

Відеосистема для визначення чисельності людей

Уляна Дзелендзяк¹, Данило Нікульшин²

¹ к.т.н., доцент, кафедра КСА, Національний університет «Львівська політехніка», вул. С. Бандери 12, 79013, Львів, e-mail: uliana.y.dzelendziak@lpnu.ua

² магістр, кафедра КСА, Національний університет «Львівська політехніка», вул. С. Бандери 12, 79013, Львів, e-mail: nikulshyn.daniel@gmail.com

У статті досліджено ефективність використання відеосистеми для виявлення та відстеження переміщень живих об'єктів, зокрема людей. Розроблено модель підрахунку кількості осіб у кадрі з використанням апаратного модуля Arduino ESP_Camera та програмного забезпечення на мові Python. Для обробки відеопотоку застосовано бібліотеки NumPy та OpenCV, що забезпечують виявлення та відстеження рухів у режимі реального часу. Результати експерименту підтверджують, що система здатна точно фіксувати переміщення людей навіть за умов присутності кількох рухомих об'єктів, що свідчить про її практичну ефективність у реальних умовах. Запропонований підхід забезпечує високу точність виявлення рухів у відеопотоках у реальному часі, що робить його перспективним для подальшого застосування у сфері автоматизованого відеоспостереження.

Ключові слова: відеосистема, мікроконтролер, кроковий двигун, датчик, відеопотік, детектор обличчя, штучний інтелект, дескриптор, алгоритми виявлення руху.

Вступ. Відеосистеми підрахунку чисельності людей набувають великого значення у багатьох галузях, таких як безпека, спостереження, охорона здоров'я, транспорт, торгівля, тощо. Застосування цих систем дозволяє отримувати оперативну інформацію про потоки людей, оцінювати завантаженість приміщень у публічних місцях, а також сприяти прийняттю стратегічних рішень для оптимізації управління простором.

У секторі безпеки запропоновану систему можна використовувати для моніторингу зон з високим рівнем безпеки, таких як аеропорти, урядові будівлі та військові об'єкти. Завдяки точному виявленню та відстеженню переміщень людей система може допомогти виявити потенційні загрози безпеці в режимі реального часу, дозволяючи персоналу служби безпеки вживати відповідних заходів для запобігання інцидентам.

У сфері спостереження запропоновану систему можна використовувати для моніторингу громадських місць, таких як міські вулиці та парки, для виявлення та відстеження підозрілої поведінки. Це може допомогти правоохоронним органам виявити та затримати осіб, причетних до злочинної діяльності.

У секторі охорони здоров'я запропоновану систему можна використовувати для моніторингу пацієнтів у лікарнях і будинках престарілих, надаючи дані в режимі

реального часу про рухи та поведінку пацієнтів. Це може допомогти медичним працівникам визначити потенційні проблеми зі здоров'ям і забезпечити більш ефективне лікування.

У транспортних системах та громадських місцях підрахунок людей допомагає відстежувати заповненість транспортних засобів, забезпечувати дотримання норм безпеки, оптимізувати розклад руху.

У магазинах і торгових центрах системи підрахунку людей допомагають визначати конверсію відвідувачів у покупців, оптимізувати розклад роботи персоналу, оцінювати ефективність маркетингових кампаній.

Використання відеосистем підрахунку чисельності людей, зокрема побудованих на OpenCV та супутніх технологіях, може бути адаптовано для військових цілей, зокрема для підрахунку чисельності особового складу військ противника. Вони дозволяють ефективно здійснювати підрахунок особового складу противника на базі відео з дронів, супутників або камер спостереження. Проте для ефективного застосування в реальних умовах необхідно враховувати специфіку бойового середовища, необхідність адаптації під маскування, перекриття та низьку якість відео.

1. Основні підходи до підрахунку чисельності людей

Розвиток комп'ютерного зору та глибокого навчання кардинально змінив підходи до розв'язання задачі підрахунку людей у сучасних відеосистемах. Одним із класичних підходів є використання методів детекції, які орієнтовані на виявлення окремих осіб у кадрі. Серед таких методів варто відзначити алгоритми, що використовують традиційні ознаки (наприклад, Haar Cascades, HOG разом з SVM), а також сучасні нейронні мережі, такі як YOLO, SSD та Faster R-CNN. Системи, засновані на цих методах, демонструють високу точність у випадках помірної щільності людей. Проте, за умов скупчення чи часткового перекриття осіб точність може знижуватися. Стаття [1] демонструє ефективність поєднання детектора YOLOv3 із алгоритмом трекінгу Deep SORT для вирішення задачі підрахунку чисельності людей в реальному часі. Ця робота пропонує глибоке навчання для автоматичного моніторингу соціального дистанціювання за допомогою відеоспостереження. Вона використовує модель об'єктного детектування YOLOv3 для виявлення людей та алгоритм DeepSORT для відстеження з ідентифікаторами. Результати показують, що комбінація YOLO v3 та DeepSORT забезпечує найкращі результати з балансом між точністю та швидкістю кадрів в секунду для моніторингу соціального дистанціювання в реальному часі.

Іншим популярним підходом є використання щільнісних (density map) підходів, коли система не визначає кожну людину окремо, а будує карту, де кожен піксель відповідає "густоті" людей у зображенні. Цей метод особливо дієвий у високощільних сценах, де традиційні детекційні методи стикаються з труднощами. У статті [2] пропонується архітектура, що базується на розширеній згортці (dilated convolution), яка дозволяє побудувати високоточні щільнісні карти за умов надзвичайної скупченості людей.

Для вирішення проблем, пов'язаних із варіацією масштабів людей у кадрі (через зміни перспективи чи відстані до камери), застосовують моделі з багатомасштабною

обробкою. У роботі [3] запропоновано ефективну багатошарову та багатоколонкову згорткову нейронну мережу (MSMC) для оцінки щільності натовпу. Модель складається з трьох компонентів: вилучення ознак, злиття ознак та регресії ознак. Використовується VGG16 для швидшого вилучення ознак, а також згортки з різними розмірами для обробки об'єктів різного розміру. Після блоку злиття застосовується розширена згортка для посилення рецептивного поля без збільшення кількості параметрів. Застосування мультиколоночних архітектур дозволяє аналізувати зображення на різних масштабах, що покращує точність підрахунку у складних умовах.

Одним із викликів при підрахунку людей є уникнення подвійного обліку у сценаріях, де люди переміщуються по кадру. Для цього інтегрують алгоритми трекінгу, які відслідковують переміщення осіб у часі. Системи на базі Kalman filter у поєднанні з алгоритмом Hungarian, а також рішення типу Deep SORT, дозволяють ефективно управляти процесом трекінгу і покращувати точність підрахунку. Стаття [4] описує систему трекінгу людей з верхнього виду, яка поєднує YOLOv3 для виявлення об'єктів та Deep SORT для трекінгу. Використання Kalman filter та алгоритму Hungarian забезпечує точне відстеження та підрахунок осіб у реальному часі.

В останні роки основна увага в дослідженнях спрямована на використання глибоких нейронних мереж. Глибокі згорткові нейронні мережі (CNN) стали основним інструментом у сучасних відеосистемах підрахунку. Новітні моделі, такі як P2PNet, реалізують прямий регресійний підхід до визначення центрів людей на зображенні. Також досліджується використання рекурентних мереж (LSTM) для обліку часової інформації та генеративних мереж (GAN) для покращення якості карт. У статті [5] наведено новий підхід до підрахунку та локалізації людей у натовпі, використовуючи модель P2PNet, яка безпосередньо передбачає координати центрів голів без створення проміжних карт щільності. Для оптимального співставлення передбачених точок з реальними використовується алгоритм Hungarian.

Сучасні відеосистеми підрахунку чисельності людей досягають високої точності завдяки поєднанню традиційних методів комп'ютерного зору та сучасних підходів на базі глибокого навчання. Поєднання методів детекції, побудови щільнісних карт, багатомасштабного аналізу та трекінгу створюють потужний інструментарій, здатний працювати в реальному часі навіть за умов складних візуальних завдань. Перспективними напрямками залишаються багатокамерні системи, тривимірна аналітика та обчислення на периферійних пристроях.

2. Розроблення відеосистеми для визначення чисельності людей

Для розроблення відеосистеми для визначення чисельності людей використовувались: біполярний кроковий двигун (Bipolar Stepper Motor), на якому розташована камера ESP32, плата керування біполярним кроковим двигуном (Driver Board), плата мікроконтролера Arduino Nano (Microcontroller Board). На рис. 1 наведена загальна структура розробленої відеосистеми.

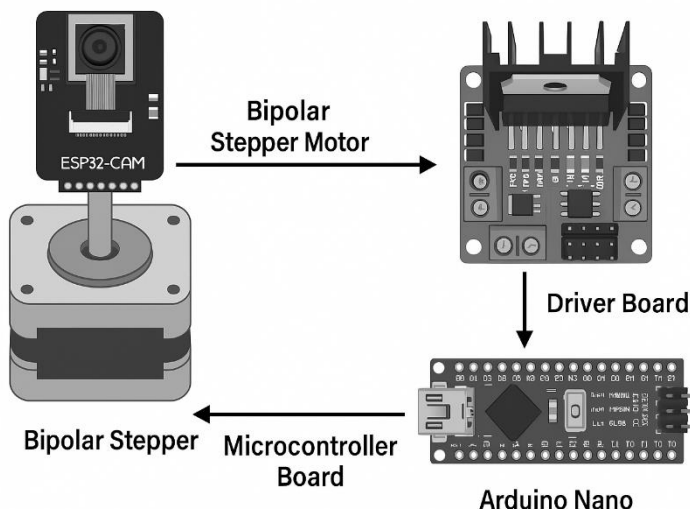


Рис. 1. Загальна структура відеосистеми для визначення чисельності людей

Використання Arduino Nano в системі було обумовлено його здатністю надавати достатньо інтерфейсів для програм користувача, а також здатністю створювати швидкі прототипи з найменшими розмірами. Цей перший вбудований мікроконтролер можна програмувати за допомогою програмного забезпечення Arduino, яке є дуже зручним для нових користувачів. Завдяки найменшій вбудованій апаратній частині Arduino Nano зручно використовувати в проектах макетних плат [6].

Крокові двигуни мають унікальну конструкцію. Це дозволяє контролювати їх з високою точністю без використання механізмів зворотного зв'язку. Кроковий вал керується електромагнітними котушками. Пересувають його маленькими «кроками» вперед або назад.

Кут повороту ротора залежить від кількості імпульсів струму, що подаються, а кутова швидкість ротора точно дорівнює частоті імпульсів, помноженої на кут повороту ротора за один цикл роботи двигуна. Кут повороту двигуна під дією одного імпульсу може мати різні значення в залежності від конструкції двигуна. Як правило, це значення коливається від кількох градусів до кількох десятків градусів. Крокові двигуни, в залежності від призначення, пристосовані для виконання від часток оберту в хвилину до декількох тисяч обертів в хвилину. Існує два типи однокрокових двигунів: уніполярні та біполярні. Ми використовували біполярний двигун у нашій відеосистемі [7].

Модуль WiFi ESP32 CAM Bluetooth із модулем камери OV2640 2 МП для розпізнавання обличчя має дуже конкурентоспроможний модуль камери малого розміру, який може працювати незалежно як мінімальна система з площею лише 40x27 мм; струм глибокого сну до 6 мА і широко використовується в різних програмах IoT. За бажанням клієнта модуль ESP32-CAM отримує накопичувачі та передає зображення по Wi-Fi. Це основна мета цього модуля [8].

На рис. 2 наведена базова схема з'єднань розробленої відеосистеми.

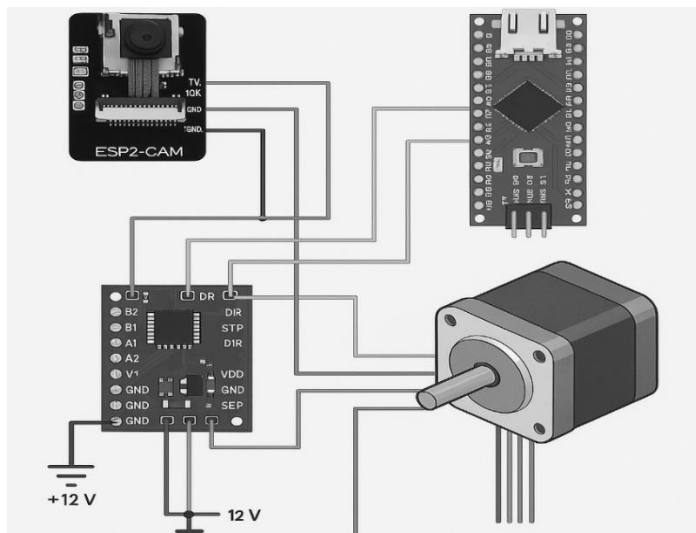


Рис. 2. Базова схема з'єднань

У цій системі з'єднані такі компоненти:

1. Arduino Nano ↔ Драйвер крокового двигуна (A4988).
2. Драйвер крокового двигуна ↔ Кроковий двигун (біполярний).
3. Живлення двигуна: Драйвер ↔ Джерело живлення.
4. ESP32-CAM ↔ Arduino Nano.

Принцип роботи системи полягає в наступному:

1. Ініціалізація системи:
 - При подачі живлення Arduino Nano ініціалізує порти для керування драйвером крокового двигуна.
 - Камера ESP32 завантажує прошивку, з'єднується з Wi-Fi та готується до зйомки або передавання зображення.
2. Керування обертанням крокового двигуна:
 - Arduino Nano подає імпульси на STEP і DIR входи драйвера.
 - DIR визначає напрямок обертання двигуна (вліво або вправо).
 - Кожен імпульс на STEP відповідає одному кроку двигуна.
 - Драйвер у відповідь керує струмом через обмотки двигуна, що спричиняє обертання вала.
3. Переміщення камери:
 - Камера закріплена на валу крокового двигуна або на поворотному кронштейні, керованому двигуном.
 - Обертання двигуна змінює напрямок огляду камери, наприклад, для сканування простору або стеження за об'єктом.
4. Робота камери ESP32:
 - Камера може працювати автономно або за командою від Arduino (через UART або інший інтерфейс).

- Знімки або відео передаються через Wi-Fi на сервер або на мобільний пристрій.
- Можлива інтеграція з інтелектуальними алгоритмами (наприклад, виявлення руху) для керування напрямком повороту камери.

Типовий сценарій використання:

- Камера початково сканує простір на 180°.
- Arduino переміщує кроковий двигун крок за кроком, роблячи паузи для зйомки на кожній позиції.
- Зображення передаються через ESP32 на сервер.
- За необхідності, Arduino може отримати команду на зміну напрямку або точного положення камери.

Програмна складова відеосистеми розроблена з використанням мови програмування Python для забезпечення обробки та візуалізації даних. Для цього було використано низку інструментів і методів, у тому числі NumPy та OpenCV у Python для обробки та аналізу даних [9, 10].

Одним із поширених підходів до використання NumPy і OpenCV для виявлення руху людини є аналіз руху пікселів між послідовними кадрами відеопотоку. Це можна зробити за допомогою алгоритмів виявлення руху, таких як алгоритм оптичного потоку Лукаса-Канаде, який обчислює зміщення пікселів між двома кадрами відеопотоку. Коли вектори руху обчислені, їх можна використовувати для оцінки напрямку та швидкості людей на відео [11].

Інший підхід до використання NumPy і OpenCV для виявлення руху людини полягає у використанні алгоритмів машинного навчання. Це передбачає навчання моделі на наборі даних позначених зображень або відеокадрів для розпізнавання візерунків і особливостей, які вказують на рух людини. Потім навчену модель можна використовувати для аналізу нових відеопотоків і виявлення людей [12, 13]. Відстеження рухомих об'єктів можна виконувати за допомогою передового методу фільтра Калмана (KF). Алгоритм успішно застосовується до стандартних наборів відео [14]. У сучасному світі, завдяки стрімкому розвитку інформаційних технологій, все ширше застосовується комп'ютерна техніка, яка використовує для створення зображень алгоритми виявлення рухомих об'єктів у режимі реального часу [15, 16, 17].

Нижче наведено приклад Python-коду для виявлення людини у відеопотоці з ESP32-CAM. У цьому прикладі використовується дескриптор HOG (гістограма орієнтованих градієнтів) у поєднанні з детектором людини, реалізованим за допомогою бібліотеки OpenCV.

```
# import the necessary packages
import numpy as np
import cv2
# initialize the HOG descriptor/person detector
hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
cv2.startWindowThread()
# open webcam video stream
#cap = cv2.VideoCapture(0)
cap = cv2.VideoCapture('http://192.168.43.67:81/stream')
# the output will be written to output.avi
out = cv2.VideoWriter(
    'output.avi',
    cv2.VideoWriter_fourcc(*'MJPG'),
    15.,
    (640,480))
while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()
    # resizing for faster detection
    frame = cv2.resize(frame, (640, 480))
    # using a greyscale picture, also for faster detection
    gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
    # detect people in the image
    # returns the bounding boxes for the detected objects
    boxes, weights = hog.detectMultiScale(frame, winStride=(8,8) )
    boxes = np.array([x, y, x + w, y + h] for (x, y, w, h) in
boxes])
    for (xA, yA, xB, yB) in boxes:
        # display the detected boxes in the colour picture
        cv2.rectangle(frame, (xA, yA), (xB, yB),
            (0, 255, 0), 2)

        # Write the output video
        out.write(frame.astype('uint8'))
        # Display the resulting frame
        cv2.imshow('frame', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    # When everything done, release the capture
    cap.release()
    # and release the output
    out.release()
    # finally, close the window
    cv2.destroyAllWindows()
    cv2.waitKey(1)
```

Реалізований сценарій здійснює захоплення відеопотоку з модуля ESP32-CAM, після чого виконується масштабування кадрів до зниженої роздільної здатності з метою підвищення швидкодії процесу виявлення об'єктів. Задля подальшої оптимізації обробки зображення конвертується у відтінки сірого. Виявлення людей здійснюється з використанням дескриптора орієнтованих градієнтів (HOG) у поєднанні з вбудованим детектором облич. У результаті обробки формуються обмежувальні рамки навколо виявлених об'єктів. Ці рамки надалі візуалізуються на оригінальному кольоровому зображенні для ідентифікації розташування виявлених осіб. Крім того, оброблені кадри зберігаються у відеофайл «output.avi» із використанням функціоналу VideoWriter бібліотеки OpenCV.

ESP32-CAM була встановлена на кроковому двигуні та підключена до Arduino. Це дозволило камері повертатися вперед і назад на 60 градусів. Весь механізм збірки

складається з пластикової коробки, в якій розміщена плата Arduino і плата керування двигуном. У цій коробці є спеціальний отвір, через який повертається камера. Двигун, на якому встановлена камера ESP32, також знаходиться всередині цього боксу, забезпечуючи безпеку її роботи та захищаючи від можливих пошкоджень (рис. 3).

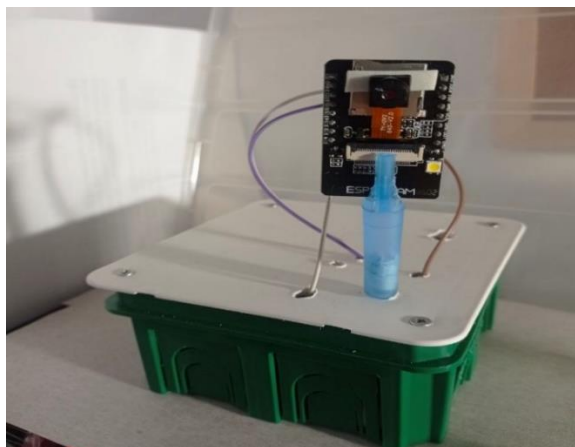


Рис. 3. Відеосистема для визначення чисельності людей

Нижче наведено послідовність дій для інтеграції модуля ESP32-CAM із програмним середовищем Python:

1. Здійснити підключення модуля ESP32-CAM до локальної мережі за допомогою бездротового з'єднання (Wi-Fi) або Ethernet.
2. Ініціалізувати сервер потокової передачі відео на ESP32-CAM шляхом завантаження відповідного скетча. Одним із типових прикладів є скетч «CameraWebServer» із бібліотеки ESP32 для середовища Arduino, який забезпечує трансляцію відеопотоку за URL-адресою.
3. У середовищі Python здійснити підключення до відеопотоку за допомогою функції VideoCapture() бібліотеки OpenCV, вказавши відповідну URL-адресу як параметр.
4. Після успішного встановлення з'єднання здійснюється обробка відеокадрів за допомогою інструментів OpenCV, зокрема для задач обробки зображень, виявлення об'єктів і реалізації алгоритмів машинного навчання.
5. Результати обробки можуть бути виведені на екран або збережені у файл із використанням відповідних функцій OpenCV.

3. Аналіз роботи відеосистеми

На рисунках 4 і 5 представлено результати роботи відеосистеми при обробці вхідних зображень із різною якістю — низькою та високою. Наведені приклади ілюструють залежність ефективності та точності роботи алгоритму від параметрів вхідного зображення. Зокрема, продемонстровано, як алгоритм обробляє зображення з низькою роздільною здатністю або високим рівнем шуму у порівнянні з високоякісними зображеннями з чіткою структурою. Отримані результати

дозволяють зробити висновки щодо стійкості алгоритму до завад та його здатності коректно інтерпретувати дані за різних умов функціонування.

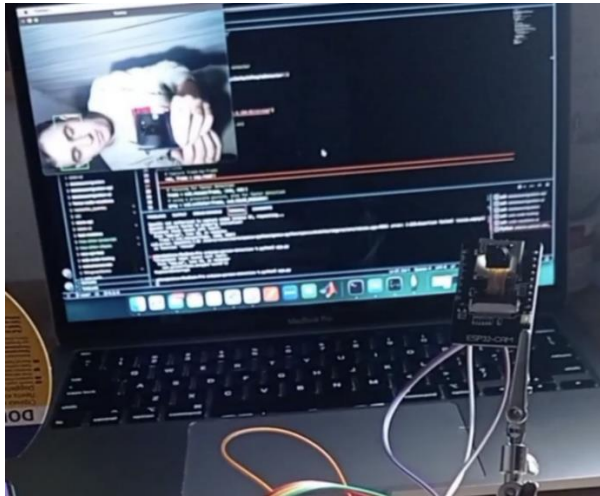


Рис. 4. Демонстрація реалізації з високоякісним зображенням

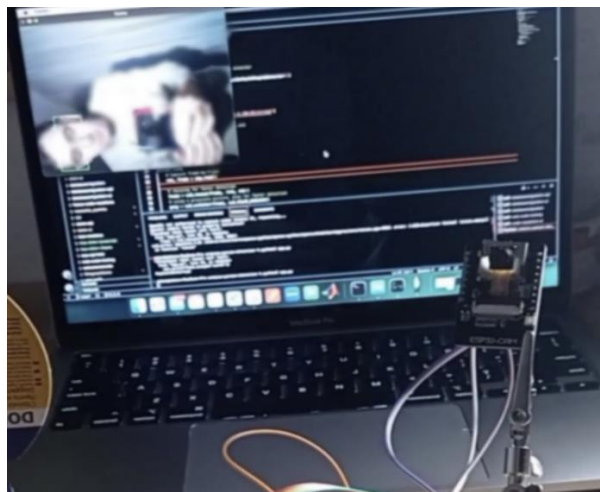


Рис. 5. Демонстрація реалізації із зображенням низької якості

При обробці зображення високої якості (рис. 4) алгоритм забезпечує коректне розпізнавання всіх ключових елементів, зберігаючи високий рівень деталізації та генеруючи очікуваний результат. Це свідчить про високу ефективність алгоритму за сприятливих умов і підтверджує його потенціал для практичного використання. Отже, можна стверджувати, що точність роботи відеосистеми для виявлення та підрахунку осіб, реалізованої на базі OpenCV, значною мірою залежить від якості вхідного зображення. У таблиці 1 представлено порівняльний аналіз впливу роздільної здатності зображення (високої та низької) на точність виявлення.

Таблиця 1

Вплив роздільної здатності зображень на точність виявлення

Роздільна здатність	Приклад формату	Середня точність з OpenCV + DNN	Коментар
Низька ($\leq 640 \times 360$)	SD, 360p	60–75%	Погіршується точність ідентифікації контурів; проблеми з виявленням людей у натовпі
Середня ($\sim 1280 \times 720$)	HD, 720p	85–92%	Базовий рівень для більшості моделей; добра продуктивність
Висока (1920×1080 і вище)	Full HD, 1080p	93–97%	Висока деталізація; найкраща ефективність трекінгу, мінімум помилкових спрацювань

Ці дані демонструють, що зменшення якості зображення може знизити точність виявлення об'єктів (людей) майже на 40%.

Висновки. У результаті проведеного дослідження було розроблено ефективну, доступну та багатофункціональну відеосистему для виявлення та відстеження рухів людей у режимі реального часу. Основною метою роботи було створення рішення, яке поєднує апаратні та програмні компоненти для забезпечення точного моніторингу простору з можливістю подальшої автоматизації. Запропонований підхід базується на використанні модуля ESP32-CAM для бездротової передачі відео, мікроконтролера Arduino Nano для керування, а також крокового двигуна, який забезпечує точне позиціонування камери. Програмна частина реалізована за допомогою мови програмування Python і бібліотек NumPy та OpenCV, що дозволяє здійснювати аналіз відеопотоку, виявлення руху та відстеження об'єктів у реальному часі. Перевагами розробленої системи є точне позиціонування завдяки кроковому двигуну, гнучкість керування через Arduino, бездротова передача відео через ESP32, можливість автоматизації та інтеграції зі штучним інтелектом (наприклад, автоматичне стеження за об'єктом).

У процесі реалізації було виявлено кілька ключових чинників, що впливають на ефективність роботи системи: чутливість відеосистеми до змін у кадрі; роздільна здатність відеопотоку, що визначає якість обробки зображень; якість алгоритмів виявлення та відстеження руху. Для підвищення точності системи запропоновано: використовувати відео з вищою роздільною здатністю; оптимізувати алгоритми виявлення руху; провести налаштування параметрів чутливості в залежності від умов середовища.

Отже, результати експериментального дослідження підтвердили ефективність запропонованої системи. Інтеграція апаратного забезпечення з потужними інструментами обробки зображень дозволяє досягти високої точності та стабільності виявлення і відстеження рухів людини у реальному часі. Система є ефективною та бюджетною для задач відеоспостереження, підрахунку чисельності людей, моніторингу об'єктів, розумних пристроїв з камерами, дослідницьких або навчальних проєктів. Система має великий потенціал для подальшого розвитку та адаптації до різноманітних завдань. При бажанні її легко розширити: додати Bluetooth, Wi-Fi-керування, штучний інтелект або поворот у двох площинах (2 двигуни) - залежно від

задач, які вона виконує. Така гнучкість дозволяє інтегрувати систему у складніші проєкти з вищими вимогами до автоматизації, точності відстеження та керування.

Література

- [1] Pun, N. S., Sonbhadra, S. K., Agarwal, S., & Rai, G. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and DeepSORT techniques, 2020. arXiv preprint arXiv:2005.01385. <https://arxiv.org/abs/2005.01385>
- [2] Li, Y., Zhang, X., & Chen, D. (2018). CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, p. 1091–1100, doi: 10.1109/CVPR.2018.00120.
- [3] Chen, L., Wang, G. & Hou, G. Multi-scale and multi-column convolutional neural network for crowd density estimation. *Multimed Tools Appl* 80, 2021, p. 6661–6674, doi: 10.1007/s11042-020-10002-8.
- [4] Imran Ahmed, Misbah Ahmad, Awaiz Ahmad, Gwanggil Jeon. Top View Multiple People Tracking by Detection Using Deep SORT and YOLOv3 with Transfer Learning within 5G Infrastructure International Journal of Machine Learning and Cybernetics, Volume 12, Issue 11/2021, p.3053–3067, doi: 10.1007/s13042-020-01220-5.
- [5] Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., & Wu, Y. Rethinking Counting and Localization in Crowds: A Purely Point-Based Framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, p. 3365–3374. IEEE, doi: 10.1109/ICCV48922.2021.00335.
- [6] Nano, 2025, [Online]. Available: <https://docs.arduino.cc/hardware/nano/>.
- [7] Arduino and Stepper Motor Configurations, 2025, [Online]. Available: <https://docs.arduino.cc/learn/electronics/stepper-motors/>.
- [8] Nano ESP32, 2025, [Online]. Available: <https://docs.arduino.cc/hardware/nano-esp32/>.
- [9] Arun Kumar Jhapate, Sunil Malviya, Monika Jhapate, “Unusual Crowd Activity Detection using OpenCV and Motion Influence Map”, *2nd International Conference on Data, Engineering, and Applications (IDEA)*, Bhopal, India, 2020, pp. 308 - 313, doi: 10.1109/IDEA49133.2020.9170704.
- [10] Yashika Tomar, Himanshu, Sanjana Devi, Husanpreet Kaur, “Human Motion Tracker using Open CV and Mediapipe”, *3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, Bengaluru, India, 2023, pp. 1199 – 1204, doi: 10.1109/ICIMIA60377.2023.10425865.
- [11] O. Pereverziev, K. Trapezon, “Research of software algorithms for tracking the movement of objects in electronic security systems”. Vcheni zapysky TNU imeni Vernadskogo. Seriya Tekhnichni nauky, Vol. 33 (72), № 6, p. 288-293, 2022, doi: 10.32782/2663-5941/2022.6/47.
- [12] Narendra Kumar Rao B, Nagendra Panini Challa, E S Phalguna Krishna, S. Sreenivasa Chakravarthi, “Facial Landmarks Detection System with OpenCV Mediapipe and Python using Optical Flow (Active) Approach”, *3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2023, pp. 92 - 96, doi: 10.1109/ICACITE57410.2023.10182585.
- [13] S. Damotharasamy, “Approach to model human appearance based on sparse representation for human tracking in surveillance,” *IET Image Processing*, 14(11), 2383-2394, 2020, doi: 10.1049/iet-ipr.2018.5961 2.
- [14] A. B. Sadkhan, S. R. Talebiyan, and N. Farzaneh, "Detection and Moving Object Tracking in images using an improved Kallman Filter (KF) by an Invasive weed optimization algorithm", *International Conference on Advanced Computer Applications (ACA)*, Maysan, Iraq, 2021, pp. 186-192, doi: 10.1109/ACA52198.2021.9626817.
- [15] K.M. Abughalieh, S.G. Alawneh, “Predicting Pedestrian Intention to Cross the Road,” *IEEE Access*, 8, 72558–72569, 2020, doi: 10.1109/ACCESS.2020.2987777.
- [16] Tejonidhi R. Deshpande, Sagar U. Sapkal, "Development of Object Tracking System Utilizing Camera Movement and Deep Neural Network", *IEEE Region 10 Symposium (TENSYP)*, 2022, pp.1-6, doi: 10.1109/TENSYP54529.2022.9864420.
- [17] K. Lenac, A. Cuzzocrea and E. Mumolo, "A Novel Genetic Scan-Matching-Based Registration Algorithm for Supporting Moving Objects Tracking Effectively and Efficiently," in *IEEE Access*, vol. 9, pp. 91741-91753, 2021, doi: 10.1109/ACCESS.2021.3091520.

Video system for determining the number of people

Ulyana Dzelendzyak, Danylo Nikulshyn

The article investigates the effectiveness of using a video system to detect and track the movements of living objects, particularly people. A model for counting the number of people in a frame using the Arduino ESP_Camera hardware module and Python software has been developed. NumPy and OpenCV libraries have been used to process the video stream, which provide detection and tracking of movements in real time. The experimental results confirm that the system is able to accurately record the movements of people even in the presence of several moving objects, which indicates its practical effectiveness in real conditions. The proposed approach provides high accuracy in detecting movements in real-time video streams, which makes it promising for further application in the field of automated video surveillance.

Отримано 30.05.25