових спрацьовувань – використання методів машинного навчання для зменшення ймовірності помилкового розпізнавання об’єктів, які можуть бути помилково прийняті за людину;

3. Відправка повідомлень власникам у разі виявлення людини: а) генерація сповіщення – після підтвердження присутності людини в кадрі, система створює сповіщення про потенційне вторгнення; б) відправка повідомлення – сповіщення може бути відправлено через різні канали зв’язку, такі як SMS, електронна пошта, push-повідомлення на мобільний додаток, або через месенджери (наприклад, WhatsApp, Telegram); в) налаштування оповіщень – користувачі можуть налаштувати пріоритетність та канали отримання сповіщень, а також задати умови для різних типів подій;

4. Розпізнавання домашніх тварин у відібраному кадрі: а) аналіз кадру – нейронна мережа аналізує кожен кадр, виявляючи наявність домашніх тварин; б) ідентифікація тварин – програма використовує окремі моделі або спеціально налаштовані нейронні мережі для точного розпізнавання різних типів домашніх тварин; в) уникнення помилкових спрацьовувань – програма налаштована так, щоб розрізняти звичайну активність домашніх тварин від потенційних загроз, зменшуючи ймовірність помилкового спрацьовування.

Додаткові функції та розширення:

1. Налаштування чутливості: а) регулювання чутливості – користувачі можуть налаштувати рівень чутливості до руху для уникнення зайвих сповіщень; б) зони спостереження – можливість налаштування зон спостереження, де програма буде найбільш чутливою до руху;

2. Історія та журнали: а) збереження подій – програма веде журнал всіх подій, включаючи розпізнані рухи та відправлені сповіщення; б) аналіз історії – користувачі можуть переглядати історію подій для аналізу та налаштування системи;

3. Інтеграція з іншими системами: а) системи безпеки – програма може інтегруватися з існуючими системами безпеки для покращення загальної ефективності; б) системи автоматизації – підключення до систем "розумного дому" для площ, автоматичного керування освітленням, замками тощо.

Режими роботи: денний режим (оптимізація для роботи при денному освітленні), нічний режим (використання інфрачервоних камер або налаштування для роботи при низькому освітленні).

Використовуючи масштабування за допомогою підтримки декількох камер є можливість одночасного підключення до кількох камер для моніторингу різних зон у будинку. Моніторинг великих площ оптимізує роботу в будинках різного розміру та конфігурації.

Ці функції роблять програмне забезпечення надійним інструментом для забезпечення безпеки та моніторингу активності у житлових приміщеннях, забезпечуючи високу точність розпізнавання та своєчасне інформування користувачів про потенційні загрози.

2.2 Розробка моделі системи розпізнавання

На рисунку 2.1 показана модель системи розпізнавання рухів об’єктів на вході в домівку.

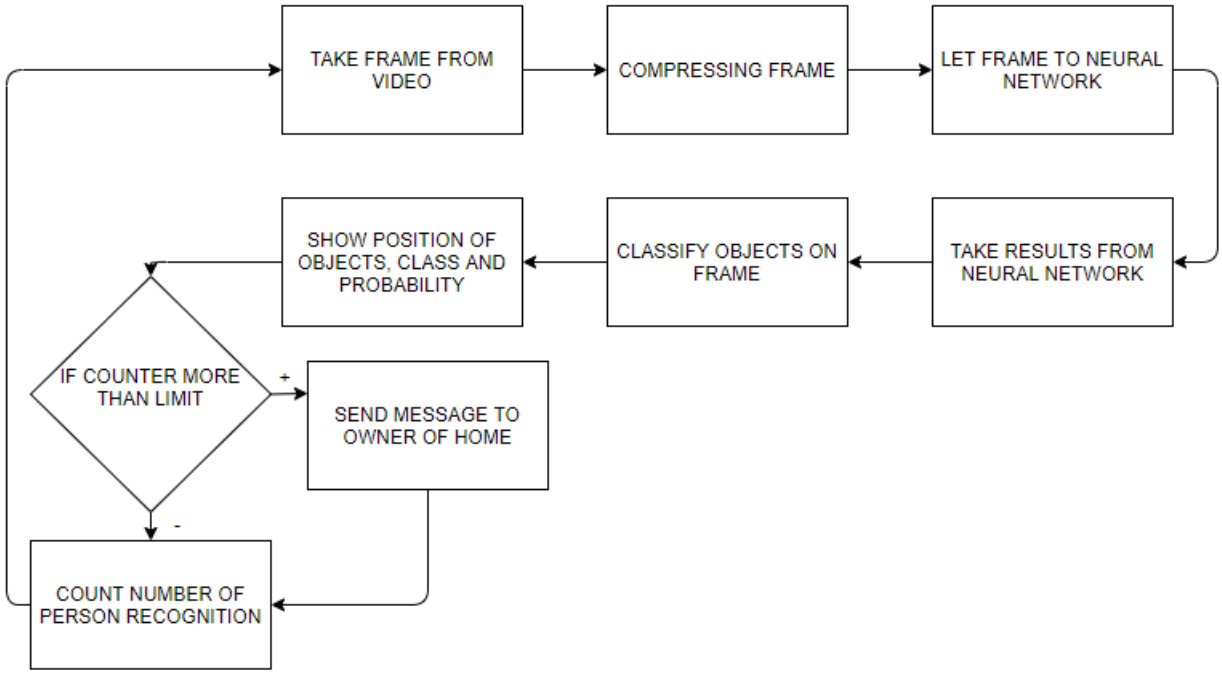


Рисунок 2.1 – Модель для розпізнавання рухів об’єктів на вході в домівку

Програма запуститься, включить камеру і, в свою чергу, почне підбір кадрів. Ці кадри надсилаються в буфер, і кожен кадр зберігається для подальшої обробки. Щоб скоротити час обробки, кадр стискається до меншого розміру, а основний піксель – це середнє значення стиснених пікселів. Потім зображення розширюється в одновимірний масив і надсилається в нейронну мережу. Нейронна мережа намагається розпізнати об’єкт і повертає деякі значення при розпізнаванні об’єкта. Ці значення використовуються для визначення місця розташування об’єкта, його класифікації та ймовірності правильної класифікації. Як тільки програма отримує мережеві результати, вона поміщає ці результати в кадр, який зберігається в буфері пам’яті. Потім система витягує готовий кадр і поступово підраховує кількість кадрів, на яких розпізнано обличчя людини. Наступним кроком буде перевірка кількості підрахованих квадратів. Коли кількість кадрів перевищує певну межу, система надсилає власнику домівки повідомлення про вторгнення в будинок. Потім система перезапускає цикл.

2.3 Проектування алгоритму для розпізнавання

Згідно моделі системи розпізнавання розробимо алгоритм програми розпізнавання.

Модель системи – це графічне представлення, яке використовується для візуалізації, специфікації, конструювання та документування компонентів програмного забезпечення. В контексті програми розпізнавання об’єктів на вході в домівку модель може включати кілька діаграм, які допомагають детально описати структуру та поведінку системи (рис. 2.2).

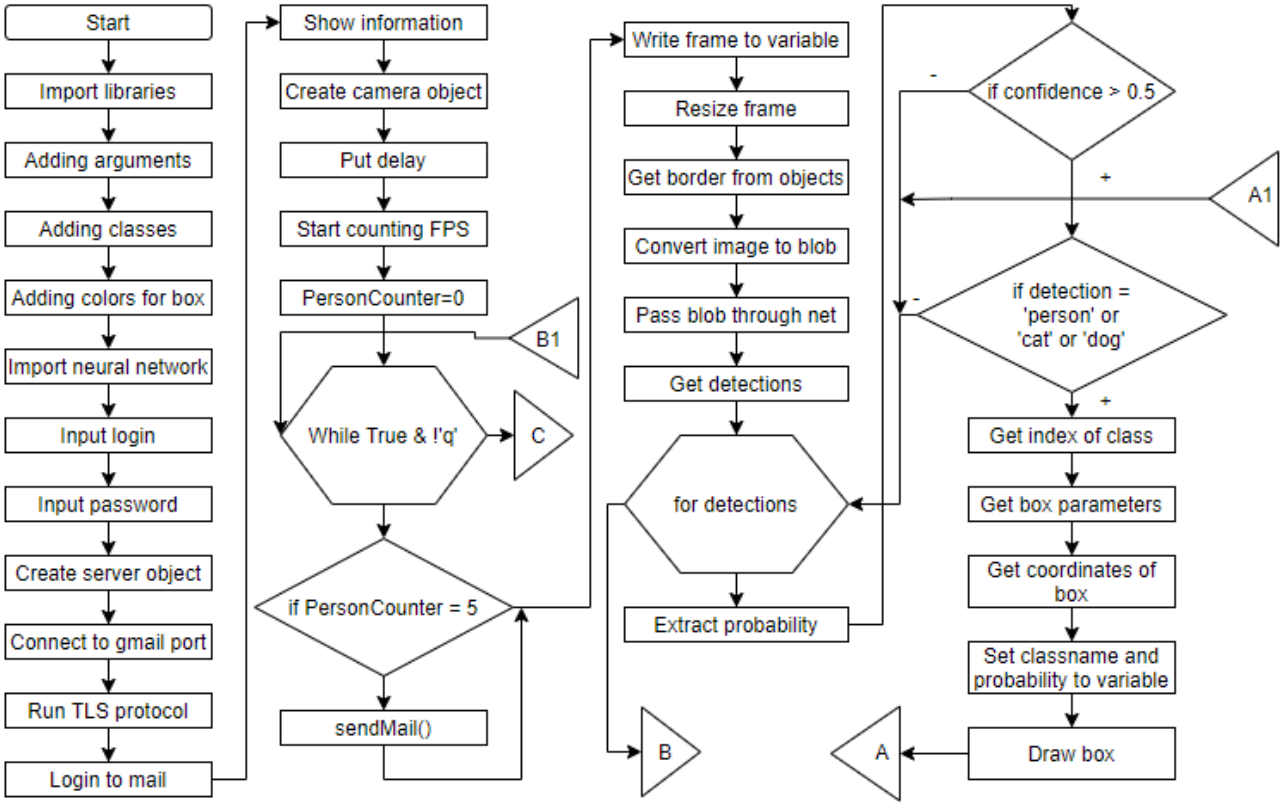


Рисунок 2.2 – Основна частина алгоритму розпізнавання

На рисунку 2.3 наведено продовження алгоритму розпізнавання, а саме напрямків A, B, C,

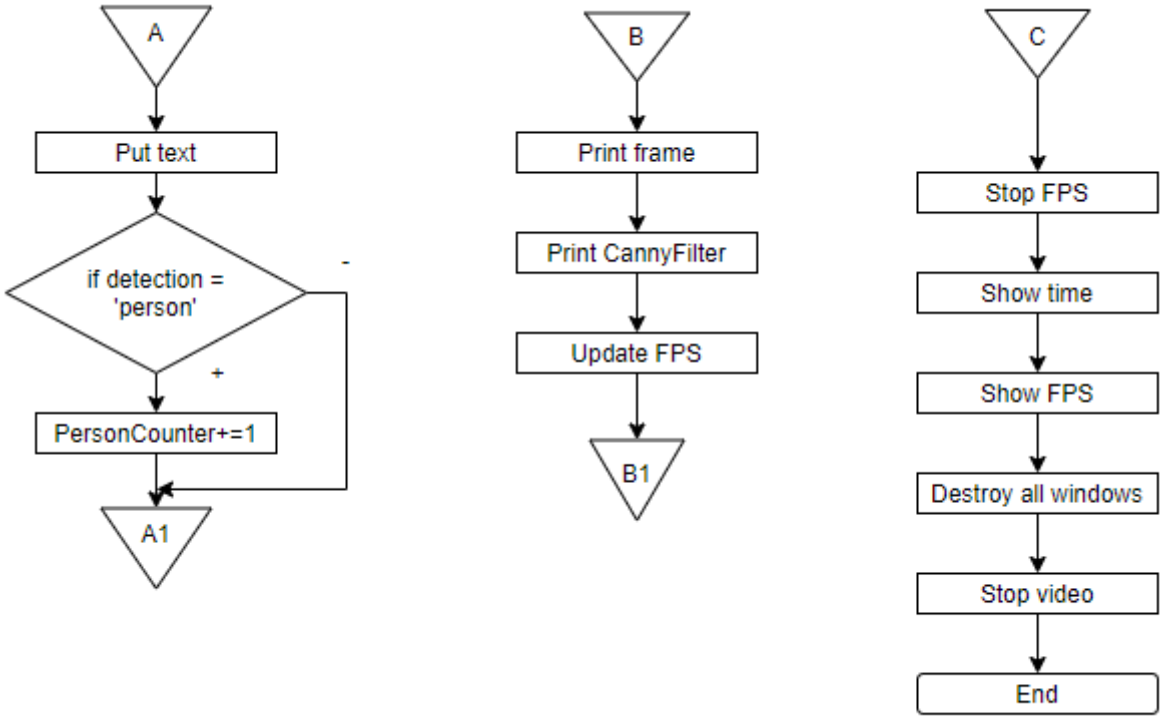


Рисунок 2.3 – Друга частина алгоритму розпізнавання

Алгоритм можна розділити на 3 частини: ініціалізація, основний цикл і завершення.

На початковому етапі система запускається шляхом підключення всіх додаткових бібліотек і утиліт, необхідних для роботи програми. Потім програма починає активувати аргументи, які використовуються для вибору майбутньої нейронної мережі, її прототипу та коефіцієнтів ймовірності. Далі програма додає клас нейронної мережі до змінної, що представляє масив, і колір блоку для виділення об’єктів у фреймі. Програма додає клас нейронної мережі до змінної, що представляє масив, і колір блоку для виділення об’єктів у фреймі. Потім одним з двох способів підключається нейронна мережа: ввести ім’я файлу в коді або створити змінну, яка дозволить ввести подібний тип нейронної мережі через інтерфейс. Наступним кроком є запуск системи сповіщень, тому спочатку в змінну користувача вводиться логін електронної пошти, а потім пароль. Потім створюється об’єкт сервера, який повинен відповідати за вхід у пошту і зв’язування повідомлення. По-перше, об’єкт підключається до служби Gmail за допомогою

одного з портів для відправки повідомлення, і важливо, щоб пошта в системному повідомленні мала доступ до інших додатків, оскільки викликається протокол безпеки TLS (Transport Layer Security) і виконується вхід у систему. Потім система повідомить про включення камери і створить об’єкт, який відповідає за роботу камери. Наступним кроком у системі є затримка, яка допомагає увімкнути камеру до того, як система почне знімати кадр. І далі, система повідомить про початок підрахунку кадрів і почне підрахунок. Система також визначає змінні лічильників людей у структурі, які необхідні для ідентифікації облич та інформування домовласника про вторгнення та завершує фазу ініціалізації й запускає фазу основного циклу.

Основний цикл задається умовою "поки правильний", і якщо цикл не переривається ззовні або оператором "break", цикл завжди перевіряє кількість кадрів, в яких об’єкт спочатку позначений як людина. Коли вони перевищують певну межу, система повідомляє власника про вторгнення. Ця перевірка виконується кожного разу при запуску циклу. Потім система вибирає кадр із потоку та зберігає його у змінній. Розмір цього фрейму змінюється і перетворюється в двійковий файл. Паралельно кадр обробляється оператором Кенні для отримання контуру об’єкта в кадрі. Двійковий код зображення проходить через нейронну мережу і вона повертає результати у вигляді числових властивостей, що використовуються для класифікації та організації об’єктів. Потім програма виконує ще один цикл, який перевіряє числові властивості, надані мережею. Спочатку віднімаємо ймовірність класу, потім перевіряємо надійність мережі і, якщо надійність перевищує певну межу, перевіряємо клас, який нам потрібен. Якщо у кадрі присутня людина або домашня тварина, система вибирає індекс необхідного класу, ініціалізує компоненти для кожного з цих об’єктів і вибирає імена класів, ймовірності та розташування у фреймворку. Класи та ймовірності записуються у змінні, блоки наносяться на координати об’єкта та додаються змінні з класами та ймовірностями. Потім система обчислює, чи є в кадрі об’єкти людського класу. Якщо є контакт, змінна, яка є лічильником кадрів з цим контактом, збільшується на 1. Потім цикл управління кадром триває до тих пір, поки всі об’єкти в кадрі не

будуть знайдені. Якщо об’єктів не залишилося, програма малює оброблені кадри, відображає ескіз і оновлює кількість кадрів. Після цього програма повертається в основний цикл. Якщо під час циклу натиснута клавіша виходу, програма завершує останній шлях циклу і переходить до фази завершення.

На етапі завершення програма припиняє підрахунок кадрів, показуючи час виконання і кількість кадрів. Після цього додаток закриває всі вікна в кадрі, зупиняє відео і виходить з програми.

2.4 Висновок до розділу 2

Визначено основну задачу – розробити модель, яка здатна ефективно розпізнавати обличчя на вході в домівку. Було окреслено ключові вимоги до системи, включаючи високу точність розпізнавання та мінімізацію помилкових спрацьовувань. Визначення задачі та постановка цілей створили основу для подальшого проектування та розробки моделі.

Розроблена модель для розпізнавання рухів, яка візуально представляє структуру та взаємодію компонентів системи розпізнавання. Модель включає діаграми випадків використання, класів, послідовності, діяльності та компонентів, що допомагає чітко уявити, як система повинна функціонувати. Висновком є створення детальної візуалізації системи, яка сприяє розумінню та впровадженню проекту.

Розроблено алгоритм роботи системи розпізнавання. Алгоритм описує послідовність дій від захоплення відеопотоку до відправки сповіщень власнику у разі виявлення людини. Включає кроки з обробки кадрів, аналізу результатів та фільтрації помилкових спрацьовувань. Висновок полягає в тому, що розроблений алгоритм забезпечує ефективну та надійну роботу системи розпізнавання рухів на вході в домівку.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АЛГОРИТМУ РОЗПІЗНАВАННЯ ОБЛИЧ ЛЮДЕЙ НА ВХОДІ В ДОМІВКУ

3.1 Програмні засоби

Для аналізу мов програмування були взяті Java, C++ та Python. Java – мова об’єктно-орієнтованого програмування, яка відома своєю популярністю і широким застосуванням у різних галузях, таких як фінанси, бази даних (Big Data), мобільні додатки. Вона є мультиплатформерною, що в свою чергу означає, програми, які написані на Java, можуть працювати на різних операційних системах. Деякі переваги Java включають мультиплатформність, безпеку, можливість роботи з багатьма потоками, автоматичне керування пам’яттю, а також відмінність для розробки корпоративних додатків. Також Java має активну спільноту користувачів та розробників.

Основні переваги Java включають:

1. Програми, написані на Java, можуть запускатися на різних операційних системах без змін вихідного коду;
2. Використовує систему безпеки, яка дозволяє створювати безпечні програми, запобігаючи вразливостям і атакам зловмисників;
3. Має вбудовану підтримку для роботи з багатьма потоками одночасно, що полегшує розробку паралельних програм;
4. Використовує систему, яка автоматично видаляє непотрібні об’єкти з пам’яті, що спрощує управління пам’яттю;
5. Має велику кількість бібліотек і фреймворків, які полегшують розробку великих корпоративних програм;
6. Має велику та активну спільноту, що сприяє обміну досвідом та розвитку екосистеми.

Загальними недоліками Java є:

1. У деяких випадках може бути менш ефективною за швидкодією, особливо в областях, де вимагається висока продуктивність;

2. Відома своїм великим споживанням пам’яті та іншими ресурсами, що може бути проблемою в обмежених середовищах, наприклад, на вбудованих системах або пристроях з обмеженими ресурсами;

3. Існують можливості для вразливостей, особливо при використанні старих версій Java Runtime Environment (JRE). Крім того, обмеженість управління пам’яттю може призвести до витоку пам’яті або інших проблем з ресурсами;

4. Для виконання програм на Java потрібно мати встановлену віртуальну машину Java (JVM), що може призвести до збільшення розміру програми і вимагати додаткових обчислювальних ресурсів для виконання;

5. Не є ідеальним вибором для розробки низькорівневих або вбудованих систем, оскільки вона відокремлена від апаратного забезпечення та не надає прямого доступу до апаратних ресурсів.

Останнім часом мова програмування C++ втрачає свою популярність у сфері об’єктно-орієнтованого програмування, але залишається популярною для розробки системних, прикладних програм та ігор.

Деякі переваги C++ включають широке використання, можливість роботи на низькому рівні для оптимізації, універсальність та високу швидкодію. Однак мова має свої недоліки, такі як складність у навчанні, можливість працювати на низькому рівні, що може бути проблемою для новачків, погана сумісність з веб-додатками та складність пошуку помилок.

Мова програмування Python здобула значну популярність завдяки своїй простоті та зручності. Весь синтаксис розроблений таким чином, щоб бути зрозумілим як для програмістів, так і для користувачів.

Основні переваги мови програмування Python включають:

1. Простий та зрозумілий синтаксис;

2. Вивчення мови легке та швидке;

3. Універсальна мова програмування;

4. Багато бібліотек та активна спільнота;

5. Мультиплатформений та переносний код;

6. Популярна у наукових дослідженнях;

7. Високо цінується в AI-розробках.

Основними недоліками мови програмування Python є:

1. Низька швидкодія в порівнянні з іншими мовами;

2. Проблеми з компіляцією та виконанням;

3. Неідіоматичний код може викликати проблеми;

4. Перенесення коду може бути складним;

5. Глобальний інтерпретатор (GIL) обмежує паралельність;

6. Відступи обов’язкові та важко відслідковувати.

Після вивчення кількох мов зупинився на мові Python через її простоту використання та велику базу бібліотек, що полегшує написання програм, пов’язаних зі штучним інтелектом.

Наступним кроком є вибір бібліотек, які будуть використані під час написання програми розпізнавання. Спочатку важливо звернути увагу на бібліотеки, які зосереджені на зображеннях, зокрема на комп’ютерному зорі. Серед бібліотек найпопулярнішою є бібліотека OpenCV.

OpenCV – це бібліотека комп’ютерного зору, яка використовується в C++. Інтегровано в такі мови програмування, як Java, Python, Ruby, Matlab та інші. OpenCV надає засоби для отримання даних з камер або попередньо створених фільмів, обробки даних і використання їх для розширення іншої обробки в інших бібліотеках, включаючи модулі для взаємодії з нейронними мережами.

OpenCV не має математичної бібліотеки, тому їй потрібна NumPy, вони залежать один від одного. NumPy – це бібліотека програмування, яка полегшує роботу з математичними рівняннями, функціями та графіками. Вона використовується в різних наукових дослідженнях. Основною властивістю є висока швидкість при роботі з багатовимірними масивами на рівні мови MATLAB.

Imutils – додаткова бібліотека для OpenCV, яка полегшує просте впровадження таких функцій, як зміна розміру, обертання зображення,

трансформація, побудова графіків тощо. Ця бібліотека надає прості засоби для керування функціями OpenCV із більш лаконічним синтаксисом.

Smtplib – поширена бібліотека Python, яка полегшує створення посилань на повідомлення електронної пошти. ArgParse – поширена бібліотека Python, яка полегшує конфігурацію спеціальних аргументів, необхідних для виконання компонентів програми. Time – звичайна бібліотека Python, яка використовується для керування часом у програмах.

3.2 Розробка структури програмного забезпечення

Відповідно до запропонованої моделі розроблено загальну структуру програмного забезпечення. Структура роботи програмного забезпечення наведена на рисунку 3.1.

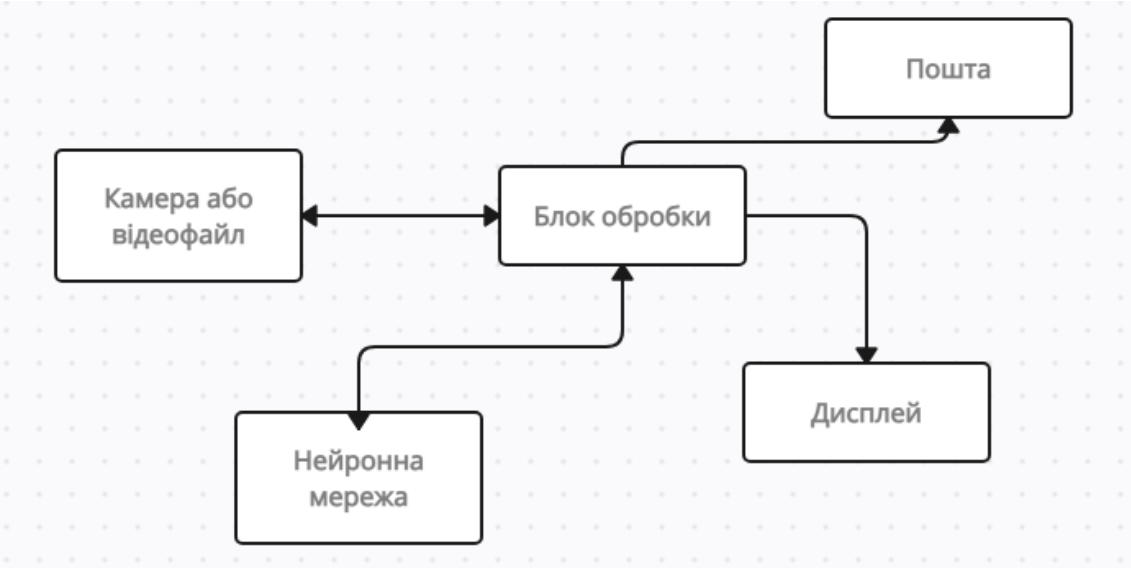


Рисунок 3.1 – Структура роботи програмного забезпечення

Відповідно до розробленої структури, після початкового налаштування даних система запитуватиме про переміщення поточного кадру, програма обробить запит і надасть дані в нейронну мережу. Отримана відповідь від нейронної мережі означає, що новий кадр створено і передано на дисплей користувача. Коли система

знаходить у кадрі об’єкт людини, вона перевіряє кілька наступних кадрів на наявність обличчя. Якщо об’єкт присутній, це передбачає вторгнення в домівку і про це надсилається повідомлення.

Після вивчення різних методів розпізнавання руху, нейронні мережі виявилися найбільш ефективними.

Наступним кроком є вибір способу реалізації використання нейронних мереж у програмі: створити власну мережу з нуля чи знайти вже навчену мережу. Для системи розпізнавання облич є заздалегідь розроблені навчені мережі, які мають високий ступінь точності розпізнавання, тому вони відповідають поставленим завданням.

Для програми призначена загальнодоступна нейронна мережа MobileNet_SSD, створена за допомогою Caffe. Дане середовище написано мовою C++, що є перевагою для сумісності з бібліотекою комп’ютерного зору OpenCV.

Нейронна мережа MobileNet_SSD здатна розпізнавати 21 різний об’єкт, включаючи літаки, велосипеди, птахів, човни, машини, котів, стільці, корів, столи, собак, коней і телевізор.

Для виконання завдання потрібні лише клас обличчя. Щоб проігнорувати інші об’єкти, система буде оцінювати класи і виконуватиме дії лише над необхідним класом, обраним у завданні.

3.3 Розробка програмного забезпечення для роботи алгоритму

Згідно алгоритму розробки, створено код програмного забезпечення. Код складається з трьох етапів: ініціалізації, основний циклу та завершення. Ініціалізаційний етап розпочинається з підключення всіх необхідних бібліотек для роботи. Однак не всі ці бібліотеки є стандартними та входять до складу середовища Python. Тому розглянуто процес встановлення зовнішніх модулів, необхідних для роботи програми. Для встановлення більшості бібліотек Python є можливість скористатися утилітою PyPi. Ця утиліта надає можливість встановлювати,

оновлювати, видаляти та керувати зовнішніми бібліотеками. Використаємо PyPi для встановлення OpenCV. Нижче подані зображення, на яких показано поетапний процес встановлення бібліотеки (рис. 3.2–3.3).

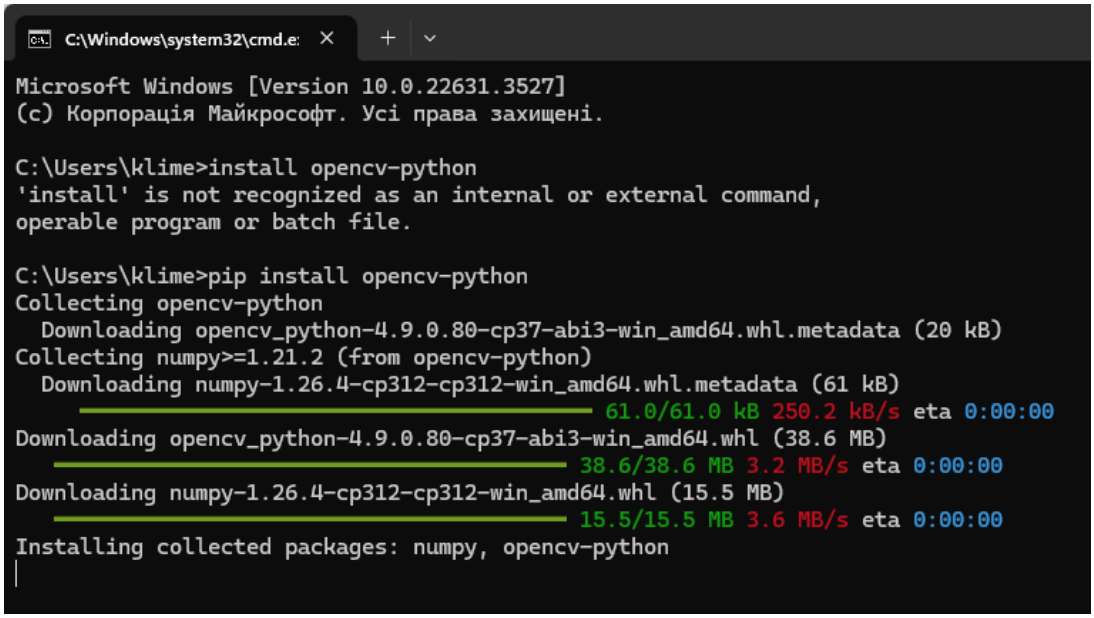


Рисунок 3.2 – Введення команди pip для встановлення бібліотеки OpenCV

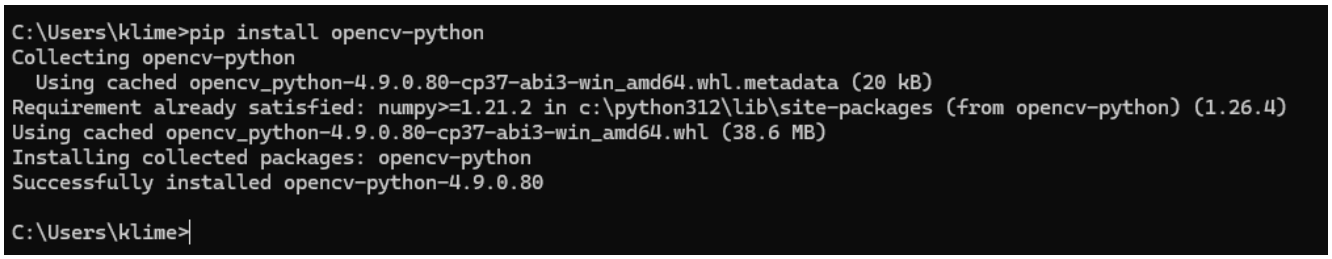


Рисунок 3.3 – Успішна установка бібліотеки OpenCV

Як можна помітити, з OpenCV автоматично встановлюється NumPy. Встановимо за тим же принципом інші необхідні бібліотеки: Imutils, ArgParse (рис. 3.4–3.7).

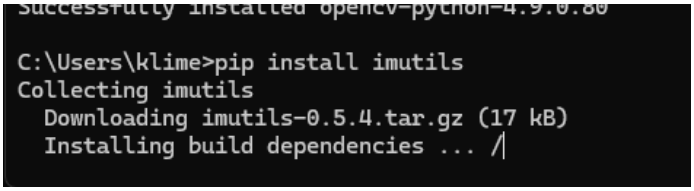


Рисунок 3.4 – Виконання встановлення бібліотеки Imutils

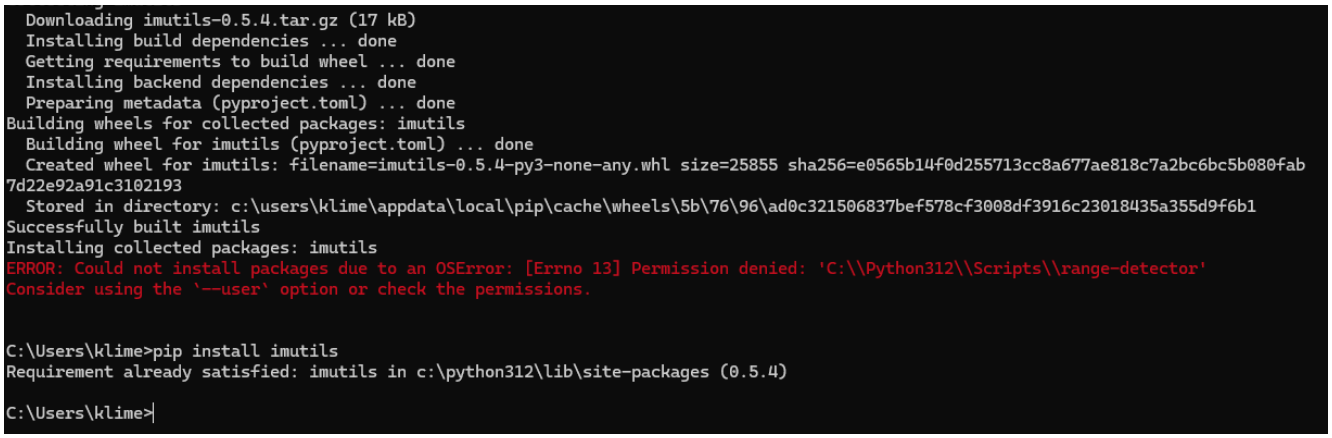


Рисунок 3.5 – Успішне встановлення бібліотеки Imutils

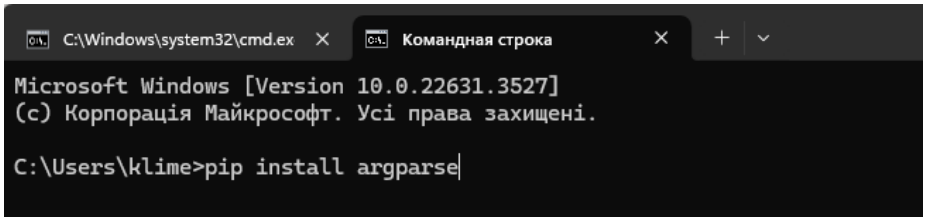


Рисунок 3.6 – Виконання встановлення бібліотеки ArgParse

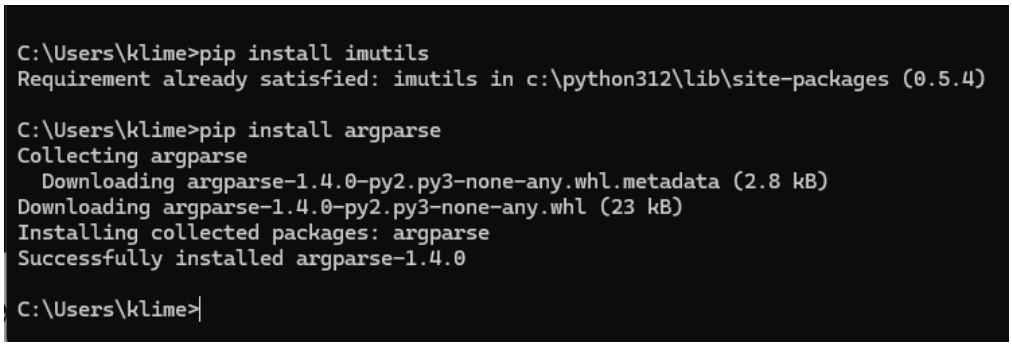


Рисунок 3.7 – Успішне встановлення бібліотеки ArgParse

Після установки усіх необхідних зовнішніх бібліотек можна приступати до написання коду.

Для підключення бібліотек у мові Python використовується ключове слово "import" (рис. 3.8). Синтаксис: "import <назва бібліотеки>". Це дозволяє включити весь код бібліотеки до поточної програми. У випадку, коли бібліотека є об’ємною, а програма використовує лише кілька функцій, можна скористатися такою конструкцією: "from <назва бібліотеки> import <назва функції>".

```
from immutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import time
import cv2
import smtplib as sl
```

Рисунок 3.8 – Підключення бібліотек для роботи програми

Це дозволить уникнути завантаження усієї бібліотеки, зменшуючи обсяг програми для користувача. Деякі модулі мають довгі назви або можуть бути складними для сприйняття.

У такому випадку можна використати конструкцію "import <назва бібліотеки> as <скорочена назва>", щоб уникати повного написання назви при кожному виклику об’єктів бібліотеки. На рисунку 3.8 показано, що для роботи з відеопотоком буде використаний модуль Imutils з класами VideoStream та FPS.

Далі проводиться підключення модулю NumPy, необхідного для коректної роботи модулю OpenCV. Наступною бібліотекою є ArgParse, яка дозволяє налаштовувати користувацькі аргументи, у даному випадку – аргументи для підключення нейронної мережі та налаштування коефіцієнту впевненості системи.

Далі йде модуль Time, необхідний для створення затримки програми під час запуску об’єкта VideoStream для камери чи готового відео.

Наступним кроком є підключення OpenCV, яке буде використано для завантаження нейронної мережі та обробки кадрів.

Для оповіщень планується використати Smtplib, тому проводиться його підключення для подальшої роботи з системою оповіщень.

Підключення бібліотек завершено. Наступним етапом буде створення та налаштування користувацьких аргументів для системи.

На рисунку 3.9 наведено код створення аргументів.

```
argumparse = argparse.ArgumentParser()
argument.add_argument("-p", "--prototxt",
    required=True,
    help="path to Caffe 'deploy' prototxt
        file")
argument.add_argument("-m", "--model",
    reauired=True,
    help="path to Caffe pre-trained model")
argument.add_argument("-c", "--confidence",
    type=float, default=0.2,
    help="minimum probability to filter weak
        detections")
args = vars(argumparse.parse_args())
```

Рисунок 3.9 – Створення та налаштування вхідних аргументів програми

На цьому рисунку можна побачити процес створення об’єкту класу ArgumentParser з іменем argumparse. За допомогою даного об’єкта реалізується додавання трьох аргументів: аргумент введення прототексту нейронної мережі, аргумент введення моделі нейронної мережі та аргумент введення коефіцієнту впевненості системи. Після цього ці аргументи заносяться до списку, з якого вони будуть викликані кожен у свій час виконання.

Далі починається конфігурування програми під налаштування нейронної мережі, а саме числові параметри, які ця мережа повертає. Нижче наведений фрагмент коду, що відповідає за присвоєння класам мережі та надання їм колірної палітри (рис. 3.10).

```
19  CLASSES = ["backgroung", "aeroplane", "bicycle",
        "bird", "boat", "bottle", "bus", "car", "cat",
        "chair", "cow", "digningtable", "dog", "horse",
        "motorbike", "person", "pottedplant", "sheep",
        "sofa", "rtain", "tvmonitor"]
20  COLORS = [[0,0,0], [0,0,0], [0,0,0], [0,0,0], [0,0
        ,0], [0,0,0], [0,0,0], [255,0,0], [0,0
        ,0], [0,0,0], [0,0,0], [0,255,0], [0,0,0], [0,0
        ,0], [0,0,255], [,0,0,0], [,0,0,0], [,0,0,0], [
        ,0,0,0], [,0,0,0]]
```

Рисунок 3.10 – Додавання класів нейронної мережі та кольору класів

На цьому фрагменті додаються два списки, класів та кольору класів. Вони будуть потрібні для обробки результатів роботи нейронної мережі.

Далі підключається нейронна мережа (рис. 3.11).

```
21  print("Message: (important) Loading of neural
        network model.")
22  net = cv2.dnn.readNetFromCaffe(args["prototxt"],
        args["model"])
```

Рисунок 3.11 – Підключення нейронної мережі

На даному етапі система повідомляє про успішний запуск нейронної мережі, використовуючи бібліотеку OpenCV для створення об’єкту моделі та передачі необхідних аргументів. Цей процес показує, наскільки добре модуль комп’ютерного зору інтегрується з середовищем Caffe, відобразивши високу сумісність обох компонентів.

Далі виконується ініціалізація об’єктів для забезпечення з’єднання з поштовою скринькою. Наведений на рисунку 3.12 наочний приклад коду ілюструє процес ініціалізації, надаючи більше контексту щодо кроку, який виконується.

```
23  maillogin = 'diploma.recognition.bot@gmail.com'
24  mailpass = 'RdzO6vqIMvD'
25  so=sl.SMTP('smtp.gmail.com', 587)
```

Рисунок 3.12 – Ініціалізація об’єктів для використання пошти

На цьому етапі програма відображає процес створення змінних для логіну та паролю пошти системи, а також формує об’єкт для взаємодії з системою повідомлень. Для цього вона створює хост сервісу Gmail з окремим портом, який буде використовуватися для відправлення повідомлень. Після цього система починає встановлення з’єднання з поштою.

На рисунку 3.13 наведено фрагмент коду, який відповідає за процедуру авторизації.

```
26  so.starttls()
27  so.login(maillogin, mailpass)
```

Рисунок 3.13 – Авторизація до поштової скриньки

У вказаному фрагменті можна відзначити, що для успішного встановлення зв’язку з електронною поштою система повинна спочатку створити захищене з’єднання, а потім пройти процедуру авторизації (рис. 3.14). Важливою відміткою є те, що програма не зможе отримати доступ до вашої пошти до тих пір, поки ви не надасте дозвіл на використання пошти небезпечними додатками. Цей дозвіл можна надати через розділ "Ненадійні додатки".

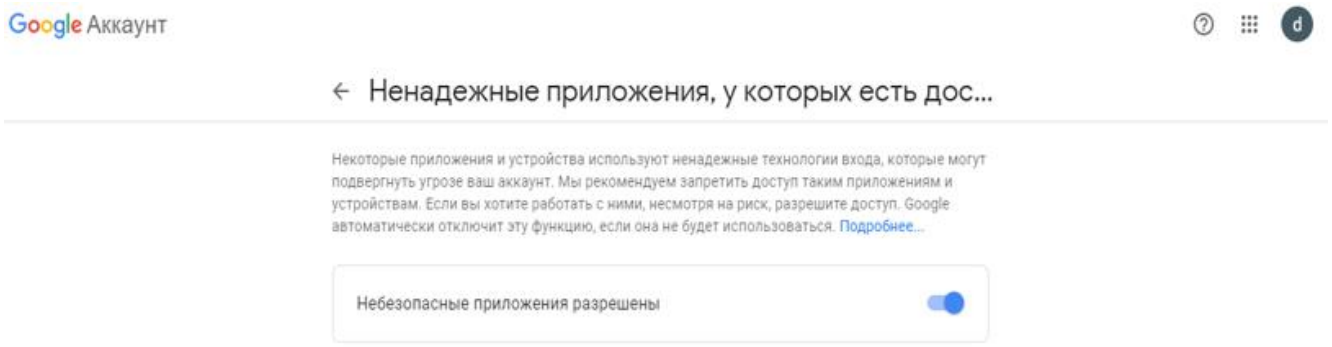


Рисунок 3.14 – Дозвіл на використання пошти у ненадійних додатках

Далі програма починає запуск камери або завантажує готове відео (рис. 3.15).

```
28  print("Message: (important) Switch on your camera.")
29  StreamObj = VideoStream(src=0).start()
30  time.sleep(2.0)
31  fps = FPS().start()
```

Рисунок 3.15 – Ініціалізація камери або відео

У цьому фрагменті можна побачити, як програма сповіщає про вмикання камери або відео та подальші дії. Спочатку створюється об’єкт, який відповідає за роботу камери. Після визначення джерела відеопотоку, цей об’єкт запускає камеру, після чого програма призупиняється, камера завантажується та розпочинає свою роботу. Далі створюється окремий об’єкт для підрахунку кадрів. І на останньому етапі ініціалізації проводиться створення змінної лічильника, необхідної для підрахунку кількості кадрів зображень з людьми (рис. 3.16).

```
personCounter = 0
```

Рисунок 3.16 – Ініціалізація лічильника

Наступний етап – етап основного циклу. В ньому відбувається основна робота програми. Починається він із визначення циклу (рис. 3.17).

```
while True:
```

Рисунок 3.17 – Початок безкінечного циклу

Так як час виконання програми є необмеженим, для цього використовується безкінечний цикл, який буде припинятися при натисканні кнопки виходу. Після цього проводиться перевірка лічильника на досягнення ліміту. У випадку перевищення допустимого значення лічильника система автоматично повідомляє власника про можливе вторгнення. Поданий на рисунку 3.18 фрагмент коду ілюструє цей процес.

```
if personCounter == 5;
    so.sendmail(maillogin, "gameforvdmvoid@gmail
        .com", "I see the Person on your camera.")
```

Рисунок 3.18 – Перевірка на ліміт кадрів з об’єктами типу “person”

З фрагменту видно, що спочатку відбувається повторний вхід у систему електронної пошти, а потім вхід до самої поштової скриньки користувача. Наступним кроком є написання повідомлення. Крім тексту, тепер можна модернізувати лист, додаючи файли та зображення. Після процедури перевірки настає етап вилучення вкладення та його обробка (рис. 3.19).

```
36  videoframe = StreamObj.read()
37  videoframe = imutils.resize(videoframe, width=400)
38  cannyfilter = cv2.Canny(videoframe, 100, 200)
```

Рисунок 3.19 – Витягування кадру та його обробка

Спершу кадр подається в об’єкт videoframe, після чого його розмір стискається до 400 пікселів, і потім з нього окремо виділяються контури об’єктів. Різниця між останніми двома числами функції Canny() вказує на кількість контурів, які будуть враховані. Чим менше ця різниця, тим більше деталей буде виділено.

Потім оброблений кадр конвертується в бінарний об’єкт, який буде переданий через нейронну мережу. На рисунку 3.20 наведено фрагмент коду.

```
(h, w) = videoframe.shape[:2]
binaryvar = cv2.dnn.blobFromImage(cv2.resize
    (videoframe, (300, 300)), 0.007843, (300,300),
    127.5)
```

Рисунок 3.20 – Створення бінарного об’єкту для нейронної мережі

На цьому зображенні можна побачити процес, за якого система отримує розміри кадру для майбутнього використання. Після цього вона створює бінарний об’єкт за допомогою бібліотеки OpenCV, який представляє собою стисле перетворене зображення. Подальше використання цього об’єкта передбачає проходження через нейромережу, яка повертає числові характеристики. Фрагмент коду, пов’язаний з цим процесом, подається на рисунку 3.21.

```
net.setInput(binaryvar)
detections = net.forward()
```

Рисунок 3.21 – Процес розпізнавання

У розглянутому фрагменті коду демонструється процес передачі об’єкту бінарного зображення до мережі. Після отримання вхідних даних, мережа автоматично виконує внутрішні обчислення для обробки зображення та повертає оновлений результат. Для збереження цього результату зазвичай створюється змінна під назвою "detections", до якої потім надходять вихідні дані з мережі.

Потім система починає новий цикл. Він працює до тих пір, доки програма не перевірить усі розпізнані об’єкти (рис. 3.22).

```
for i in np.arange(0, detections.shape[2]):
    confidence = detections[0, 0, i, 2]
```

Рисунок 3.22 – Цикл перевірки розпізнаних об’єктів

У даному фрагменті коду відображено створення циклу, який продовжує виконуватися, доки наявні об’єкти. Далі в змінну "confidence" записується ймовірність поточного класу об’єкту. Після цього відбувається перевірка: якщо ймовірність перевищує певний поріг, у даному випадку 0.5, то програма не пропустить об’єкт і перейде до наступного. На рисунку 3.23 наведено сам фрагмент коду.

```
if confidence > args["confidence"]:
```

Рисунок 3.23 – Перевірка вірогідності об’єкта

Якщо вірогідність вище, то об’єкт обробляється в наступному кроці. Фрагмент коду наведений на рисунку 3.24.

```
    if CLASSES[int(detections[0, 0, i, 1])] ==
        'person' or CLASSES[int(detections[0, 0, i,
        1])] == 'cat'\
or CLASSES[int(detections[0, 0, i, 1])] == 'dog':
    classIndex = int(detections[0, 0, i, 1])
    box = detections[0, 0, i, 3:7] * np.array([w, h,
        w, h])
    (startX, startY, endX, endY) = box.astype("int")
```

Рисунок 3.24 – Визначення потрібного класу

З наведеного фрагменту коду можна зрозуміти, що система спочатку проводить відбір потрібних класів серед визначених об’єктів для програми. Якщо ж жоден з класів не відповідає критеріям, програма розпочинає цикл заново. У випадку знаходження потрібного класу, програма отримує його індекс та створює

блок з координат, одержаних з мережі. Отримані координати блоків зберігаються у вигляді кортежу.

Далі, наступним кроком є візуалізація блоків та пов’язана з ними інформація про об’єкти. На рисунку 3.25 подано фрагмент коду для реалізації даного процесу.

```
label = "{}: {:.2f}%".format(CLASSES[classIndex],
    confidence * 100)
    cv2.rectangle(videoframe, (startX, startY),
        (endX, endY),
        COLORS[classIndex], 2)
    y = startY - 15 if startY - 15 > 15 else startY
        + 15
    cv2.putText(videogame, label, (startX, y),
    cv2.FONT_HERSHEY_SIMPLEX, 0.5,
        COLORS[classIndex], 2)
```

Рисунок 3.25 – Малювання блоків об’єктів

Далі докладніше розглянемо цей фрагмент. По-перше, програма створює текстову змінну, яка містить назву класу та ймовірність цього класу. Потім йде процес малювання самого блоку. Перший аргумент videoframe вказує, що блок малюється на кадрі. Далі слідують координати об’єкта, після чого використовується колір блоку відповідно до поточного класу. Далі вказується товщина лінії. Потім визначається висота позиції, де потрібно відобразити інформацію, збережену в змінній label. Після визначення позиції текст змінної вставляється на кадр, набираючи свої налаштування шрифту, кольору та розміру.

Наступним етапом є визначення об’єктів на кадрі. Якщо система виявляє людей на ньому, то лічильник інкрементується. На рисунку 3.26 наведений фрагмент коду для детальнішого огляду.

```
if CLASSES[int(detections[0, 0, i, 1])] ==
    'person':
    personCounter += 1
```

Рисунок 3.26 – Визначення об’єкту “person”

Далі програма виводить результати. Виводяться два кадри: оброблений за допомогою блоків та оброблений за допомогою оператора Канні. Фрагмент коду наведений на рисунку 3.27.

```
cv2.imshow("VideoFrame", videoframe)
cd2.imshow("CannyFilter", cannyfilter)
```

Рисунок 3.27 – Вивід оброблених кадрів

Останні кроки програми, це перевірка кнопки виходу та оновлення кадрів. Розглянемо фрагмент коду, що наведений на рисунку 3.28.

```
key = cv2.waitKey(1) & 0xFF
if key == ord("q"):
    break
fps.update()
```

Рисунок 3.28 – Перевірка програми на вихід

Програма реагує на всі введені команди з клавіатури. Якщо користувач натискає певну клавішу, наприклад, 'q', цикл виконує усі необхідні операції до того, як результати будуть виведені на екран, і потім завершується. У випадку, якщо жодну кнопку не натиснуто, програма оновлює кількість кадрів на секунду і починає цикл спочатку. Останній етап – завершення. Програма завершує підрахунок кадрів і виводить результат на екран. Потім програма закриває всі вікна і припиняє потік відео. На рисунку 3.29 наведений фрагмент коду.

```
fps.stop()
print("Message: (important) elapsed time: {
    :.2f}".format(fps.elapsed()))
print("Message: (important) approx. FPS: {:.2f}"
    .format(fps.fps()))
cv2.destroyAllWindows()
StreamObj.stop()
input('The end!')
```

Рисунок 3.29 – Кінець виконання алгоритму програми

3.4 Тестування алгоритму роботи програмного забезпечення

Після успішного написання коду настає час перейти до етапу тестування. Для запуску програми необхідно вказати обов'язкові аргументи нейронної мережі згідно з описом вище.

Спробуємо спочатку запустити програму без жодних аргументів. Для цього переходимо до командного рядка, відкриваємо папку з програмою та виконуємо запуск за допомогою наступної команди (рис. 3.30).

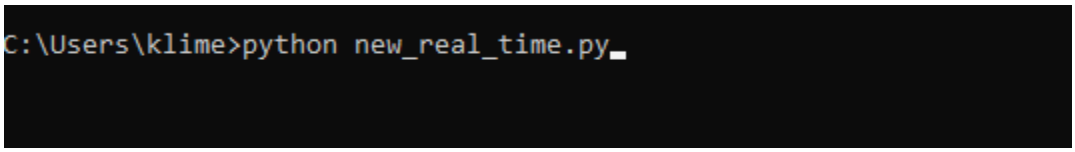


Рисунок 3.30 – Результат запуску програми без аргументів

Таким чином, переконалися що, програма не працює, якщо не додати потрібних аргументів. Тепер спробуємо запустити програму з ними (рис. 3.31).

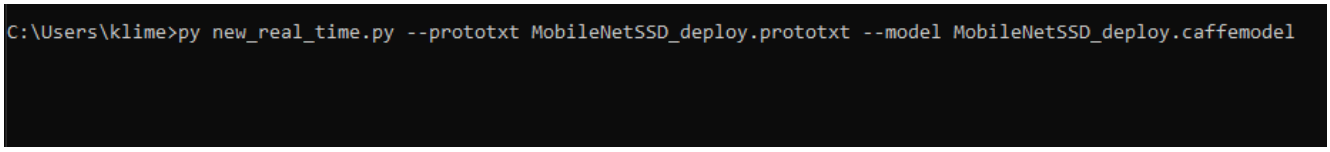


Рисунок 3.31 – Запуск програми з аргументами

Якщо програма запустилася, то бачимо приведений нижче результат (рис. 3.32).

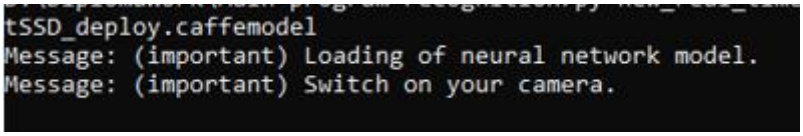


Рисунок 3.32 – Валідний результат роботи програми

Також побачимо, як камера увімкнеться та на екрані з'явиться оброблений кадр (рис. 3.33).

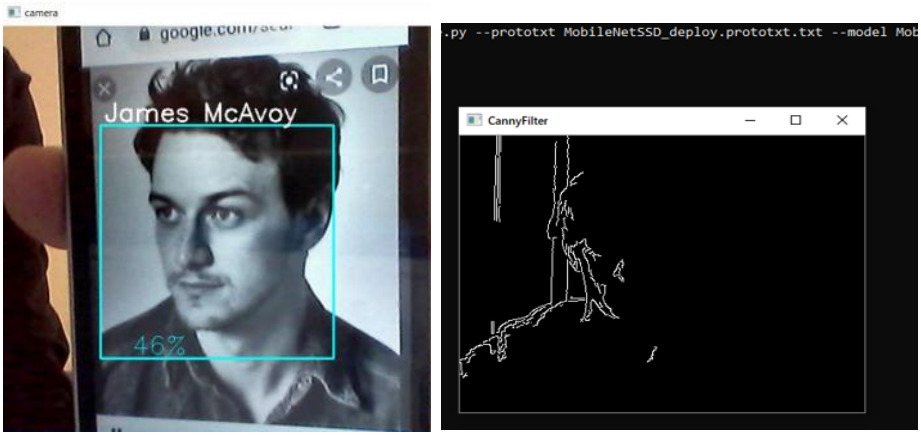


Рисунок 3.33 – Виведені на екран оброблені кадри

На знімку видно, що мережа розпізнає людину, причому вірогідність правильного розпізнавання, тобто впевненість, коливається у межах від 30 % до 99 %.

На рисунку 3.34 та 3.35 наведено фрагменти з повідомленням.

Рисунок 3.34 – Повідомлення від системи

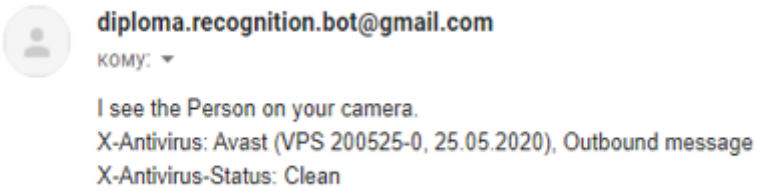


Рисунок 3.35 – Відкрите повідомлення

Таким чином, переконались, що програма працює без проблем, але має невеликі погрішності через особливості нейронних мереж.

Щоб вимкнути роботу програми, потрібно натиснути клавішу виходу. В даному випадку програма завершується після натискання клавіші ‘q’.

Як видно з рисунка 3.36, програма зупиняє роботу, кадри закриваються та на консоль виводиться час роботи та кількість кадрів за секунду. Після чого програма інформує про кінець роботи та зупиняється.

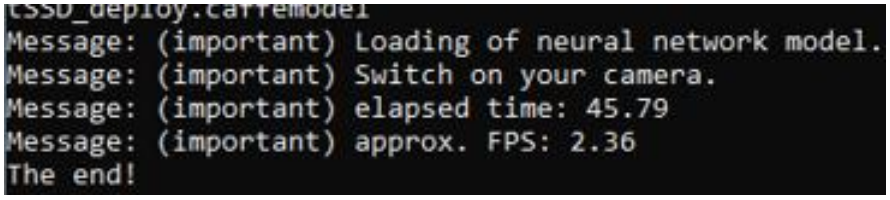


Рисунок 3.36 – Завершення роботи алгоритму програми

3.5 Висновок до розділу 3

У даному розділі проаналізовано процес вибору інструментів для розробки програмного забезпечення алгоритму розпізнавання облич при вході в домівку. Увага зосереджена на кількох ключових аспектах, таких як розробка програмного продукту та вибір програмних засобів, створення та навчання нейронних мереж, розробка структури програмного забезпечення, а також тестування алгоритму роботи програмного забезпечення.

Важливо підкреслити, що правильний вибір інструментів та методів розробки є вирішальним для успішної імплементації алгоритму розпізнавання контурів. Нейронні мережі, наприклад, дозволяють досягти високої точності у розпізнаванні об’єктів, але вони потребують складноощів при налаштуванні та навчанні. Однак, за допомогою правильної методології та інструментів розроблено програмне забезпечення, яке ефективно виконує поставлені завдання.

Розробка програмного забезпечення для роботи алгоритму та його подальше тестування є критичними етапами, що вимагають уважності та систематичного підходу. Лише завдяки цим етапам можна переконатися у правильності та ефективності алгоритму.

Таким чином, вибір інструментів та методів розробки програмного забезпечення для алгоритму розпізнавання облич вимагає комплексного підходу та

уваги до деталей. Правильно спроектоване програмне забезпечення відіграє ключову роль у досягненні успіху у даній області.

ВИСНОВКИ

Розглянуто сучасні методи та підходи до розпізнавання рухів об’єктів у відеопослідовностях. Було відзначено, що розвиток технологій відеоспостереження та комп’ютерного зору значно полегшив аналіз поведінки об’єктів у відео. Зокрема, зростаюча роль штучного інтелекту, зокрема нейронних мереж, дозволяє досягати високої точності розпізнавання та ідентифікації об’єктів.

Визначено основні виклики та проблеми, пов’язані з розпізнаванням рухів людей і тварин у відеопослідовностях. Серед основних проблем – високий рівень складності обробки великих обсягів відеоданих, варіативність умов зйомки (освітлення, кути огляду тощо), а також складність розпізнавання об’єктів у різних контекстах (наприклад, у натовпі або при частковому перекритті об’єктів).

Розгляд систем охорони показав, що сучасні системи відеоспостереження активно використовують методи розпізнавання рухів для підвищення ефективності безпеки. Впровадження автоматичних систем аналізу відеопотоків дозволяє швидше реагувати на потенційні загрози, забезпечувати моніторинг великих територій та мінімізувати людський фактор в охороні.

Розглянуто різні методи розпізнавання руху, такі як оптичний потік, фонове віднімання та використання нейронних мереж. Кожен з методів має свої переваги та недоліки, і вибір конкретного методу залежить від умов застосування та вимог до точності розпізнавання. Було відзначено, що комбінування методів може значно підвищити точність та надійність системи.

Розглянуто роль нейронних мереж у розпізнаванні об’єктів на зображеннях. Було розглянуто різні типи нейронних мереж, такі як згорткові нейронні мережі (CNN), які особливо ефективні для обробки зображень та відео. Використання нейронних мереж дозволяє досягати високої точності розпізнавання завдяки можливості навчання на великих обсягах даних та здатності до самовдосконалення.

Визначено основну задачу – розробити модель, яка здатна ефективно розпізнавати обличчя людини на вході в домівку. Було окреслено ключові вимоги

до системи, включаючи високу точність розпізнавання та мінімізацію помилкових спрацьовувань. Визначення задачі та постановка цілей створили основу для подальшого проектування та розробки моделі.

Розроблена базова модель для розпізнавання рухів. Було описано етапи створення моделі, включаючи збір даних, попередню обробку, вибір архітектури нейронної мережі, навчання моделі та її оптимізацію. Результатом є створена модель, яка забезпечує точне розпізнавання облич людини на вході в домівку.

Розроблено алгоритм роботи системи розпізнавання. Алгоритм описує послідовність дій від захоплення відеопотоку до відправки сповіщень власнику у разі виявлення людини. Включає кроки з обробки кадрів, аналізу результатів та фільтрації помилкових спрацьовувань. Розроблений алгоритм забезпечує ефективну та надійну роботу системи розпізнавання облич людини на вході в домівку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Zhou H., Niu X. An image processing algorithm for the measurement of multiphase bubbly flow using predictor-corrector method. International Journal of Multiphase Flow. 2020. Vol. 128. P. 103277. URL: https://doi.org/10.1016/j.ijmultiphaseflow.2020.103277.

2. Elharrouss O., Almaadeed N., Al-Maadeed S. A review of video surveillance systems. Journal of Visual Communication and Image Representation. 2021. Vol. 77. P. 103116. URL: https://doi.org/10.1016/j.jvcir.2021.103116

3. You can master Computer Vision, Deep Learning, and OpenCV. [електронний ресурс] : [Веб-сайт]. – Електроні дані. – 2009-2020. – Режим доступу: https://www.pyimagesearch.com/.

4. Gary Bradsky, Adrian Kaehler Learning OpenCV Computer Vision with the OpenCV Library – O'Reilly, 2008.

5. Kuglin C.D., Hines D.C. The phase correlation image alignment method // Proc. or Intern. Conf. Cybernetics Society, New York, USA. 1975. P. 163–165.

6. Lowe D. Distinctive image features from scale invariant key points // Intern. Journal of Computer Vision. 2004. Vol. 60. P. 91–110.

7. Lowe D. Object recognition from local scale-invariant features // Proc. of Intern. Conf. on Computer Vision, Corfu, Greece. 1999. P. 1150–1157.

8. Bay H., Ess A., Tuytelaars T., Van Gool L. SURF: Speed up Robust Features // Computer Vision and Image Understanding. 2008. Vol. 110, N 3. P. 346–359.

9. Khan N., McCane B., Wyvill G. SIFT and SURF performance evaluation against various image deformations on benchmark dataset // Proc. of Intern. Conf. on Digital Image Computing: Techniques and Applications, Queensland, Australia. 2011.

10. Kotapalle G. R., Kotni S. Security Using Image Processing and Deep Convolutional Neural Networks. 2018 IEEE International Conference on Innovative Research and Development (ICIRD), Bangkok, 11–12 May 2018. 2018. DOI: 10.1109/ICIRD.2018.8376292

11. Jonathan Long, Evan Shelhamer, Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. The IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

12. Neetu Rani. Image Processing Techniques: A Review. Journal on Today's Ideas – Tomorrow's Technologies. 2017. Vol. 5, no. 1. P. 40–49. URL: https://doi.org/10.15415/jotitt.2017.51003

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр
за темою:
«РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ ЛЮДЕЙ НА ВХОДІ В ДОМІВКУ»

Виконав:
студент групи КНД-41
Довгун Сергій
Керівник:
доцент кафедри комп'ютерних наук, доктор філософії
Березовська Юлія Володимирівна

Слайд 1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ

Мета дослідження: підвищити ефективність систем охорони за рахунок розпізнавання облич людей на вході в домівку.

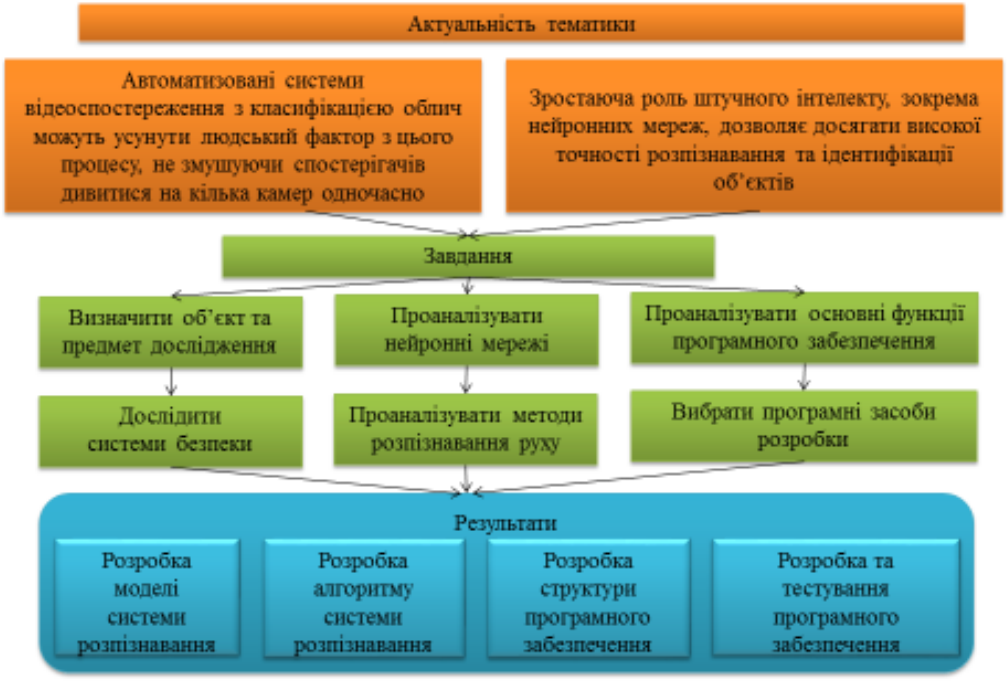
Об'єкт дослідження: процес розпізнавання і класифікації рухомих об'єктів.

Предмет дослідження: методи і алгоритми розпізнавання рухомих об'єктів.

2

Слайд 2

СТРУКТУРНО-ЛОГІЧНА СХЕМА ДОСЛІДЖЕННЯ



3

Слайд 3

СИСТЕМИ БЕЗПЕКИ

Рисунок 1.1 – Камери стеження

Рисунок 1.2 – Бездротова система охорони

Рисунок 1.3 – Камера з автоматичним стеженням пересування

4

Слайд 4

МЕТОДИ РОЗПІЗНАВАННЯ РУХУ В СИСТЕМАХ ВІДЕОСПОСТЕРЕЖЕННЯ

ВІДНІМАННЯ ФОНУ

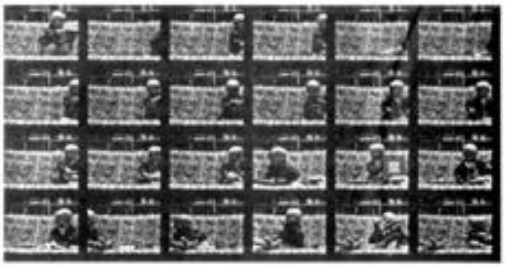


Рисунок 1.4 – Віднімання фону і визначення меж кадрів

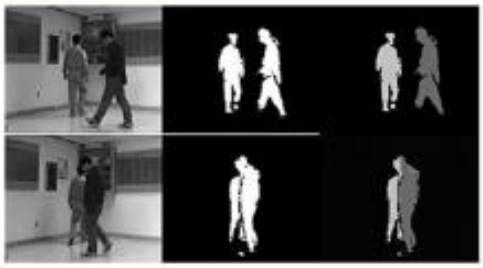


Рисунок 1.5 – Приклад роботи методу віднімання фону

СТЕЖЕННЯ ЗА ТОЧКАМИ

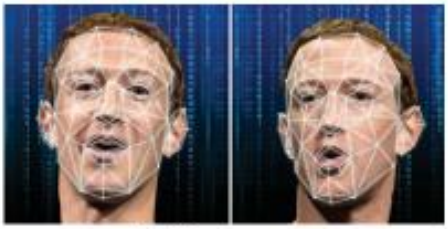


Рисунок 1.6 – Метод стеження за точками

Слайд 5

МЕТОДИ РОЗПІЗНАВАННЯ РУХУ В СИСТЕМАХ ВІДЕОСПОСТЕРЕЖЕННЯ

ФАЗОВА КОРЕЛЯЦІЯ

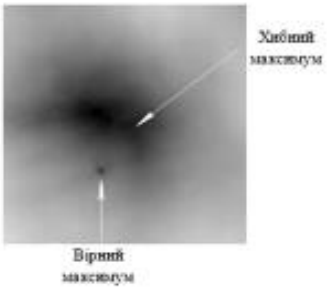
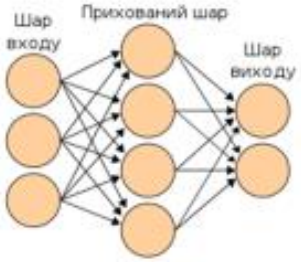


Рисунок 1.7 – Фазова кореляція

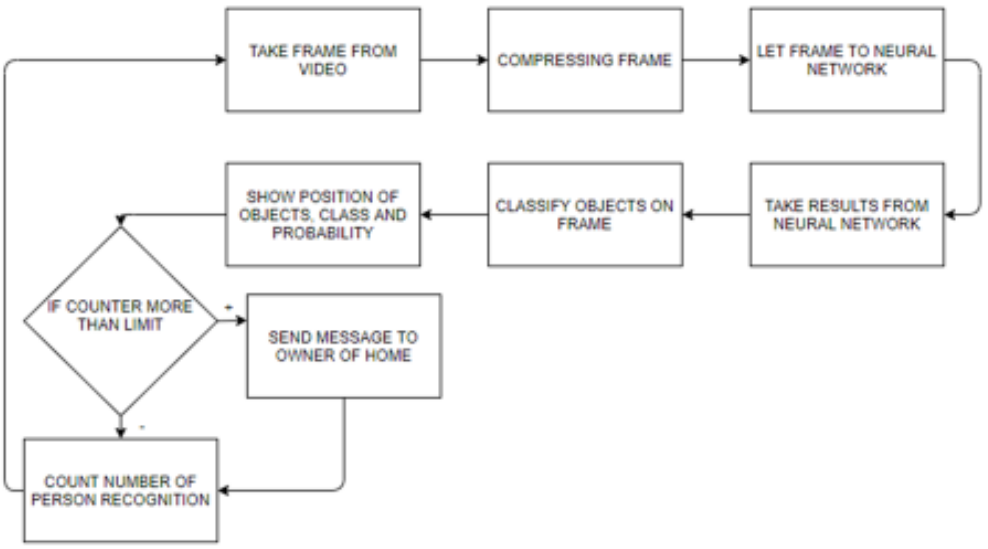


НЕЙРОННІ МЕРЕЖІ

Рисунок 1.8 – Схема нейронної мережі

Слайд 6

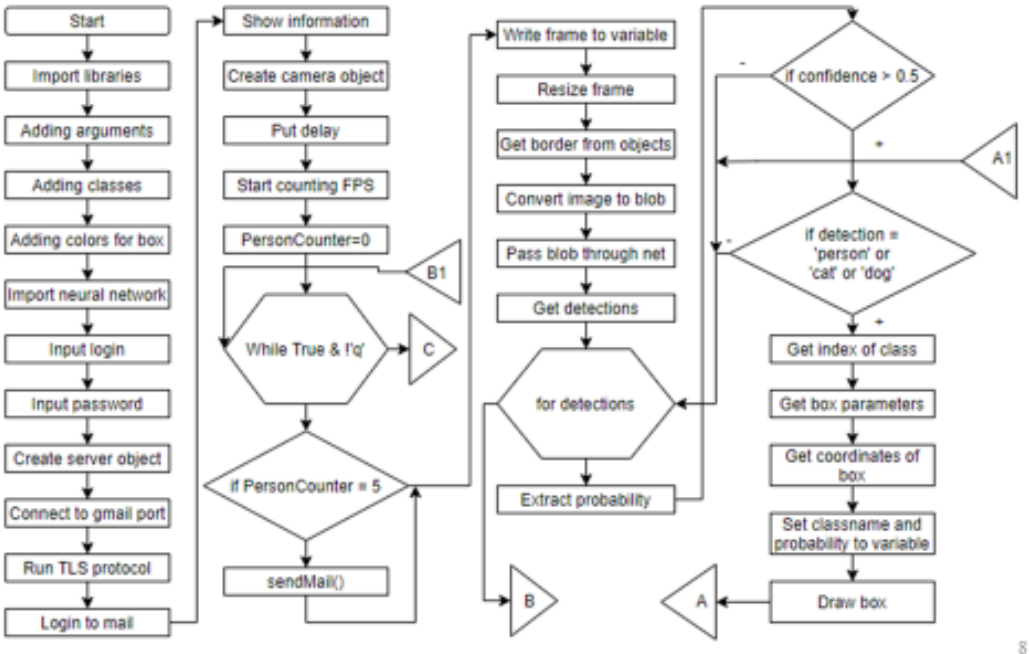
МОДЕЛЬ СИСТЕМИ РОЗПІЗНАВАННЯ РУХІВ ОБ'ЄКТІВ



7

Слайд 7

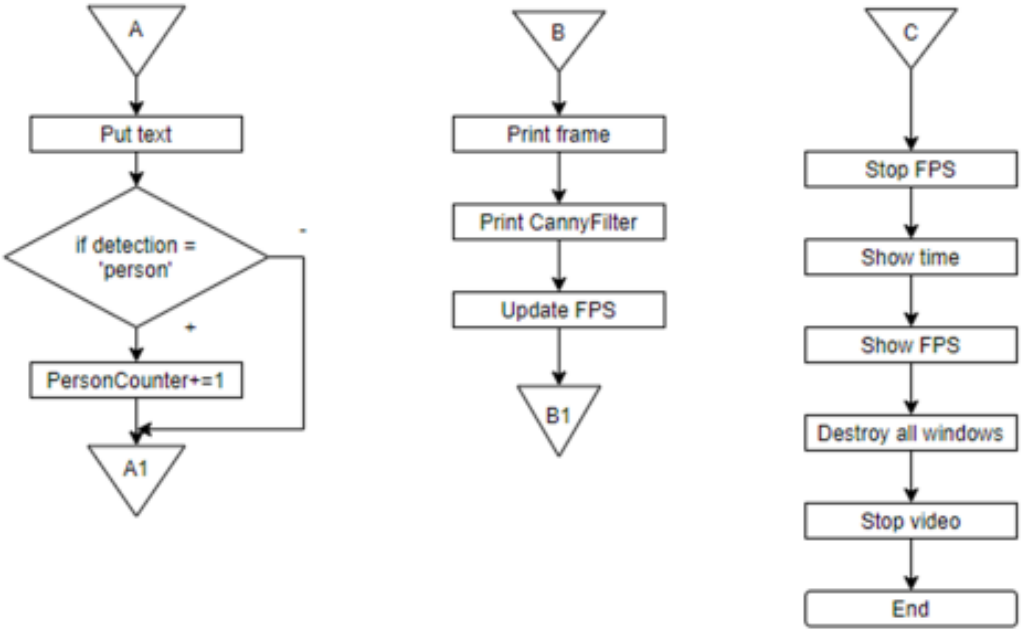
АЛГОРИТМ ПРОГРАМИ РОЗПІЗНАВАННЯ



8

Слайд 8

АЛГОРИТМ ПРОГРАМИ РОЗПІЗНАВАННЯ



9

Слайд 9

СТРУКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

10

Слайд 10

ТЕСТУВАННЯ АЛГОРИТМУ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```
C:\Users\klime>py new_real_time.py --prototxt MobileNetSSD_deploy.prototxt --model MobileNetSSD_deploy.caffemodel
```

Рисунок 1 – Запуск програми з аргументами

```
tSSD_deploy.caffemodel
Message: (important) Loading of neural network model.
Message: (important) Switch on your camera.
```

Рисунок 2 – Валідний результат роботи програми

Рисунок 3 – Виведені на екран оброблені кадри

```
Message: (important) Loading of neural network model.
Message: (important) Switch on your camera.
Message: (important) elapsed time: 45.79
Message: (important) approx. FPS: 2.36
The end!
```

Рисунок 4 – Завершення роботи алгоритму програми

Слайд 11

ВИСНОВКИ

- Розглянуто системи охорони.
- Проаналізовано сучасні методи та підходи до розпізнавання рухів об’єктів у відеопослідовностях.
- Досліджено роль нейронних мереж у розпізнаванні об’єктів.
- Розроблено модель, яка здатна ефективно розпізнавати обличчя людини на вході в домівку.
- Розроблено алгоритм роботи системи розпізнавання.
- Розроблено та протестовано програмне забезпечення.

Слайд 12