Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: *Комп'ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту*

Виконав(ла): студент(ка) *IV* курсу, групи *СІ-41*
спеціальності *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

		Величко Д.В.
	(підпис)	(прізвище та ініціали)
Керівник		*Луцик Н.С.*
	(підпис)	(прізвище та ініціали)
Нормоконтроль		*Тиш Є.В.*
	(підпис)	(прізвище та ініціали)
Завідувач кафедри		*Осухівська Г.М.*
	(підпис)	(прізвище та ініціали)
Рецензент		*Приймак М.В.*
	(підпис)	(прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет *комп'ютерно-інформаційних систем і програмної інженерії*
(повна назва факультету)

Кафедра *комп'ютерних систем та мереж*
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
« *02* » *03* 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *бакалавр*
(назва освітнього ступеня)

за спеціальністю *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

студенту *Величко Діані Вадимівні*
(прізвище, ім'я, по батькові)

1. Тема роботи *Комп'ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту*

Керівник роботи *Луцик Надія Степанівна, PhD, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « *28* » *02* *2023* року № *4/7-238*

2. Термін подання студентом завершеної роботи *22.06.2023*

3. Вихідні дані до роботи *Технічне завдання*

4. Зміст роботи (перелік питань, які потрібно розробити)
Аналіз технічного завдання. Аналіз вимог та постановка завдання. Аналіз можливих рішень. Розробка архітектури системи. Обґрунтування вибору апаратного забезпечення. Обґрунтування вибору програмного забезпечення. Обґрунтування методу реалізації функції ідентифікації екстреної ситуації. Алгоритм роботи системи. Реалізація проєктних рішень. Реалізація функції ідентифікації екстрених ситуацій в комп'ютеризованій системі методами машинного навчання та її тестування

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Загальна архітектура системи
Схема електрична принципова
Блок-схема алгоритму роботи системи
Результати роботи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	*Пилипець М.І., д.т.н., професор*		

7. Дата видачі завдання *02.03.2023*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	*Ознайомлення з завданням до кваліфікаційної роботи*	*02.03-05.03*	*Виконано*
2	*Розробка технічного завдання*	*06.03-12.03*	*Виконано*
3	*Робота над розділом «Аналіз технічного задання»*	*13.03-20.03*	*Виконано*
4	*Розробка архітектури системи*	*20.03-28.03*	*Виконано*
5	*Вибір апаратного забезпечення*	*23.03-29.03*	*Виконано*
6	*Вибір програмного забезпечення*	*29.03-04.04*	*Виконано*
7	*Розробка алгоритму роботи системи*	*04.04-19.04*	*Виконано*
8	*Розробка прототипу та налаштування системи*	*19.04-30.04*	*Виконано*
9	*Формування набору даних для навчання моделі штучного інтелекту*	*01.05-18.05*	*Виконано*
10	*Реалізація функції ідентифікації екстрених ситуацій*	*19.05-25.05*	*Виконано*
11	*Перевірка прийнятих рішень*	*26.05-02.06*	*Виконано*
12	*Підготовка розділу «Безпека життєдіяльності, основи охорони праці»*	*03.06-08.06*	*Виконано*
13	*Оформлення пояснювальної записки та графічної частини*	*09.06-16.06*	*Виконано*
14	*Нормоконтроль*	*16.06*	*Виконано*
15	*Захист кваліфікаційної роботи*	*23.06*	*Виконано*

Студент _____ *Величко Д.В.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Луцик Н.С.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Комп’ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту // Кваліфікаційна робота бакалавра // Величко Діана Вадимівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, група СІ-41 // Тернопіль, 2023 // с. – 80 , рис. – 18 , табл. – 3 , додат. – 4, бібліогр. – 19.

Ключові слова: ВІДЕОНАГЛЯД, КОМП’ЮТЕРИЗОВАНА СИСТЕМА, ESP32-CAM, АЛГОРИТМ, ШТУЧНИЙ ІНТЕЛЕКТ.

Кваліфікаційна робота присвячена розробці комп’ютеризованої системи відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту, яка здатна виявляти нетипові (екстрені) ситуації.

Пояснювальна записка складається з чотирьох розділів.

У першому розділі проведено аналіз вимог до проектованої комп’ютеризованої системи відеонагляду. Обґрунтовано актуальність теми та сфери використання даної системи.

У другому розділі було розроблено архітектуру комп’ютеризованої системи. Обґрунтовано вибір апаратної, програмної складових та методу реалізації функції ідентифікації екстреної ситуації для реалізації системи.

У третьому розділі описано реалізацію функції ідентифікації екстрених ситуацій в комп’ютеризованій системі методами машинного навчання та її тестування.

В четвертому розділі розглянуті питання безпеки життєдіяльності та охорони праці.

ABSTRACT

Computerized video surveillance system with the function of identifying emergency situations based on artificial intelligence // Bachelor's thesis / Velychko Diana Vadymivna // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, group SI-41 // Ternopil, 2023 // p. – 80 , Fig. – 18, Table – 3, Supplement – 4 , Bibliog. – 19.

Keywords: VIDEO SURVEILLANCE, COMPUTERIZED SYSTEM, ESP32-CAM, ALGORITHM, ARTIFICIAL INTELLIGENCE.

The qualification work is devoted to developing a computerized video surveillance system with the function of identifying emergency situations based on artificial intelligence, which is capable of detecting atypical (emergency) situations.

The explanatory note consists of four sections

The first section presents the analysis of the requirements for the designed computerized video surveillance system. The relevance of the topic and the scope of use of this system is substantiated.

In the second section, the architecture of the computerized system is developed. The choice of hardware, software components, and the method of implementing the emergency identification function for the system is substantiated.

Section 3 describes the implementation of the emergency identification function in the computerized system using machine learning methods and its testing.

The fourth section discusses the issues of life safety and labor protection.

ЗМІСТ

ВСТУП 7
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ 9
1.1 Аналіз вимог та постановка завдання 9
1.2 Аналіз можливих рішень 10
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА 17
2.1 Розробка архітектури комп'ютеризованої системи 17
2.2 Обґрунтування вибору апаратного забезпечення 18
2.3 Обґрунтування вибору програмного забезпечення 24
2.4 Обґрунтування методу реалізації функції ідентифікації екстреної ситуації .. 26
2.5 Алгоритм роботи системи 30
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА 32
3.1 Реалізація проєктних рішень 32
3.2 Реалізація функції ідентифікації екстрених ситуацій в комп'ютеризованій системі методами машинного навчання та її тестування 34
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ 43
4.1 Надзвичайні ситуації: визначення, причини, класифікація 43
4.2 Соціальне значення охорони праці 47
ВИСНОВКИ 50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 51
ДОДАТКИ 53

ВСТУП

Забезпечення безпеки людей є важливою проблемою сьогодення. Існуючі системи не завжди можуть своєчасно здійснювати виявлення небезпечної ситуації та інформувати про неї. Сучасні технології надають можливість створення комп'ютеризованих систем відеонагляду, які дозволять мінімізувати ризики для людей та покращити рівень їх безпеки. Ключовим елементом таких систем є штучний інтелект, який є однією з найперспективніших технологій, яка може забезпечити ефективне функціонування системи відеонагляду та виявлення екстрених ситуацій. Використання таких технологій може зменшити відповідальність персоналу, забезпечити вчасну реакцію на небезпечні ситуації та запобігти можливим наслідкам, тобто дозволяє автоматизувати процеси аналізу та ідентифікації екстрених ситуацій, які можуть загрожувати безпеці людей.

Комп'ютеризована система відеонагляду, яка б мала функцію ідентифікації нетипових (екстрених) ситуацій може допомогти виявити потенційно небезпечні обставини, які загрожують безпеці людей та майна. Штучний інтелект, який використовується в системі, здатний автоматично аналізувати відеопотік та виявляти зміни, які можуть сигналізувати про наявність екстрених ситуацій. Крім того, така комп’ютеризована система може сприяти ефективнішій роботі служб безпеки, оскільки вона здатна автоматично виявляти та повідомляти про екстрені ситуації, що звільняє людей від необхідності постійного спостереження за відеопотоком, підвищити точність ідентифікації таких ситуацій, що забезпечує оперативну реакцію на потенційні небезпеки та зменшення ризиків від них.

Таку систему можна застосовувати в різних приміщеннях для різних галузей діяльності, наприклад, в банках, магазинах, школах та інших установах. До того, використання такої системи може знизити витрати на забезпечення безпеки, оскільки вона замінює людський чинник в процесі спостереження за відеопотоком.

Отже, розробка такої комп’ютеризованої системи відеонагляду, яка б містила функцію ідентифікації небезпечних (екстрених) ситуацій є важливою та актуальною задачею.

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог та постановка завдання

Комп’ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту (КСВНІЕС) призначена для відеомоніторингу в приміщеннях організацій, підприємств та установ різного призначення, наприклад таких як: банки, школи, торгівельні та розважальні заклади, приміщення сфери обслуговування та ін., а також як складова систем «розумного будинку».

Відповідно до поставленого завдання КСВНІЕС повинна виявляти та інформувати про виникнення екстрених (нетипових) ситуацій шляхом сповіщення відповідних служб, тому для його реалізації необхідно:

- запропонувати ознаки ідентифікації екстреної ситуації системою;
- проаналізувати системи відеонагляду з метою їх удосконалення відповідно до поставлених вимог;
- обґрунтувати оптимальну для інтеграції у систему відеонагляду модель ідентифікації екстрених (нетипових) ситуацій відповідно до запропонованих ознак;
- розробити архітектуру КСВНІЕС та обґрунтувати її забезпечення;
- розробити алгоритм роботи КСВНІЕС;
- здійснити налаштування КСВНІЕС для трансляції відеоданих;
- сформувати набір даних (датасет) для навчання моделі штучного інтелекту та здійснити перевірку прийнятих рішень.

Варто відмітити, що система повинна сама приймати рішення про небезпеку (екстрену, нетипову ситуацію) за допомогою використання штучного інтелекту та інформувати про неї.

Екстреною ситуацією (ЕС) система буде вважати таку, коли у людини, яка знаходиться в приміщенні, будуть підняті обидві руки вгору. Тобто система повинна розпізнати позу людини і при фіксації заданої пози – «руки вгору», інформувати про небезпеку.

Така сигнальна поза для виявлення небезпеки вибрана тому що, якщо є реальна загроза для людини, наприклад, людина піддається примусу, коли хтось їй погрожує, що може свідчити про пограбування, залякування або затримання, то найчастішою реакцією на такі подразники є саме підняття рук догори, отже, поза жертви із піднятими руками є серйозним сигналом. У загарбників така поза жертви не викликатиме підозри, а системі вона буде сигналом про екстрену ситуацію.

До КСВНІЕС, враховуючи мету її використання, висуваються також такі вимоги як: безперебійна та стабільна робота; енергоефективність, низька вартість, невеликі розміри камери відеонагляду.

1.2 Аналіз можливих рішень

Поставлені завдання щодо розробки КСВНІЕС можна реалізувати використовуючи різне забезпечення як апаратне, так і програмне. Оскільки КСВНІЕС може бути складовою систем «розумних будинків» або систем відеонагляду, тому варто за основу взяти типову структуру систем відеонагляду (СВН) і в неї внести певні зміни відповідно до вимог технічного завдання, враховуючи необхідність використання засобів штучного інтелекту (ШІ).

Традиційні СВН складаються з сукупності програмно-апаратних засобів, які реєструють та транслюють відеопотік для нагляду за об’єктом спостереження [1, 2].

На рисунку 1.1 показано узагальнену структуру СВН. Складовими такої системи є: відеокамера, контролер з WI-FI для передавання інформації, система

віддаленого спостереження, яка, в свою чергу, містить сервер та комп’ютер (планшетний комп’ютер або смартфон).

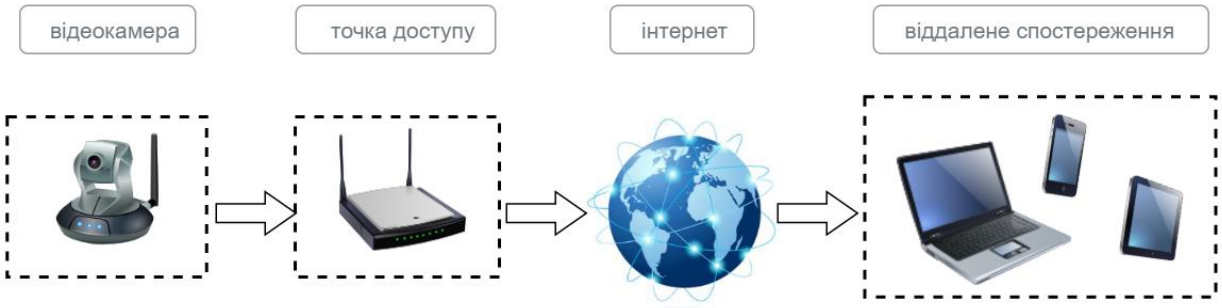


Рисунок 1.1 – Загальна структура системи відеонагляду

Система, яка розробляється, крім відеонагляду повинна ідентифікувати екстрені ситуації з метою інформування про них. Тому звичайну систему відеонагляду, узагальнена структура якої зображена на рис.1.1, потрібно вдосконалити, передбачивши функцію виявлення нетипової (екстреної) ситуації [1].

Реалізувати КСВНІЕС можна використовуючи різне забезпечення. Наприклад, на основі ESP32,Raspberry Pi чи STM32. Кожен з них має як переваги, так і свої недоліки, тому важливим є зважити на них при розробці КСВНІЕС, щоб знайти оптимальне рішення.

Raspberry Pi - це одноплатний комп'ютер, який дозволяє забезпечити високий рівень функціональності та гнучкість у використанні. Застосування Raspberry Pi для систем відеонагляду забезпечує широкі можливості налаштування та опрацювання відеоданих, наприклад, таких як виявлення руху, розпізнавання об’єктів, людей та інші функції за допомогою різних програм та бібліотек [3]. Raspberry Pi, в порівнянні з ESP32 та STM32, є потужнішим, але і споживає більше енергії, а також є дорожчим.

ESP32 - це мікроконтролер з підтримкою Wi-Fi та Bluetooth, який має низьку споживану потужність та невеликі розміри. Використання ESP32 у

системах відеонагляду дозволяє підключати камери та передавати відеодані через Wi-Fi до інших пристроїв, наприклад, таких як смартфони або сервери [4].

ESP32-CAM - це спеціальна версія ESP32, яка має вбудовану камеру і призначена саме для застосування у СВН. Таким чином, вбудований модуль камери дозволяє безпосередньо підключати камеру до ESP32 без необхідності використовувати додаткові модулі або розширювачі. Підтримка Wi-Fi та Bluetooth дозволяє передавати відеодані через бездротові мережі до інших пристроїв або хмарних серверів, а низьке споживання енергії - ефективно використовувати ESP32-CAM у системах з живленням від батарей або сонячних панелей [5]. До того ж ESP32-CAM має компактний розмір, що є ще додатковою перевагою при використанні її для реалізації КСВНІЕС.

Мікроконтролери серії STM32 також можна використати при розробці систем відеонагляду. STM32 має високу продуктивність та легку розробку програмного забезпечення та інші можливості, але необхідно врахувати відсутність вбудованої камери [6, 7]. Порівняно з Raspberry Pi, STM може мати обмежену здатність управління та опрацювання відеоданих, а з ESP32 - меншу підтримку бездротових інтерфейсів.

Отже, враховувавши техніко-економічні вимоги до КСВНІЕС та порівнявши варіанти можливих рішень, найдоцільніше використати ESP32-CAM, що обґрунтовується його вбудованою камерою, низьким споживанням енергії, підтримкою бездротових інтерфейсів, доступною вартістю, компактним розміром, простотою використання та програмування.

Як вже зазначено, в КСВНІЕС, екстреною ситуацією (ЕС), яка вказуватиме на потенційну небезпеку для людини, запропоновано вважати позу людини "руки вгору". Оскільки, люди при потенційній небезпеці здебільшого змушені підняти руки вверх, як знак ризику, або як засіб підкорення. Така поза викликає увагу, підкреслює емоційну реакцію на ті чи інші події і, як правило, використовується, коли є загроза.

Для ідентифікації ЕС КСВНІЕС з відео доцільно використовувати штучний інтелект (ШІ), а саме використати модель розпізнавання пози людини. Вхідними даними КСВНІЕС є відеопотік. Перед подачею вхідних даних на модель ШІ необхідно здійснити їх опрацювання, тобто «витягти» тільки ту інформацію, яка необхідна для вирішення завдання - ідентифікації екстреної ситуації, про яку сигналізує конкретна поза людини.

Усі підходи до розпізнавання пози можна згрупувати в методи «знизу вгору» та «зверху вниз» [8].

Підхід "знизу вгору" у розпізнаванні пози передбачає, що спочатку виявляються всі частини кожної людини (оцінюють кожен суглоб тіла) на зображенні. Після цього виявлені частини асоціюються або групуються для визначення, які частини належать кожній окремій людині на зображенні. Тобто, методи «знизу вгору» спочатку оцінюють кожен суглоб тіла, а потім групують їх, щоб сформувати унікальну позу. Підхід «знизу вгору» локалізує всі ключові точки людського тіла на вхідному зображенні та збирає їх за допомогою запропонованих алгоритмів кластеризації. Іншими словами, цей підхід безпосередньо виявляє всі ключові моменти на зображенні та пов’язує їх із ключовими точками людей [8].

Натомість, методи «зверху вниз» спочатку запускають детектор особи, а потім оцінюють суглоби тіла в межах виявлених обмежувальних рамок. Тобто, низхідний підхід базується на детектуванні людини, який передбачає формування обмежувального прямокутника навколо людини. Далі, виявлене зображення людини обрізається та передається в систему оцінки пози. Іншими словами, спочатку йде етап виявлення особи, а потім етап визначення ключових точок. Цей підхід все ще переважає в загальнодоступних наборах даних, наприклад, таких як MS COCO10, і його можна узагальнити на основі таких аспектів, як контекстне моделювання, ефективна стратегія навчання та методи постобробки.

Найпопулярнішими методами розпізнавання пози є: OpenPose, HRNet, DeepCut, AlphaPose, DeepPose, PoseNet, DensePose [9].

OpenPose є одним із найпопулярніших підходів «знизу вгору» для оцінки пози людини. Ця архітектура передбачає оцінку пози кількох людей у реальному часі. При чому OpenPose — це програма з відкритим кодом для виявлення кількох осіб з високою точністю визначення ключових точок тіла, стопи, руки й обличчя. Перевагою OpenPose є те, що це API, який надає користувачам гнучкість вибору вихідних зображень із відеокамер, веб-камер тощо, що важливо для вбудованих системних програм (наприклад, інтеграція з камерами та системами відеонагляду). Він підтримує різні апаратні архітектури, такі як графічні процесори CUDA, графічні процесори OpenCL або пристрої, що працюють лише з процесором. Полегшена версія достатньо ефективна для додатків Edge Inference з опрацюванням на пристрої в онлайн режимі.

High-ResolutionNet (HRNet) — нейронна мережа для визначення пози людини. Це архітектура, яка використовується в задачах опрацювання зображень для знаходження ключових точок (суглобів) конкретного об’єкта чи людини на зображенні. Перевагою цієї архітектури над іншими є те, що нейронна мережа підтримує представлення високої роздільної здатності під час оцінки поз. Наприклад, архітектура HRNet ефективна для виявлення пози людини під час спортивних трансляцій.

DeepCut — ще один популярний висхідний підхід для оцінки пози людини, який використовується для визначення поз кількох людей. Модель працює, виявляючи кілька людей на зображенні, а потім прогнозує місця з’єднання для кожного об’єкта. DeepCut можна застосувати до відео або зображень із кількома особами/об’єктами, наприклад, футболу, баскетболу тощо.

AlphaPose – метод регіональної оцінки пози кількох людей з підходом «зверху вниз», який використовується для визначення поз за наявності

неточних обмежувальних рамок людини. Тобто це оптимальна архітектура для оцінки поз людини за допомогою оптимально визначених обмежувальних рамок. Архітектура AlphaPose застосовна для виявлення пози як однієї, так і кількох осіб на зображеннях або відео.

DeepPose - засіб оцінки пози людини, який використовує глибокі нейронні мережі. Глибока нейронна мережа DeepPose фіксує всі з'єднання, об'єднує шар об'єднання, шар згортки та повністю пов'язаний шар, щоб утворити частину цих шарів.

PoseNet — це архітектура оцінювача пози, побудована на TensorFlow.js для роботи на «легких» пристроях, таких як браузери або мобільні пристрої та з можливістю визначення поз однієї або кількох людей.

DensePose - це техніка оцінки пози, яка спрямована на відображення всіх пікселів RGB-зображення на 3D-поверхню людського тіла.

Складною задачею, яка вимагає високої обчислювальної потужності є багатокадрова оцінка пози людини в складних ситуаціях. КСВНІЕС необхідно розпізнати координати тіла людини в двовимірному просторі - його скелет, зокрема, позу людини «руки вгору». Для отримання «скелета», що рухається, спочатку потрібно виділити цей об'єкт, для чого існують відповідні алгоритми. Детектори людських суглобів показують хорошу продуктивність для статичних зображень, але їх продуктивність часто є недостатньою, коли моделі машинного навчання застосовуються до відеопослідовностей для відстеження пози в реальному часі. Деякими проблемними моментами є опрацювання розмиття в русі, розфокусування відео, оклюзії пози та нездатність зафіксувати часову залежність між відеокадрами. Застосування звичайних рекурентних нейронних мереж викликає емпіричні труднощі при моделюванні просторових контекстів [9].

Враховуючи переваги та недоліки відомих методів розпізнавання пози людини, а також необхідність швидкого виявлення ЕС, в КСВНІЕС найдоцільніше використовувати PoseNet.

Як було вказано раніше, PoseNet - це модель машинного навчання, яка використовується для розпізнавання пози людини на зображенні, відео або з камери. Вона використовує нейронну мережу глибокого навчання для визначення положення та орієнтацію ключових точок тіла людини на зображенні, таких як руки, ноги, голова і т.д. У результаті - отримується інформація про позу людини на фотографії чи відео [10].`

Модель PoseNet можна використовувати як у вигляді бібліотеки для розробників, так і у вигляді готових додатків для кінцевих користувачів. Наприклад, Google надає бібліотеку TensorFlow.js, яка містить моделі реалізації PoseNet, а також безкоштовну програму для Android та iOS -"Move Mirror" (цей експеримент більше не активний, але інформація про нього та код, використаний для його втілення досі доступні на сайті “Experiments with Google”), яка використовувала цю модель для ігрових і творчих цілей.

Модель працює шляхом використання нейронних мереж глибокого навчання для аналізу зображень або відео та визначення пози людини в кожному кадрі. Для розпізнавання ключових точок тіла PoseNet використовує метод, який названий алгоритмом RANSAC (рандомізований алгоритм середнього квадрата) [10].

Таким чином, запропоновано ідентифікувати в системі екстрену ситуацію за позою людини «руки вгору», її визначення здійснювати, використовуючи модель машинного навчання PoseNet, та реалізувати КСВНІЕС на основі ESP32-CAM.

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Розробка архітектури комп’ютеризованої системи

Поставлені завдання щодо розробки КСВНІЕС на основі ШІ можна реалізувати, застосовуючи різне забезпечення, але, враховуючи техніко-економічні вимоги до системи, використано ESP32-CAM (див.п.1.2.).

Запропонована загальна архітектура КСВНІЕС зображена на рисунку 2.1.

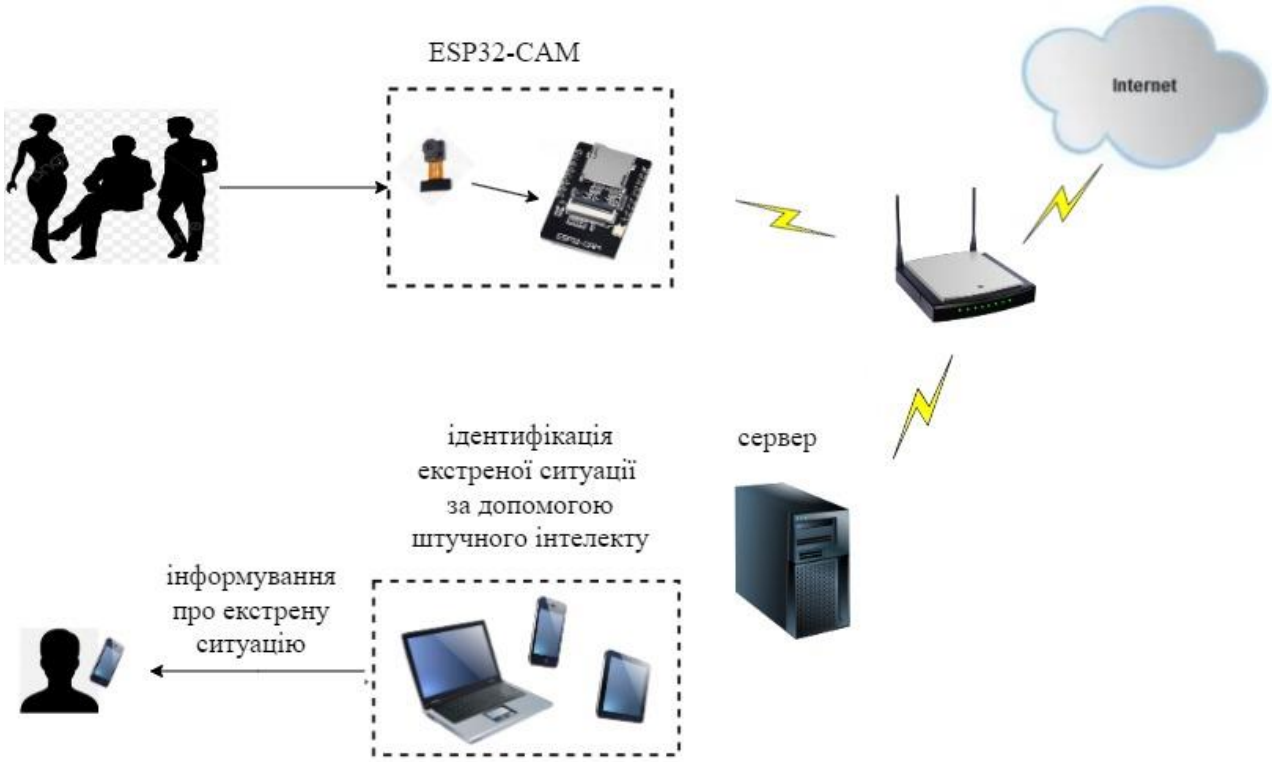


Рисунок 2.1 – Загальна архітектура КСВНІЕС

					КС КРБ 123.035.00.00 ПЗ	
Змн.	Арк.	№ докум.	Підпис	Дата		
Розроб.		*Величко Д.В*			Проєктна частина	Літ. / Арк. 17 / Акркушів
Перевірив		*Луцик Н.С.*				
Рецензент		*Приймак М.В.*				ТНТУ, каф. КС, гр. СІ-41
Н. Контр.		*Тиш Є.В.*				
Зав. каф.		*Осухівська Г.М*				

Основними компонентами КСВНІЕС, показаної на рис. 2.1, є: 2-мегапіксельна камера OV2640 (в складі модуля ESP32-CAM); модуль ESP32-CAM; сервер; смартфон, планшет чи ПК (для відображення відеоінформації та забезпечення доступу до модуля через браузер); згорткова нейронна мережа (для ідентифікації екстреної ситуації); програма, яка відправляє інформацію про настання екстреної ситуації (на смартфон, планшет, ПК).

Отже, основним компонентом КСВНІЕС є ESP32-CAM, який використано для запису та передавання відео, тобто здійснює «захоплення», перетворення зображення в цифрові сигнали, опрацювання отриманих цифрових сигналів, регулювання та передавання на сервер. Компонентом системи, який здійснює ідентифікацію екстреної ситуації, є згорткова нейронна мережа, що використовується сервісом PoseNet.

Для виявлення екстреної ситуації застосовано алгоритм визначення пози. Цей компонент системи аналізує отримані відеодані та визначає позу людини «руки вгору» (що є окремим досить складним завданням).

Зберігання даних (тобто отриманих відеозаписів та історії екстрених ситуацій) в системі запропоновано здійснювати у хмарі або на фізичному сховищі сервера. Для відправлення повідомлення про екстрену ситуацію запропоновано використати Telegram бот.

Така архітектура дозволяє отримати доступ до відеоконтенту в будь-який час, реалізувати функцію ідентифікації екстреної ситуації (пози людини «руки вгору»), а також забезпечити можливість інформування про екстрену ситуацію.

2.2 Обґрунтування вибору апаратного забезпечення

КСВНІЕС реалізована з використанням ESP32-CAM. Такий вибір обґрунтовується проведеним техніко-економічним аналізом вимог до системи

та інших варіантів можливих рішень (див. розділ 1), а також враховуючи такі фактори [4, 5]:

– наявність вбудованої камери, що є однією з вагомих переваг ESP32-CAM, зменшує необхідність в додатковому обладнанні та полегшує підключення камери до мікроконтролера, а також дозволяє спростити конфігурацію системи та зробити її компактною;

– низьке споживання енергії, що дозволяє працювати тривалий час без необхідності частої зарядки або заміни батарей та робить її ефективною;

– підтримка бездротових технологій для передавання відеоданих через бездротову мережу до інших пристроїв або хмарних серверів (використання Wi-Fi дозволяє отримувати доступ до відеопотоку через мобільні додатки або віддалений доступ);

– порівняно низька вартість, що може бути важливим фактором, особливо для проєктів з обмеженим бюджетом або великою кількістю вузлів відеонагляду;

– гнучкість та розширюваність, оскільки ESP32-CAM через велику кількість GPIO-портів підтримує різні інтерфейси, такі як I^2C, SPI, UART, що дозволяє підключати різні сенсори, давачі або актуатори для розширення можливостей системи відеонагляду;

– розмір та портативність, що дозволяє легко вбудовувати в різні системи та зробити непомітним для отримання інформації КСВНІЕС;

– простота використання та програмування, оскільки ESP32-CAM має середовище розробки, що дозволяє легко програмувати його за допомогою ArduinoIDE або MicroPython, що спрощує розробку системи відеонагляду на базі ESP32-CAM.

Модуль ESP32-CAM побудований на мікроконтролері ESP32 і передбачає 2-мегапіксельну камеруOV2640 та слот для microSD [4, 5].

Характеристики модуля наведені в таблиці 2.1, а камери – в таблиці 2.2 [4, 5].

Таблиця 2.1 – Характеристики ESP32-CAM.

Параметр	Опис
Підтримувані Wi-Fi мережі	802.11 b/g/n
Камера (входить до комплекту)	OV2640
Формат зображення:	JPG, BMP, відтінки сірого
Тактова частота мікроконтролера (макс.)	240 МГц
Безпека	WPA/WPA2/WPA2-Enterprise/WPS
Обсяг ОЗП	520 КБ вбудованої 4 Мбіт зовнішньої PSRAM
Обсяг ПЗП (у складі ESP32)	448 КБ
Обсяг ПЗП (зовнішній чіп на платі)	32 Мбіт
Робоча напруга	5 В
Максимальний струм (Wi-Fi + камера + підсвічування)	350 мА
Діапазон робочих температур	-20°C..+85°C
Відносна вологість:	< 90%
Розміри плати, мм	40,5x27x4,5
Вага	10 г

Таблиця 2.2 – Основні характеристики OV2640

Параметр	Опис
Формат зображення:	JPG, BMP, відтінки сірого
Роздільна здатність	UXGA 1600*1200
Напруга живлення	5 В (через контактний роз'єм) Основне живлення – 1,3 VDC±5% Аналогове – 2,5-3,0VDC Входу/виходу – 1,7-3,3 V
Споживана потужність у режимі використання	125мВт, 15fps, UXGA YUV режим 140мВт – для 15fps, UXGA режим компресії
Споживана потужність (stand by)	600 мкА
Основний температурний режим	0-50 ^{0}C
Вихідні формати (8-біт)	YUV (422/420)/YCbCr422 RGB565/555 8-бітні стиснуті дані 8/10-бітні Raw RGB дані
Розмір лінзи	1/4"
Основний кут	25^0
Максимальна швидкість передачі зображення	UXGA/ sXGA – 15fps SVGA – 30 fps CIF – 60 fps
Чутливість	0,6 В/Лк-с
Співвідношення сигнал/шум	40дБ
Розмір пікселів	2,2мкм*2,2мкм
Площа зображення	3590 мкм*2684 мкм
Фіксований шум	<1%

Схеми компонентів системи показано на рисунках 2.2 та 2.3. Мікроконтролер ESP32 схематично представлений на рисунку 2.2.

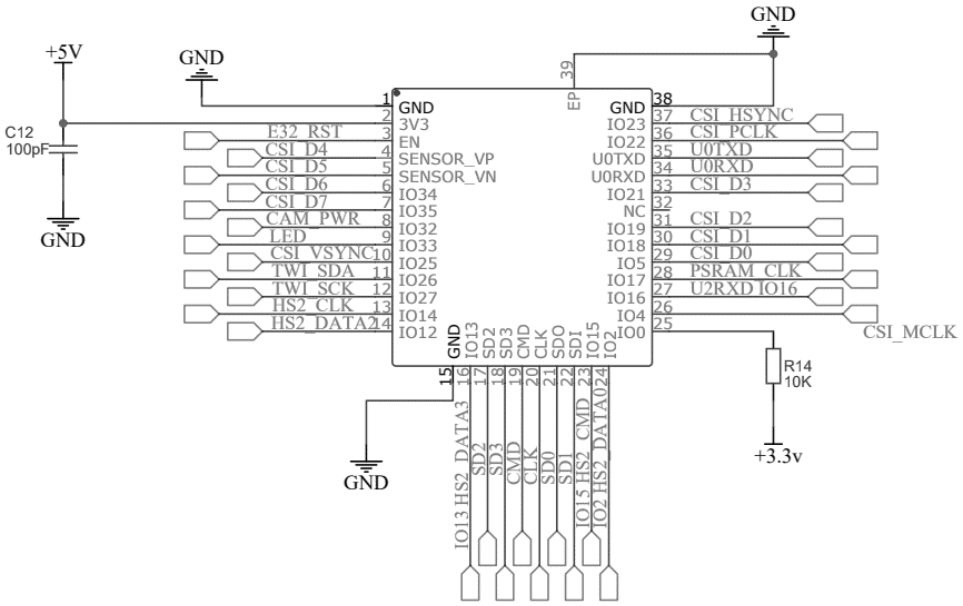


Рисунок 2.2 - Схематичне представлення ESP32.

Відеокамера OV2640 схематично представлена на рисунку 2.3.

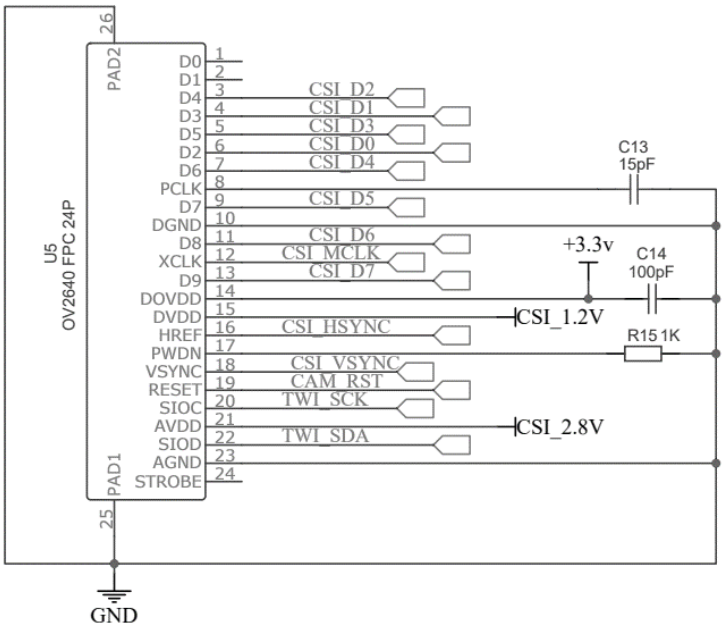


Рисунок 2.3 - Схематичне представлення камери OV2640.

В таблиці 2.3 показано, яким чином здійснюється з’єднання між камерою OV2640 та ESP32-CAM, а також наведено назву змінної в коді.

Таблиця 2.3-З’єднання камери OV2640 та ESP32-CAM

OV2640	ESP32	Назва змінної в коді
D0	GPIO 5	Y2_GPIO_NUM
D1	GPIO 18	Y3_GPIO_NUM
D2	GPIO 19	Y4_GPIO_NUM
D3	GPIO 21	Y5_GPIO_NUM
D4	GPIO 36	Y6_GPIO_NUM
D5	GPIO 39	Y7_GPIO_NUM
D6	GPIO 34	Y8_GPIO_NUM
D7	GPIO 35	Y9_GPIO_NUM
XCLK	GPIO 0	XCLK_GPIO_NUM
PCLK	GPIO 22	PCLK_GPIO_NUM
VSYNC	GPIO 25	VSYNC_GPIO_NUM
HREF	GPIO 23	HREF_GPIO_NUM
SDA	GPIO 26	SIOD_GPIO_NUM
SCL	GPIO 27	SIOC_GPIO_NUM
POWER PIN	GPIO 32	PWDN_GPIO_NUM

Для підключення ESP32-CAM до комп'ютера (при налаштуванні) та для забезпечення під’єднання живлення - використано перетворювач USB-TTL. Живлення модуля здійснюється від повербанка через USB-TTL, що дозволяє реалізувати стабільне та безперебійне його функціонування і забезпечити постійну трансляцію відеопотоку.

Схема електрична принципова наведена на кресленні КС КРБ 123.035.00.00Е3 (графічний матеріал).

2.3 Обґрунтування вибору програмного забезпечення

Програмне забезпечення (ПЗ) комп’ютеризованої системи відеонагляду з функцією ідентифікації ЕС на основі ШІ складається із:

- ПЗ налаштування та підтримки функціонування ESP32-CAM;
- ПЗ функції розпізнавання ЕС (пози людини «руки вгору» на основі ШІ (з використанням PoseNet));
- ПЗ інформування про ЕС (Telegram бот).

Для реалізації ПЗ налаштування та підтримки функціонування модуля ESP32-CAM використано Arduino IDE.

Налаштування ESP32-CAM за допомогою ArduinoIDE, здійснено за такою послідовністю [11]:

- Встановлення драйвера для USB-UART-перетворювача, який забезпечує зв'язок між ЕОМ та ESP32-CAM (для ОС Windows драйвери завантажено з офіційного сайту виробника чіпа USB-UART).
- Встановлення ArduinoIDE.
- Додавання до ArduinoIDE пакету для ESP32-CAM.
- Підключення ESP32-CAM до ЕОМ за допомогою USB-UART перетворювача та USB-кабелю.
- Увімкнення режиму завантаження в ESP32-CAM.

– Налаштування ArduinoIDE для роботи ESP32-CAM.

– Завантаження програми на ESP32-CAM.

– Відкриття монітора порту в ArduinoIDE. Для цього необхідно вибрати "Tools"-"Serial Monitor".

– Перевірка роботи ESP32-CAM.

Таке налаштування ESP32-CAM за допомогою ArduinoIDE дозволяє підключитися до мережі Wi-Fi та моніторити відео з камери OV2640 за допомогою браузера [11, 12].

На рисунку 2.4 наведено фрагмент коду налаштування та підтримки функціонування ESP32-CAM в КСВНІЕС.

```
143  void setup() {
144    Serial.begin(115200);
145    Serial.setDebugOutput(true);
146    Serial.println();
147    camera_config_t config;
148    config.ledc_channel = LEDC_CHANNEL_0;
149    config.ledc_timer = LEDC_TIMER_0;
150    config.pin_d0 = Y2_GPIO_NUM;
151    config.pin_d1 = Y3_GPIO_NUM;
152    config.pin_d2 = Y4_GPIO_NUM;
153    config.pin_d3 = Y5_GPIO_NUM;
154    config.pin_d4 = Y6_GPIO_NUM;
155    config.pin_d5 = Y7_GPIO_NUM;
156    config.pin_d6 = Y8_GPIO_NUM;
157    config.pin_d7 = Y9_GPIO_NUM;
158    config.pin_xclk = XCLK_GPIO_NUM;
159    config.pin_pclk = PCLK_GPIO_NUM;
160    config.pin_vsync = VSYNC_GPIO_NUM;
161    config.pin_href = HREF_GPIO_NUM;
162    config.pin_sscb_sda = SIOD_GPIO_NUM;
163    config.pin_sscb_scl = SIOC_GPIO_NUM;
164    config.pin_pwdn = PWDN_GPIO_NUM;
165    config.pin_reset = RESET_GPIO_NUM;
166    config.xclk_freq_hz = 20000000;
167    config.pixel_format = PIXFORMAT_JPEG;
168
169    // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
170    //                      for larger pre-allocated frame buffer.
171    if(psramFound()){
172      config.frame_size = FRAMESIZE_QVGA;
173      config.jpeg_quality = 10;
174      config.fb_count = 2;
175    } else {
```

Рисунок 2.4 – Фрагмент коду програми налаштування модуля ESP32-CAM

В Додатку Д наведено лістинг програми для налаштування та підтримки функціонування ESP32-CAM.

2.4 Обґрунтування методу реалізації функції ідентифікації екстреної ситуації

Головним завданням, яке виконується системою є ідентифікація ЕС, тобто, визначення пози людини «руки вгору» (див. п.1.2). Тому детальніше описано реалізацію саме цієї функції.

Розпізнавання пози використовує позу та орієнтацію для прогнозування та відстеження розташування людини чи об’єкта. Відповідно, розпізнавання пози дозволяє програмам оцінювати просторове розташування тіла («пози») на зображенні чи відео. Визначення пози здійснюється шляхом пошуку ключових точок людини чи об’єкта. Взявши, наприклад, людину, ключовими точками будуть такі суглоби, як лікті, коліна, зап’ястя тощо.

PoseNet — це модель тензорного потоку глибокого навчання, яка визначає позу людини, оцінюючи частини тіла, визначені як ключові точки (в цій моделі - 17), починаючи від очей, вух і закінчуючи щиколотками та колінами, включаючи зап’ястя, лікті, плечі та ніс. Ці точки з’єднані так, щоб утворити скелетну структуру тіла. Також для всіх точок і самої пози оцінюється достовірність [10].

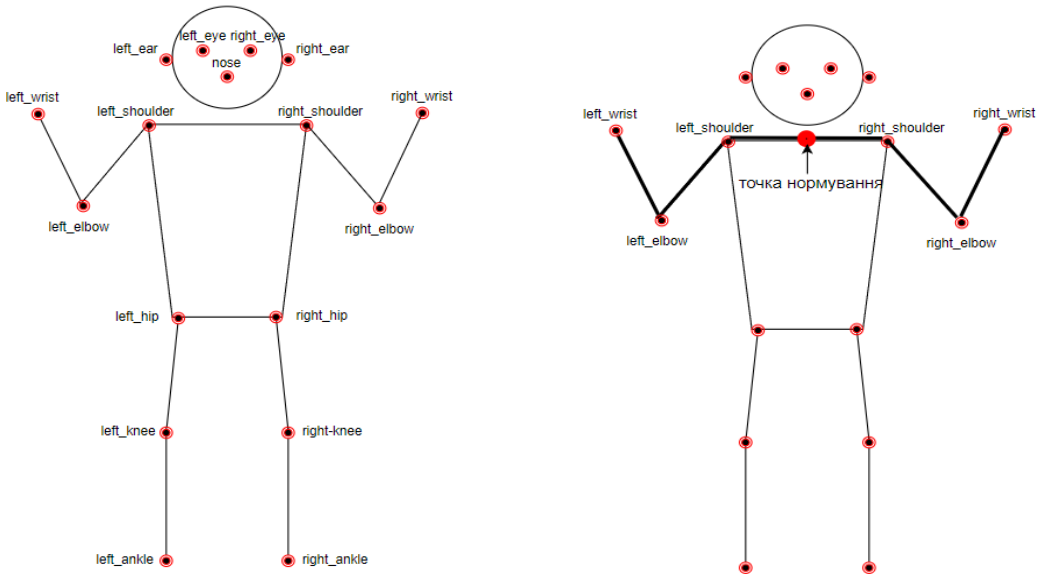
Оцінювання пози людини дозволяє визначити 17 різних ключових точок (класів): "nose" ("ніс"), "left_eye" ("ліве_око"), "right_eye" ("праве_око"), "left_ear" ("ліве_вухо"), "right_ear" ("праве_вухо"), "left_shoulder" ("ліве_плече"), "right_shoulder" ("праве_плече"), "left_elbow" ("лівий_лікоть"), "right_elbow" ("правий_лікоть"), "left_wrist" ("ліве_зап'ястя"), "right_wrist" ("праве_зап'ястя"), "left_hip" ("ліве_стегно"), "right_hip" ("праве_стегно"), "left_knee" ("ліве_коліно"), "right-knee" ("праве_коліно"), "left_ankle" ("ліва_щиколотка"), «right_ankle» («права_щиколотка») [10].

Реалізація функції розпізнавання ЕС (пози «руки вгору») на основі ШІ (з використанням PoseNet) здійснюється за таким алгоритмом:

– розкадровка відео, отриманого з камери,
– розпізнавання ключових точок (суглобів),
– визначення координат шести інформативних точок,
– нормування даних,
– розпізнавання пози «руки вгору».

Основними інформативними точками за якими виконується розпізнавання пози «руки вгору» є: "left_shoulder" ("ліве_плече"), "right_shoulder" ("праве_плече"), "left_elbow" ("лівий_лікоть"), "right_elbow" ("правий_лікоть"), "left_wrist" ("ліве_зап'ястя"), "right_wrist" ("праве_зап'ястя").

На рисунку 2.5 показано розташування ключових точок PoseNet: а) - усі 17 ключових точок (класів), б) – запропоновані 6 інформативних точок для визначення пози «руки вгору».



а) б)

Рисунок 2.5 - Розташування ключових точок PoseNet:
а) зображення усіх (17) ключових точок PoseNet;
б) запропоновані інформативні ключові точки для визначення пози «руки вгору».

Тобто запропоновано в КСВНІЕС для визначення ЕС враховувати тільки розташування 6 точок і розпізнавання пози «руки вгору» здійснювати відповідно до їх розташування. Таке рішення дозволяє зменшити обчислювальні ресурси системи, оскільки висновок приймається із врахуванням меншої кількості даних (тобто замість 17 точок в одному кадрі зображення - аналізуються тільки 6), відповідно обчислювальна складність завдання є суттєво меншою.

Кожна ключова точка при визначенні PoseNet характеризується трьома числами (*x*,*y*,*v*), де *x* і y позначають координати, а *v* - вказує, чи видима ключова точка.

Враховуючи те, що розташування ключових інформативних точок в кадрі може бути будь-де, тому здійснюється нормування їх координат відносно точки, яка знаходиться на середині відстані між плечима (див.рис.2.5 б)).

PoseNet працює з монокулярними зображеннями, а також стійкий до розмиття в русі та умов освітлення (якщо вони постійно змінюються). Крім того, не вимагає доступу до параметрів камери, хорошої ініціалізації та ручної роботи.

Визначення пози відбувається, коли вхідне RGB-зображення опрацьовується за допомогою застосування згорткової нейронної мережі, до якої застосовується алгоритм для однієї або кількох поз, або, який дає вихідні дані моделі через позначення пози на зображенні, оцінки її достовірності, положення ключових точок і оцінки достовірності ключових точок. Якщо зображення, надане PoseNet, є нечітким, тоді відображається оцінка достовірності, вказуючи, наскільки впевнено він вгадує конкретну позу на зображенні. Це здійснюється у формі відповіді JSON.

Оцінювання достовірності пози визначає загальну точність пози за допомогою впевненості в оцінці оцінки, яка змінюється від 0,0 до 1,0. Оцінювання впевненості ключової точки визначає впевненість у точності

оціненої ключової точки, яка також коливається від 0,0 до 1,0. Аналогічно до показника впевненості пози, його використовують, щоб приховати ключові точки, якщо вони вважаються недостатньо сильними [10].

Положення ключової точки - вхідне зображення, яке надано моделі, розділене на ключові точки, які є *2D(x,y)* координатами, що визначають корисні ключові точки в усьому зображенні [10].

У PoseNet є три функції, а саме «Поза», «Орієнтація» та «Повнозв'язаний рівень», які зв'язані з усіма початковими моделями, конвуляційним процесом і шаром FC із 2048 одиниць поверх останнього початкового модуля [10].

Архітектура PoseNet [10] показана на рисунку 2.6.

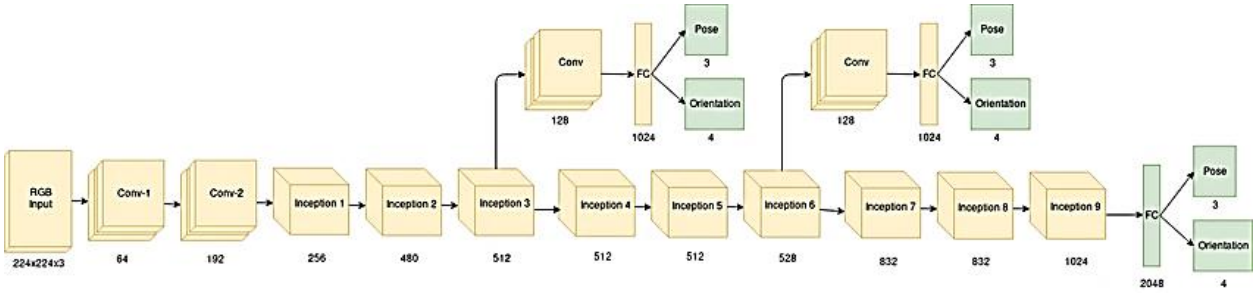


Рисунок 2.6 - Архітектура PoseNet

Вхідний, конвуляційний і об'єднаний рівень вилучають частину ознак, що є результатом процесу згортання, решта ознак є вхідними даними до повного підключеного рівня, який передбачає різні класи різних зображень.

Цей рівень перетворює вхідні дані, застосовуючи лінійні перетворення кілька разів, які надсилаються до функції активації для прогнозування класу. Потім, за допомогою функцій Sigmoid, softmax, ReLu, отримуємо класи об'єкта, введеного в модель. Хоча softmax замінює 2 паралельні повністю зв'язані шари на 3 або 4 одиниці відповідно. Це, як правило, регресія до пози (продемонстрована за *(x,y,z)* координатами) та орієнтація (представлена як кватерніон) [10].

PoseNet може розпізнати пози однієї або кількох осіб, а також дозволяє визначити позицію об’єктів у реальному часі за допомогою оцінки пози з використанням бібліотек ml5.js, ps5.js або Tensorflow.js. Модель PoseNet була навчена за допомогою MobileNet Architecture. MobileNet — це згорткова нейронна мережа, розроблена Google на основі набору даних ImageNet, яка, в основному, використовується для класифікації зображень і цільової оцінки за допомогою показників достовірності.

Перевагами PoseNet є те, що ця модель є легкою моделлю, яка використовує розділену по глибині конвулюцію для поглиблення мережі за рахунок зменшення параметрів, витрат на обчислення, що підвищує точність моделі. Таким чином, будучи легкою моделлю, вона легко запускається у веб-браузерах та на смартфонах. Це вирізняє її від інших, залежних від API бібліотек.

2.5 Алгоритм роботи системи

Важливим етапом роботи, що забезпечує систематизацію, оптимізацію та надійність у виконанні поставлених перед КСВНІЕС завдань є побудова алгоритму роботи. Враховуючи запропоновані в роботі рішення розроблено алгоритм роботи КСВНІЕС. Відповідно до якого, отриманий з модуля ECP32-CAM відеопотік спочатку проходить попереднє опрацювання, далі - розбивається на окремі кадри (розкадровується), потім в кожному кадрі, з використанням глибоких нейронних мереж (PoseNet), визначається наявність людини та розпізнаються ключові точки (суглоби). Якщо людини в кадрі немає, тоді аналізується наступний кадр. Наступним етапом є виділення шести інформативних точок (див.рис.2.5 б)) та здійснення їх нормування, далі, відповідно до розташування цих точок, навчена модель приймає рішення чи в

людини поза «руки вгору». Якщо система виявила цю позу, тоді надсилається повідомлення про ЕС, якщо ні – розглядається наступний кадр.

Алгоритм роботи КСВНІЕС показано на рисунку 2.7.

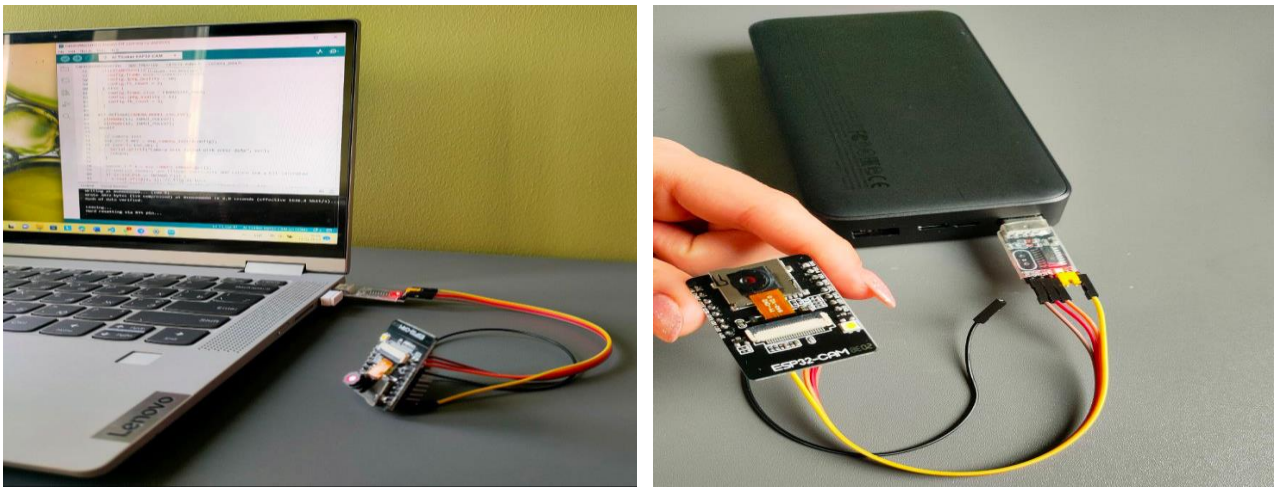
Рисунок 2.7– Алгоритм роботи КСВНІЕС

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Реалізація проєктних рішень

Проєктні рішення, запропоновані для реалізації КСВНІЕС гарантують чудову основу для подальшого вдосконалення та розробки систем відеонагляду, які можна використовувати в різноманітних сферах. ESP32-CAM, на якому побудована система, — недорогий малопотужний мікроконтролер із високопродуктивним модулем камери. Живлення його здійснюється від повербанка, що дозволяє реалізувати стабільне та безперебійне його функціонування і забезпечити трансляцію відеопотоку.

На рисунку 3.1 показано модуль ESP32-CAM, який використовується в складі системи, зокрема, процес налаштування ESP32-CAM та ESP32-CAM із живленням від повербанка через USB-TTL.



а) б)

Рисунок 3.1 – Модуль ESP32-CAM:
а) процес налаштування ESP32-CAM; б) ESP32-CAM із живленням.

Загальний вигляд КСВНІЕС показано рисунку 3.2.

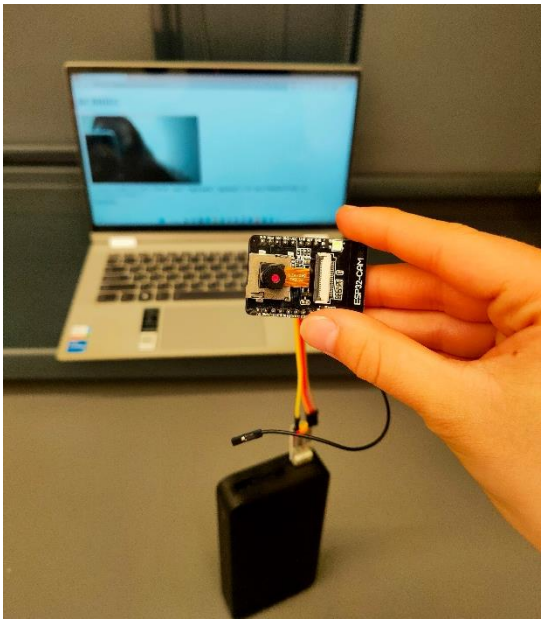


Рисунок 3.2 – Загальний вигляд КСВНІЕС

На рисунку 3.3 показано демонстрацію роботи КСВНІЕС, зокрема виконання функції розпізнавання пози. Для перевірки коректності роботи системи та з метою тестування, реалізовано вивід на екран значень, отриманих за допомогою PoseNet.

Рисунок 3.3. – Демонстрація роботи КСВНІЕС

3.2 Реалізація функції ідентифікації екстрених ситуацій в комп'ютеризованій системі методами машинного навчання та її тестування

Для розпізнавання екстреної ситуації комп'ютеризованою системою використано модель на основі ШІ PoseNet. Для виконання цього завдання реалізовано такі етапи: зйомка зразків відео; розбиття відео на кадри; класифікація зображень; виявлення особи; розпізнавання пози «руки вгору».

Оскільки в інтернеті набору даних (датасету) людей із заданою позою не знайдено, тому для розробки і тестування запропонованої КСВНІЕС сформовано свій - із 1510 зображень, з яких 930 – поза «руки вгору», 580 – пози з іншим розташуванням рук. Їх отримано із знятих відео, в яких люди різного віку та статі, в різних приміщеннях та на різній відстані від реєстратора здійснюють рухи руками, в тому числі піднімають руки вгору, імітуючи позу, яку має розпізнати система, вважаючи її екстреною ситуацією.

На рисунку 3.4 показано незначну частину зображень із сформованого набору даних, використаних для навчання, зокрема, на рис.3.4. а) – із позою «руки вгору», б) –з іншим розташуванням рук.

На рисунку 3.5 показано визначення ключових точок зображень за допомогою моделі PoseNet.

КСВНІЕС, використовуючи PoseNet, здійснює розпізнавання людей на зображеннях, а потім – розпізнавання ключових точок (суглобів). Наступним етапом є отримання координат 6 інформативних точок та здійснення їх нормування для подальшого аналізу.

а)

б)

Рисунок 3.4 – Приклад зображень з набору даних:

а) поза «руки вгору», б) поза з іншим розташуванням рук.

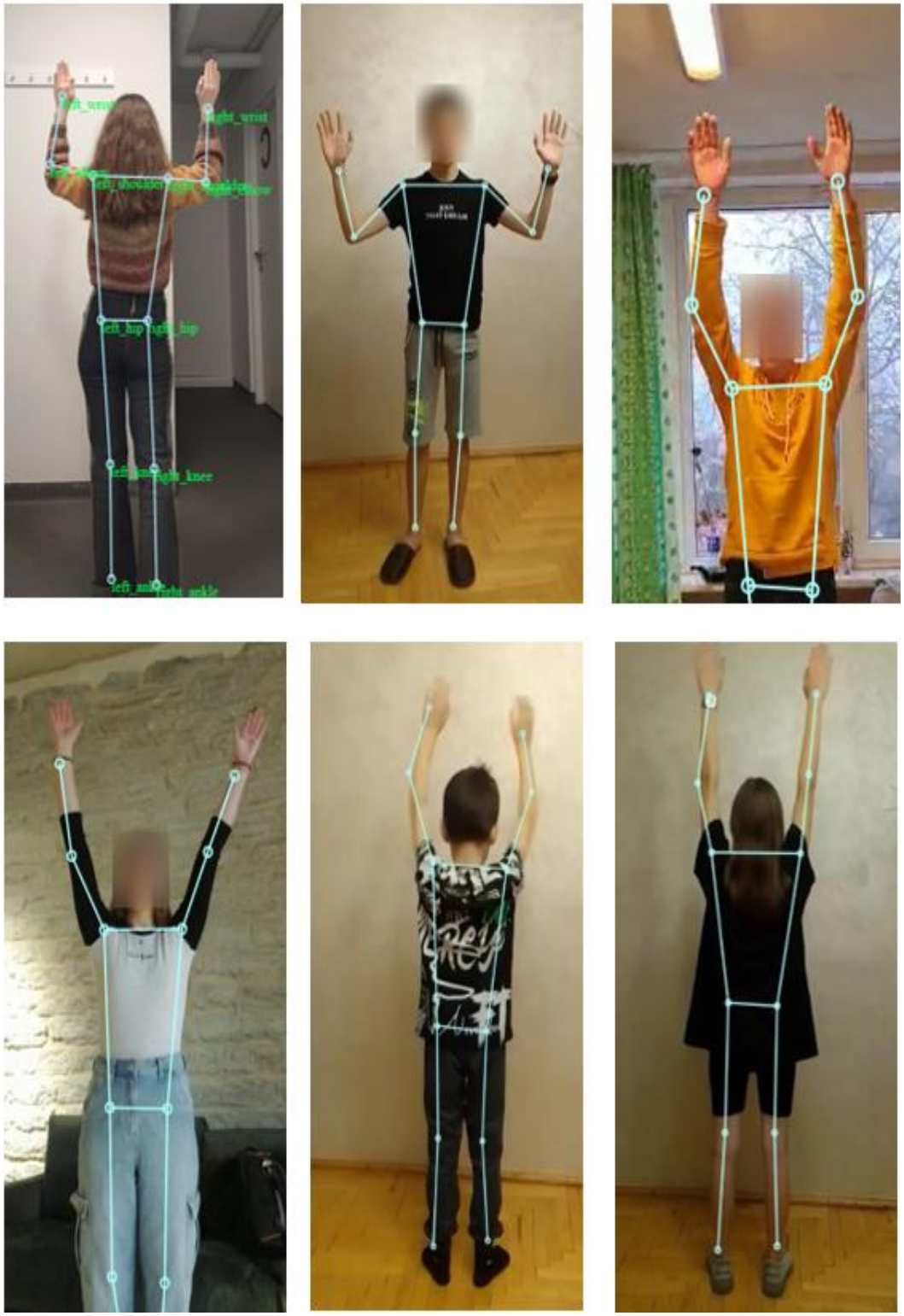


Рисунок 3.5 – Визначення ключових точок зображень

На рисунку 3.6. показано приклад отриманих на цьому етапі даних.

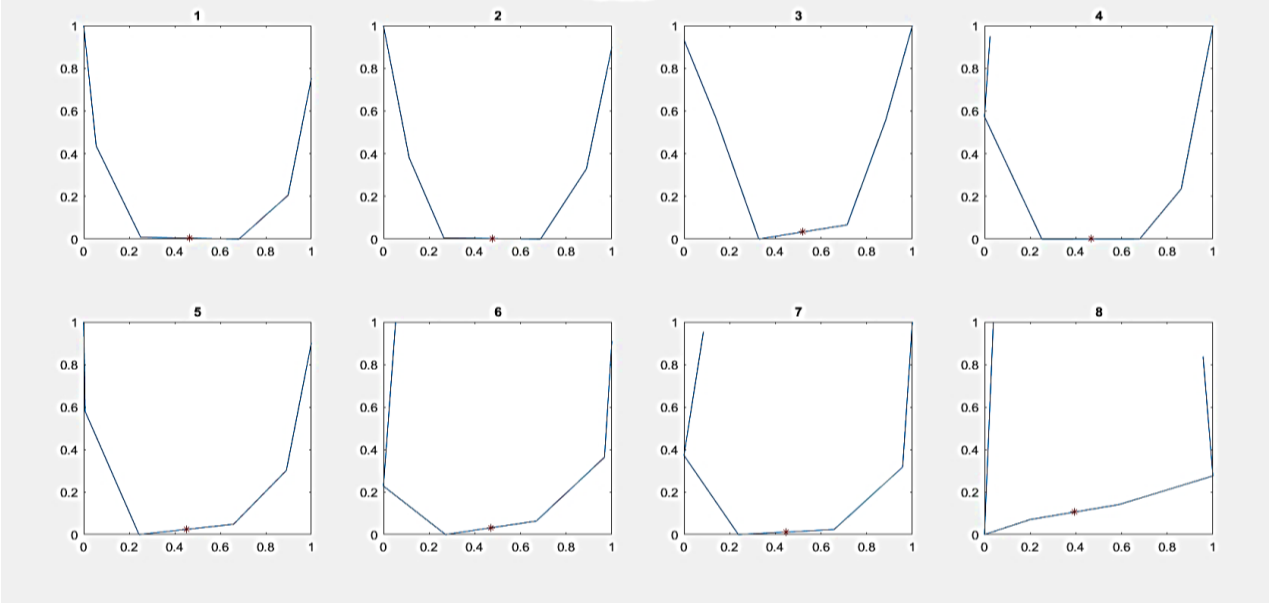
xreMain	leftWrist_x	leftElbow_x	leftShoulder_x	rightShoulder_x	rightElbow_x	rightWrist_x	BetweenShouldersP	leftWrist_y	leftElbow_y	leftShoulder_y	rightShoulder_y	rightElbow_y	rightWrist_y	BetweenSho
0.7714958752	0	0.05483512668	0.2503533115	0.682188555	0.8979213524	1	0.4662709333	1	0.435858303	0.008096114532	0	0.2053825574	0.7526105809	0.004048
0.6669495815	0	0.1123993615	0.2643069427	0.6880885571	0.8885039676	1	0.4761977499	1	0.3807012975	0.005053555864	0	0.3287788707	0.8993192406	0.002526
0.7674372709	0	0.1412016123	0.3274854677	0.7144388373	0.8836881856	1	0.5209621525	0.9340416083	0.5638027676	0	0.06633564574	0.5588190877	1	0.033167
0.7895340096	0.02439043702	0	0.2509015424	0.6798260374	0.8613343301	1	0.4653637899	0.9502593413	0.5731430186	0	0.0006553321229	0.235015878	1	0.00032766
0.7360977278	0	0.006470977753	0.2427450061	0.6581494115	0.8901979329	1	0.4504472088	1	0.578360585	0	0.04827904108	0.3014166321	0.9011057886	0.024139
0.785615488	0.05328311575	0	0.271621495	0.6691448709	0.9677428318	1	0.470383183	1	0.2275967901	0	0.06333515715	0.3633890375	0.9095628558	0.031667
0.6988137128	0.08512620505	0	0.2379532901	0.6572392637	0.9573091334	1	0.4475962769	0.9529115985	0.3718917332	0	0.02342167596	0.3176680267	1	0.011710
0.6805598376	0.03885818795	0	0.2000825515	0.5876174988	1	0.9568249731	0.3938500252	1	0	0.07054167768	0.1407463523	0.2757790668	0.8370634241	0.1056
0.7122488216	0.2088206051	0	0.2880468918	0.6833827406	1	0.9208267814	0.4857148162	0.7225401195	0.06633369637	0	0.004972448639	0.2612369446	1	0.002486
0.7038609831	0.1395972949	0	0.3028965464	0.7280910621	0.9313483885	1	0.5154938042	1	0.2632647672	0.7124032983	0.6968068281	0	0.9893617925	0.70460
0.7076499505	0.1685263509	0	0.2703465571	0.6880133515	1	0.9560660748	0.4791799543	0.8366669823	0.3537115863	0	0.005016974171	0.2049948227	1	0.0025084
0.6863535213	0.08692259118	0	0.2626791124	0.6894321826	1	0.9821244011	0.4760556475	1	0.4141285419	0	0.02430588405	0.3083177709	0.9803870669	0.012152
0.7207919571	0.0001799916563	0	0.2339894745	0.6551670836	0.909741706	1	0.444578279	1	0.4174550571	0.01844704652	0	0.2998274156	0.9911696169	0.009223
0.6877990282	0.1006185054	0	0.2907498965	0.7900225734	1	0.7680455507	0.5403862349	1	0.1397338059	0.2672545385	0	0.2913398275	0.3482477305	0.13362
0.8319355109	0.1642517132	0	0.2850407152	0.7576237144	1	0.7403051342	0.5213322148	0.0129659376	0.5143817644	1	0.9927678529	0.6760709861	0	0.99638
0.8092159156	0.1505725895	0	0.294090667	0.7428562854	1	0.7546502373	0.5184734762	0.7172416801	0.6295025	0.1856283275	0.3203242969	0	1	0.25297
0.7264905598	0.1432544947	0	0.2635149775	0.7488645759	1	0.7267850001	0.5061897767	0	0.3435458489	0.7153293759	0.03649620264	0.8478406047	1	0.37591
0.642718476	0.2628783241	0	0.2646549053	0.6942640712	1	0.7079487682	0.4794594882	1	0.410103018	0.1679912018	0	0.3804105056	0.7929880422	0.083995
0.6864469398	0.2557572855	0	0.2577498338	0.7418583472	1	0.757446903	0.4998040905	0.9829616483	0.707334975	0.2066163546	0	0.525275533	1	0.10330
0.7778727237	0	0.0150986798	0.2922149238	0.6851286804	1	0.7056104141	0.4886718021	0.392971956	0.180133951	0	0.04578787767	0.04480712795	1	0.022893
0.647496489	0.2875723312	0	0.2781957786	0.7541760812	1	0.7263749282	0.5161859299	1	0.3077485783	0	0.09331542904	0.07513206522	0.4086089157	0.046657
0.6763492644	0.2417045536	0	0.2918290153	0.7696850365	1	0.6818948849	0.5307570259	1	0.0278673544	0	0.001114491787	0.5286144559	0.9840454446	0.00055724
0.5677545995	0.09912079478	0	0.3424221737	0.7260423733	0.8904934499	1	0.5342322735	1	0	0.4708977803	0.5401364414	0.1213460008	0.7855161609	0.50551
0.4922922394	0	0.02306687719	0.2277329407	0.7149172761	0.9178071887	1	0.4713251084	1	0.3984094992	0.02309657105	0	0.3473284068	0.8128870322	0.011548

Рисунок 3.6 – Приклад отриманих нормованих даних.

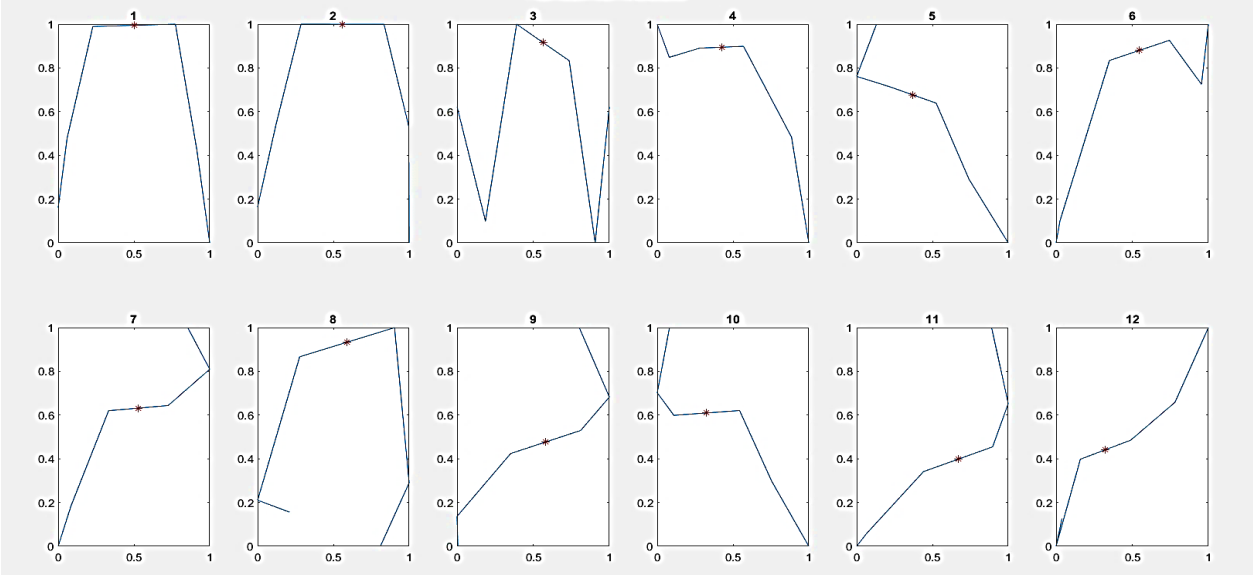
На рисунку 3.7. показано візуалізовані нормовані дані (а – руки вгору, б – інше розташування рук).

Таким чином, сформований набір даних, який використано для дослідження, складається із 1510 зображень: в тому числі 930 - «руки вгору» («0») та 580 – інші пози, тобто не «руки вгору» («1»).

Наступним етапом є ідентифікація екстреної ситуації КСВНІЕС. Визначення конкретної пози, в даному випадку – пози «руки вгору», є складним завданням, яке потребує подальших досліджень з використанням різних підходів та моделей штучного інтелекту.



а)



б)

Рисунок 3.7 – Візуалізовані нормовані дані:

а) нормовані дані пози «руки вгору»;

б) нормовані дані розташування рук в різних позах крім пози «руки вгору»

В цій роботі розглянуто підхід, враховуючи, що класів даних (розташування рук для визначення пози) є два: поза «руки вгору» (ідентифікатор EC) та будь-яка інша. Тому вирішується, по суті, завдання бінарної класифікації. Для цього доцільно використати метод опорних векторів, тобто SVM (Support Vector Machine).

Класифікатор спочатку «навчається» на об'єктах з навчальної вибірки (із сформованого набору даних), яким наперед задані мітки класів. Потім навчений алгоритм для кожного об'єкта передбачає мітку класу з тестової вибірки [13].

Алгоритм SVM використовує набір функцій, які визначені як ядро (функція, яка надається алгоритму машинного навчання). В даному випадку використано ядра Гауса. SVM, максимізуючи відстань між точками та знаходячи найкращу гіперплощину, розділяє дані на різні категорії, відповідно до набору математичних функцій [13]:

$$maximize \quad f(\mathrm{c}_1 \dots \mathrm{c}_n) = \sum_{i=1}^{n} \mathrm{c}_i - \frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} y_i \mathrm{c}_i (x_i \cdot x_j) y_j \mathrm{c}_j,$$

$$subject \quad to \quad \sum_{i=1}^{n} c_i y_i = 0, \quad and \quad 0 \le c_i \le \frac{1}{2n\lambda} \quad for \ all \ i.$$

Функції ядра дозволяють створити лінійні дискримінанти на нелінійних різноманіттях, що дає вищу точність та надійність, ніж традиційні лінійні моделі [14, 15].

Таким чином, для виконання нелінійної класифікації пози в КСВНІЕС використано SVM з ядрами Гауса. Враховуючи набір даних датасету, де кожен елемент належить до однієї з 2 категорій (поза «руки вгору» та інша), здійснюється навчання SVM, а потім – перевіряється класифікатор. Реалізація цього методу виконана в MatLab, фрагмент коду наведено на рисунку 3.8.

```
%%
clc, clearvars -except  X Y

% Xup = X([~logical(Y')], :);  % Class: 0
% Xd  = X([logical(Y')], :);   % Class: 1

% Train SVM Classifiers Using a Gaussian Kernel
% Train an SVM classifier with KernelFunction set to 'rbf' and BoxConstraint set to Inf.

cl = fitcsvm(X,Y,'KernelFunction','rbf',...
    'BoxConstraint',Inf,'ClassNames',[0,1]);

[y,scores] = predict(cl,X)

%%
labels = Y;
predicted_labels = y;

figure
cm = confusionchart(labels,predicted_labels);
cm.Title = 'True/Predicted Classes with summary';

cm.RowSummary = 'row-normalized';
cm.ColumnSummary = 'column-normalized';
cm.Normalization = 'absolute';

%%

% plot ROCs

ClassNames = { '0', '1' }
OPTROCPTs=[];
Ts=[];
```

Рисунок. 3.8 – Фрагмент коду використання SVM з ядрами Гауса

На рисунку 3.9 наведена отримана матриця невідповідностей.

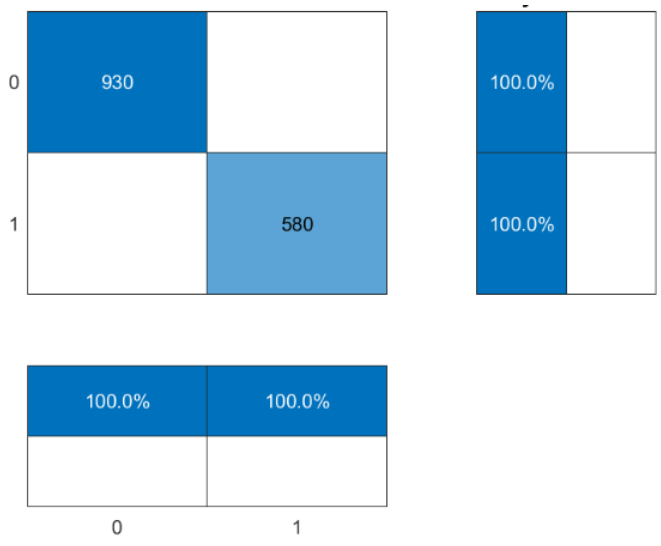
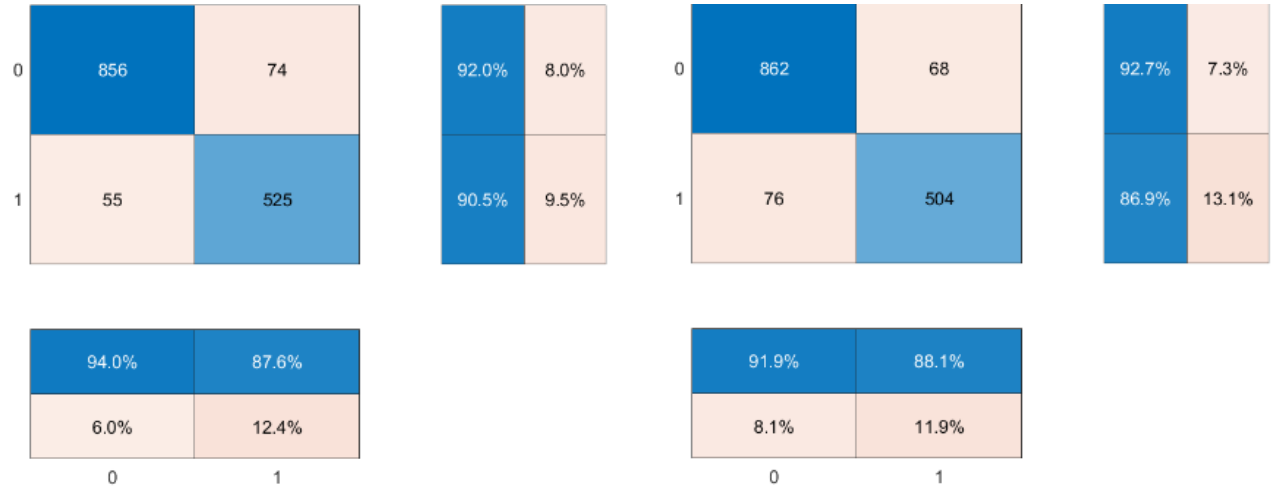


Рисунок 3.9 – Матриця невідповідностей SVM з ядрами Гауса.

Ця матриця відображає загальну кількість даних у кожній клітинці, її рядки відповідають правильному класу, а стовпці - прогнозованому класу [14]. Діагональ -відповідає правильно класифікованим класам (930 – для пози «руки вгору», 580 – для іншої). Рядок у низу рисунку показує % усіх даних, які належать до правильно та неправильно класифікованого класу. Стовпець у правому кутку рисунка показує % усіх даних, які належать відповідно до кожного класу, які правильно чи неправильно класифіковані, тобто точність [14, 15]. Отже, метод SVM з ядрами Гауса показав точність 100%.

Для порівняння побудовано ще також моделі машинного навчання на основі таких методів класифікації як: метод логістичної регресії та метод k-найближчих сусідів, та оцінено їх точність. Матриці невідповідності цих методів показані на рисунку 3.10: а) метод логістичної регресії, б) метод k-найближчих сусідів.



а) б)

Рисунок 3.10 – Матриця невідповідностей:
а) метод логістичної регресії, б) – метод k-найближчих сусідів.

Точність методу логістичної регресії при класифікації пози «руки вгору» становила 92%, а методу k-найближчих сусідів – 92,7%.

Таким чином, найкращі результати для класифікації пози в КСВНІЕС показав метод SVM з ядрами Гауса.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Надзвичайні ситуації: визначення, причини, класифікація

Надзвичайна ситуація - це непередбачувана подія або обставина, яка порушує нормальні умови життя і діяльності людей та може мати негативні наслідки для життя, здоров'я, майна або навколишнього середовища. Визначення причини та класифікація надзвичайних ситуацій допомагають зрозуміти їх природу і ефективніше реагувати на них [17].

Причини надзвичайних ситуацій можуть бути різними. Вони можуть включати природні явища, такі як стихійні лиха (повені, землетруси, урагани), техногенні аварії (промислові аварії, ядерні аварії, вибухи), соціальні конфлікти (терористичні акти, масові заворушення) та інші обставини.

Загальні ознаки надзвичайних ситуацій: наявність або загроза загибелі людей чи значне погіршення умов їх життєдіяльності; заподіяння економічних збитків; істотне погіршення стану довкілля. До надзвичайних ситуацій, як правило, призводять аварії, катастрофи, стихійні лиха та інші події, такі, як епідемії, терористичні акти, збройні конфлікти тощо [17, 18].

Аварія – це небезпечна подія техногенного характеру, що створює на об'єкті, території або акваторії загрозу для життя і здоров'я людей і призводить до руйнування будівель, споруд, обладнання і транспортних засобів, порушення виробничого або транспортного процесу чи завдає шкоди довкіллю. Це – вихід з ладу машин, механізмів, пристроїв, комунікацій внаслідок порушення технології виробництва, правил експлуатації, правил безпеки, помилок, які допущені при проектуванні, будівництві, а також внаслідок стихійних лих.

Події природного походження або результат діяльності природних процесів, які за своєю інтенсивністю, масштабом поширення і тривалістю можуть вражати людей, об'єкти економіки та довкілля, називаються небезпечними природними явищами. Руйнівне небезпечне природне явище – це стихійне лихо.

Надзвичайні ситуації мають різні масштаби за кількістю жертв, кількістю людей, що стали хворими чи каліками, кількістю людей, яким завдано моральної шкоди, за розмірами економічних збитків, площею території, на якій вони розвивались, тощо.

Вагомість надзвичайної ситуації визначається передусім кількістю жертв та ступенем впливу на оточуюче життєве середовище, тобто рівнем системи «людина – життєве середовище», якої вона торкнулася, і розміром шкоди, завданої цій системі. Виходячи з ієрархії таких систем поділяють на [18]:

– індивідуальні надзвичайні ситуації, коли виникає загроза для порушення життєдіяльності лише однієї особи;

– надзвичайні ситуації рівня мікроколективу, тобто коли загроза їх виникнення чи розповсюдження наслідків стосується сім'ї, виробничої бригади, пасажирів одного купе;

– надзвичайні ситуації рівня колективу;

– надзвичайні ситуації рівня макроколективу;

– надзвичайні ситуації для жителів міста, району;

– надзвичайні ситуації для населення області;

– надзвичайні ситуації для населення країни;

– надзвичайні ситуації для жителів континенту;

– надзвичайні ситуації для всього людства.

Як правило, чим більшу кількість людей обходить надзвичайна ситуація, тим більшу територію вона охоплює. І навпаки, при більшій площі поширення катастрофи чи стихійного лиха від нього страждає більша кількість людей.

Через це в основу існуючих класифікацій надзвичайних ситуацій за їх масштабом найчастіше кладуть територіальний принцип, за яким надзвичайні ситуації поділяють на локальні, об'єктові, місцеві, регіональні, загальнодержавні (національні), континентальні та глобальні (загальнопланетарні) [17].

Локальні надзвичайні ситуації відповідають рівню системи з однією особою та мікроколективом; об'єктові – системам з рівнем колектив, макроколектив; місцеві – системам, в які входить населення міста або району; регіональні – області; загальнодержавні – населення країни і так далі.

Надзвичайна ситуація загальнодержавного рівня – це надзвичайна ситуація, яка розвивається на території двох та більше областей або загрожує транскордонним перенесенням, а також у разі, коли для її ліквідації необхідні матеріали і технічні ресурси в обсягах, що перевищують власні можливості окремої області, але не менше одного відсотка обсягу видатків відповідного бюджету.

Надзвичайна ситуація регіонального рівня – це надзвичайна ситуація, яка розвивається на території двох або більше адміністративних районів (міст обласного значення) Автономної Республіки Крим, областей, міст Києва та Севастополя або загрожує перенесенням на територію суміжної області України, а також у разі, коли для її ліквідації необхідні матеріальні і технічні ресурси в обсягах, що перевищують власні можливості окремого району, але не менше одного відсотка обсягу видатків відповідного бюджету.

Надзвичайна ситуація місцевого рівня – це надзвичайна ситуація, яка виходить за межі потенційно небезпечного об'єкта, загрожує поширенням самої ситуації або її вторинних наслідків на довкілля, сусідні населені пункти, інженерні споруди, а також у разі, коли для її ліквідації необхідні матеріальні і технічні ресурси в обсягах, що перевищують власні можливості потенційно небезпечного об'єкта, але не менше одного відсотка обсягу видатків

відповідного бюджету. До місцевого рівня також належать всі надзвичайні ситуації, які виникають на об'єктах житлово-комунальної сфери та інших, що не входять до затверджених переліків потенційно небезпечних об'єктів [17].

Надзвичайна ситуація об'єктового рівня – це надзвичайна ситуація, яка не підпадає під зазначені вище визначення, тобто така, що розгортається на території об'єкта або на самому об'єкті і наслідки якої не виходять за межі об'єкта або його санітарно-захисної зони.

Для організації ефективної роботи із запобігання надзвичайним ситуаціям, ліквідації їхніх наслідків, зниження масштабів втрат та збитків дуже важливо знати причини їх виникнення та володіти теорією виникнення катастроф. Положення про класифікацію надзвичайних ситуацій за характером походження подій, котрі зумовлюють виникнення надзвичайних ситуацій на території України, розрізняє чотири класи надзвичайних ситуацій — надзвичайні ситуації техногенного, природного, соціально-політичного, військового характеру. Кожен клас надзвичайних ситуацій поділяється на групи, які містять конкретні їх види [18].

Надзвичайні ситуації техногенного характеру – це транспортні аварії (катастрофи), пожежі, неспровоковані вибухи чи їх загроза, аварії з викидом (загрозою викиду) небезпечних хімічних, радіоактивних, біологічних речовин, раптове руйнування споруд та будівель, аварії на інженерних мережах і спорудах життєзабезпечення, гідродинамічні аварії на греблях, дамбах.

Надзвичайні ситуації природного характеру – це небезпечні геологічні, метеорологічні, гідрологічні морські та прісноводні явища, деградація ґрунтів чи надр, природні пожежі, зміна стану повітряного басейну, інфекційна захворюваність людей, сільськогосподарських тварин, масове ураження сільськогосподарських рослин хворобами чи шкідниками, зміна стану водних ресурсів та біосфери.

Надзвичайні ситуації соціально-політичного характеру – це ситуації, пов'язані з протиправними діями терористичного та антиконституційного спрямування: здійснення або реальна загроза терористичного акту (збройний напад, захоплення і затримання важливих об'єктів, ядерних установок і матеріалів, систем зв'язку та телекомунікацій, напад чи замах на екіпаж повітряного чи морського судна), викрадення (спроба викрадення) чи знищення суден, встановлення вибухових пристроїв у громадських місцях, викрадення або захоплення зброї, виявлення застарілих боєприпасів тощо.

Надзвичайні ситуації воєнного характеру – це ситуації, пов'язані з наслідками застосування зброї масового ураження або звичайних засобів ураження, під час яких виникають вторинні фактори ураження населення внаслідок зруйнування атомних і гідроелектричних станцій, складів і сховищ радіоактивних і токсичних речовин та відходів, нафтопродуктів, вибухівки, сильнодіючих отруйних речовин, токсичних відходів, нафтопродуктів, транспортних та інженерних комунікацій.

4.2 Соціальне значення охорони праці

Закон України «Про охорону праці» визначає «Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально профілактичних заходів і засобів, спрямованих на збереження здоров’я та працездатності людини в процесі праці».

Завдання охорони праці - забезпечення безпечних, нешкідливих і сприятливих умов праці через вирішення багатьох складних завдань [19].

Соціальне значення охорони праці включає в себе різні аспекти, пов'язані з безпекою та благополуччям працівників на робочому місці. Основні аспекти соціального значення охорони праці включають: здоров'я та безпеку працівників, покращення умов праці, зниження виробничого травматизму і

профзахворювань, збільшення продуктивності, захист прав працівників, збереження людського капіталу, підвищення задоволеності працівників, зміцнення соціального діалогу, що - забезпечення сталого розвитку суспільства.

Охорона праці спрямована на запобігання травмам, хворобам та іншим негативним наслідкам, пов'язаним з виконанням трудових обов’язків. Це допомагає зберегти фізичне та психологічне здоров'я працівників, покращує їх якість життя та сприяє загальному благополуччю [19].

Охорона праці сприяє створенню безпечних та здорових умов праці. Це може включати належне оснащення робочого місця, забезпечення належної вентиляції, освітлення та ергономіки, а також зменшення шкідливих чинників, таких як шум, вібрація та отруйні речовини. Здорові та безпечні робочі умови сприяють збільшенню продуктивності працівників, тобто, коли працівники мають можливість працювати в безпечному та комфортному середовищі, вони здатні бути більш ефективними та зосередженими на своїх обов'язках.

Заходи з охорони праці сприяють зменшенню відсутності працівників на робочому місці через травми або хвороби, що в свою чергу забезпечує більшу стабільність та ефективність роботи. Охорона праці сприяє захисту прав працівників, зокрема щодо безпеки, здоров'я, працевлаштування та рівного ставлення на робочому місці. Вона включає в себе нормативно-правові акти, які гарантують права працівників і встановлюють відповідальність роботодавців за забезпечення безпеки й охорони праці. Забезпечення безпеки та здоров'я працівників допомагає уникнути втрати кваліфікованої робочої сили, зменшує витрати на підготовку та навчання нових працівників, а безпечне та здорове робоче середовище сприяє покращенню задоволеності працівників.

Охорона праці також є важливою складовою корпоративної відповідальності. Підприємства, які вкладають зусилля у створення безпечних умов праці, демонструють свою готовність нести відповідальність за добробут своїх працівників та сприяють створенню позитивного іміджу. Також охорона

праці допомагає в розвитку соціального діалогу між роботодавцями та працівниками. Забезпечення безпеки й охорони праці вимагає співпраці всіх зацікавлених сторін, включаючи роботодавців, профспілки та урядові органи. Це сприяє створенню конструктивного діалогу, вирішенню проблем та покращенню умов праці.

Забезпечення безпеки й охорони праці має значний вплив на економіку та суспільство в цілому, оскільки сприяє зниженню соціального навантаження на суспільство. Заходи з охорони праці сприяють зниженню витрат на медичне обслуговування та страхування. Впровадження ефективних заходів з охорони праці може допомогти зменшити витрати на компенсацію травм та хвороб, пов'язаних з роботою, а також на відшкодування матеріальних збитків, пов'язаних з недбалістю або порушеннями у сфері безпеки. Це сприяє оптимізації виробничих процесів та підвищенню рентабельності.

Охорона праці є важливою складовою сталого розвитку. Заходи з безпеки й охорони праці спрямовані на забезпечення екологічної, соціальної та економічної сталості. Також охорона праці може стимулювати інновації та покращення робочих процесів. Коли компанії зосереджуються на створенні безпечного та здорового робочого середовища, вони сприяють розвитку нових технологій, методів роботи та інноваційних рішень, які покращують якість та продуктивність праці.

Загалом, соціальне значення охорони праці полягає в забезпеченні безпеки, здоров'я та благополуччя працівників, підвищенні якості життя та рівня задоволення від праці, сприянні сталому розвитку та підвищенню конкурентоспроможності підприємств та суспільства в цілому.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра розроблено КСВНІЕС на основі штучного інтелекту, яка здатна виявляти нетипові (екстрені) ситуації, зокрема:

- запропоновано ідентифікацію ЕС здійснювати шляхом визначення пози «руки вгору» в людини, яка знаходиться в приміщенні;
- розроблено архітектуру КСВНІЕС на основі ESP32-CAM та обґрунтовано її апаратне та програмне забезпечення;
- обґрунтовано здійснювати ідентифікацію екстреної ситуації за 6 ключовими точками з використанням моделі машинного навчання PoseNet;
- розроблено алгоритм роботи КСВНІЕС;
- здійснено налаштування КСВНІЕС для трансляції відеоданих;
- сформовано набір даних із 1510 зображень для навчання моделі ШІ та здійснено перевірку прийнятих рішень;
- використано для побудови моделі машинного навчання методи класифікації: метод логістичної регресії, метод k-найближчих сусідів та SVM з ядрами Гауса та оцінено їх точність.

Точність методу логістичної регресії при класифікації пози «руки вгору» становила 92%, методу k-найближчих сусідів – 92,7%. SVM з ядрами Гауса класифікатор показав точність визначення 100%.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Величко Д.В. Аналіз систем виявлення екстрених ситуацій. Природничі та гуманітарні науки. Актуальні питання: Матеріали VI Міжнародної студентської науково - технічної конференції (27-28 квітня 2023р.). Тернопіль: Тернопільський національний технічний університет ім. І.Пулюя, 2023. с. 116.

2. Відеоспостереження: типи систем і застосування. URL: https://vv24.in.ua/videosposterezhennya-typy-system.php?lang=1 (дата звернення: 03.03.2023).

3. Raspberry PI. URL: https://www.raspberrypi.com/ (дата звернення: 15.03.2023).

4. ESP32 Datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (дата звернення: 03.03.2023).

5. Модуль Wi-Fi ESP32-CAM з камерою 2MP. URL: https://arduino.ua/prod3458-modyl-wi-fi-esp32-s-kameroi-2mp (дата звернення: 03.03.2023).

6. Arm. URL: https://www.arm.com/products/silicon-ip-cpu?families=cortex- m&showall=true- (дата звернення: 15.03.2023).

7. STM32 32-bit Arm Cortex MCUs. URL: https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html (дата звернення: 15.03.2023).

8. Toshpulatov, M., Lee, W., Lee, S. et al. Human pose, hand and mesh estimation using deep learning: a survey.(2022*). The Journal of Supercomputing*, volume 78, 2022. p. 616–654. URL: https://doi.org/10.1007/s11227-021-04184-7.

9. Human Pose Estimation with Deep Learning – Ultimate Overview in 2023. URL: https://viso.ai/deep-learning/pose-estimation-ultimate-overview/ (дата звернення: 29.03.2023).

10. Posenet Model in ML. URL: https://iq.opengenus.org/posenet-model/ (дата звернення: 28.04.2023).

11. Arduino IDE 1 Installation (Windows). URL: https://docs.arduino.cc/software/ide-v1/tutorials/Windows (дата звернення: 15.04.2023).

12. How to Program / Upload Code to ESP32-CAM AI-Thinker (Arduino IDE). URL: https://randomnerdtutorials.com/program-upload-code-esp32-cam/ (дата звернення: 15.04.2023).

13. Trevor Hastie, Robert Tibshirani and Jerome H. Friedman. The Elements of Statistical Learning (2nd edition). Springer-Verlag, 2008. 763 p.

14 Fawcett, Tom. An Introduction to ROC Analysis. Pattern Recognition Letters, 2006. 27 (8): 861–874. doi:10.1016/j.patrec.2005.10.010.

15. Classification: ROC Curve and AUC. URL: https://developers.google.com/machine-learning/crash-course/classification/check-your-understanding-accuracy-precision-recall (дата звернення: 25.05.2023).

16. Осухівська Г. М., Тиш Є. В., Луцик Н. С., Паламар А. М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

17. Скобло Ю. С., Соколовська Т. Б., Морозенко Д. І. та ін. Безпека життєдіяльності. Навчальний посібник для вищих навчальних закладів 3-4 рівнів акредитації. – К.: Кондор, 2003. 424 с.

18. Стиценко Т. Є., Пронюк Г. В., Сердюк Н. М., Хондак І. І. Безпека життєдіяльності: навч. посібник / Т. Є Стиценко, Г. В. Пронюк, Н. М. Сердюк, І. І. Хондак. – Харків: ХНУРЕ, 2018. – 336 с.

19. Пожарова О. В. Охорона праці: навчальний посібник / О. В. Пожарова. - Одеса, 2022. - 86 с. Режим доступу: https://doi.org/10.32837/11300.18442

ДОДАТОК А.

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

"Затверджую"
Завідувач кафедри КС
_____ Осухівська Г.М.
"____"_____2023 р.

КОМП'ЮТЕРИЗОВАНА СИСТЕМА ВІДЕОНАГЛЯДУ З ФУНКЦІЄЮ ІДЕНТИФІКАЦІЇ ЕКСТРЕНИХ СИТУАЦІЙ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ

ТЕХНІЧНЕ ЗАВДАННЯ
на 9 листках

Вид робіт: Кваліфікаційна робота

На здобуття освітнього ступеня *бакалавр*
Спеціальність *123 «Комп'ютерна інженерія»*

"УЗГОДЖЕНО"
Керівник кваліфікаційної роботи
_____ PhD, доц. Луцик Н.С.
"____"_____2023 р.

"ВИКОНАВЕЦЬ"
Студентка групи СІ-41
_____ Величко Д.В.
"____"_____2023 р.

Тернопіль 2023

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп’ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.035.00.00

1.2 Виконавець

Студентка групи СІ-41, факультету комп’ютерно-інформаційних систем і програмної інженерії, кафедри комп’ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Величко Діана Вадимівна.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№ 4/7-238 від 28.02.2023 р.).

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 28.02.2023 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2023 р.

1.5 Порядок оформлення та пред’явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ДСТУ, ЄСКД та ЄСПД.

Пред’явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90%, наявності пояснювальної записки та графічного матеріалу.

Пред’явлення результатів кваліфікаційної роботи відбувається шляхом захисту на засіданні ЕК, представлення основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Комп’ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту призначена для відеомоніторингу в приміщеннях організацій, підприємств та установ різного призначення, наприклад таких як: банки, школи, торгівельні та розважальні заклади, приміщення сфери обслуговування та ін., а також як складова систем «розумного будинку».

Така система повинна приймати рішення про небезпеку (екстрену, нетипову ситуацію) за допомогою використання штучного інтелекту та інформувати про неї.

Організація такої системи повинна бути орієнтована на одержання і трансляцію відео в реальному часі та забезпечувати можливість інформування у випадку виникнення екстреної ситуації.

2.2 Мета створення системи

Метою створення комп’ютеризованої системи відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту є автоматизація процесу виявлення небезпечних ситуацій та інформування про них відповідних служб. Для досягнення поставленої мети роботи необхідно розв’язати такі задачі:

- запропонувати ознаки ідентифікації екстреної ситуації системою;
- проаналізувати системи відеонагляду з метою їх удосконалення відповідно до поставлених вимог;
- обґрунтувати оптимальну для інтеграції у систему відеонагляду модель ідентифікації екстрених ситуацій відповідно до запропонованих ознак;
- розробити архітектуру комп’ютеризованої системи відеонагляду з функцією ідентифікації екстрених ситуацій та обґрунтувати її забезпечення;
- розробити алгоритм роботи комп’ютеризованої системи відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту;
- здійснити налаштування комп’ютеризованої системи відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту для трансляції відеоданих;
- сформувати набір даних (датасет) для навчання моделі штучного інтелекту та здійснити перевірку прийнятих рішень.

2.3 Характеристика об'єкту

2.3.1 Основні задачі та функції об’єкту

При автоматизації процесу відеонагляду за об’єктом з метою виявлення екстрених ситуацій необхідно передбачити застосування відповідного апаратного забезпечення та середовища передавання даних, зокрема:

– відеокамери з необхідною роздільною здатністю;

– апаратного пристрою управління процесом захоплення і передавання відеопотоку з камери;

– сервера зберігання бази даних.

Програмне забезпечення комп’ютеризованої системи повинно включати:

- програмне забезпечення налаштування та підтримки функціонування відеокамери та мікроконтролера;
- програмне забезпечення функції розпізнавання екстрених ситуацій;
- програмне забезпечення інформування про екстрені ситуації.

Проектована комп’ютеризована система повинна забезпечити підвищення надійності та безпеки при знаходженні людей в приміщеннях різного призначення.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

Загалом, комп’ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту повинна задовольняти таким вимогам:

- безперебійна та стабільна робота;
- трансляція відеопотоку у режимі реального часу
- невеликі розміри камери відеонагляду;
- енергоефективність,
- низька вартість.

3.1.1 Вимоги до структури та функціонування системи

В загальному випадку, модель проектованої комп’ютеризованої системи відеонагляду повинна забезпечувати реалізацію основних функцій щодо

трансляції відео, виконувати виявлення екстрених ситуацій та сповіщати про них, а також зберігати відео на відповідному сервері.

До функцій комп'ютеризованої системи відеонагляду належать:

– здатність одержувати відео з камери в реальному часі;

– можливість передавання відеопотоку через веб-сервер;

– можливість виявляти людей;

– здатність ідентифікувати екстрену (нетипову) ситуацію;

– можливість інформування про екстрену ситуацію;

– здатність до гнучкості та масштабування апаратного забезпечення системи.

3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Взаємодію камери та сервера варто організувати на основі технології Wi Fi з частотою передачі даних на рівні 2,4 ГГц.

3.1.3 Вимоги по діагностуванню системи

Діагностування комп'ютеризованої системи відеонагляду проводиться у відповідності до регламенту, визначеного організацією, що впроваджуватиме дану систему.

3.1.4 Перспективи розвитку, модернізація системи

До перспектив розвитку проектованої системи належить здатність до масштабування, тобто збільшення кількості модулів та створення системи управління відеопотоками з багатьох пристроїв.

3.1.5 Вимоги до надійності системи

Надійність функціонування комп’ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій на основі штучного інтелекту гарантується авторизованим доступом до вебсервера трансляції відеопотоку у реальному часі, а також на рівні авторизованого доступу до сервера зображень і збереженого відео. Крім того, на фізичному рівні повинен бути обмежений доступ до апаратного забезпечення комп’ютеризованої системи і лише особи з відповідними правами доступу можуть вносити зміну або здійснювати обслуговування системи.

3.1.6 Вимоги до функцій та задач, які виконує система

Функціональні вимоги та задачі, які повинна реалізовувати комп’ютеризована система полягають в наступному:

– захоплення відеопотоку з відеокамери;

– з’єднання мікроконтролера з веб-сервером трансляції відео;

– здатність забезпечувати виявлення людей в приміщенні;

– можливість ідентифікувати екстрену ситуацію;

–-можливість інформування про екстерну ситуацію;

– здатність до гнучкості та масштабування апаратного забезпечення системи.

3.1.7 Вимоги до апаратного забезпечення

Основними вимогами до апаратних засобів системи відеонагляду є:

– відеокамера,

– контролер з WI-FI для передавання інформації,

– система віддаленого спостереження, яка, в свою чергу, має містити сервер та комп’ютер (планшетний комп’ютер або смартфон).

Вимоги до робочих станцій комп’ютеризованої системи:

– процесор з тактовою частотою > 2,0 ГГц;

– оперативна пам’ять об’ємом 4 Гб;

– розмір HDD не менше 120 Гб.

Вимоги до сервера:

– процесор з тактовою частотою 2,2 ГГц/ядро з кількістю ядер > 10;

– оперативна пам’ять розміром від 32 ГБ;

– об’єм HDD – 120 Тб.

3.1.8 Вимоги до програмного забезпечення

Системне програмне забезпечення мікроконтролера.

Програмне забезпечення робочих станцій – Linux, Windows 10.

Програмне забезпечення сервера – Linux подібні, Windows Server.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

– пояснювальної записки;

– графічного матеріалу:

1 Архітектура комп’ютеризованої системи.

2 Схема електрична принципова.

3 Блок-схема алгоритму роботи системи.

4 Результати роботи.

*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ п/п	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1.	Розробка технічного завдання	06.03-12.03
2.	Робота над розділом «Аналіз технічного задання»	13.03-20.03
3.	Розробка архітектури системи	20.03-28.03
4	Вибір апаратного забезпечення	23.03-29.03
5	Вибір програмного забезпечення	29.03-04.04
6.	Розробка алгоритму роботи системи	04.04-19.04
7.	Розробка прототипу та налаштування системи	19.04-30.04
8.	Формування набору даних для навчання моделі штучного інтелекту	01.05-18.05
9.	Реалізація функції ідентифікації екстрених ситуацій	19.05-25.05
10.	Перевірка прийнятих рішень	26.05-02.06
11.	Виконання завдань до розділу «Безпека життєдіяльності, основи охорони праці»	03.06-08.06
12.	Оформлення пояснювальної записки та графічної частини	09.06-16.06
13.	Нормоконтроль	16.06
14.	Захист кваліфікаційної роботи	23.06

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

ДОДАТОК Б.

Тези конференції

Міністерство освіти і науки України,
Тернопільський національний технічний університет
імені Івана Пулюя
Маріборський університет (Словенія)
Технічний університет в Кошице (Словаччина)
Каунаський технологічний університет (Литва)
Львівський національний університет
імені Івана Франка,
Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
Луцький національний технічний університет,
Чернівецький національний університет
імені Юрія Федьковича,
Вроцлавський економічний університет (Польща)
Університет технологій та економіки
імені Хелени Ходковської (Польща)
Донбаська державна машинобудівна академія

Студентське наукове товариство

VI МІЖНАРОДНА

студентська науково - технічна конференція

"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ.

АКТУАЛЬНІ ПИТАННЯ"

27-28 квітня 2023 р.

(збірник тез конференції)

Тернопіль 2023

УДК 654.9
Величко Д. - ст. гр. СІ-41
Тернопільський національний технічний університет імені Івана Пулюя

АНАЛІЗ СИСТЕМ ВИЯВЛЕННЯ ЕКСТРЕНИХ СИТУАЦІЙ

Науковий керівник: Ph.D., доцент Луцик Н.С.

Velychko D.
Ternopil Ivan Puluj National Technical University

ANALYSIS OF DANGEROUS SITUATION DETECTION SYSTEMS

Supervisor: Ph.D., Assoc. Prof. Lutsyk N.S.

Ключові слова: екстрена ситуація, система відеонагляду, штучний інтелект
Keywords: dangerous situation, video surveillance system, artificial intelligence

На сьогоднішній день вкрай важливим завданням є безпека людей. На жаль, бувають обставини, за яких необхідно терміново повідомити про екстрену ситуацію, яка загрожує здоров'ю, життю чи майну людей. Для виявлення та інформування про такі випадки існують різні технічні засоби, наприклад, «тривожна кнопка», системи охоронної сигналізації, системи відеонагляду і т.ін. Але такі системи не завжди можуть забезпечити своєчасне виявлення та інформування про небезпечну ситуацію.

«Тривожна кнопка» - прихована від сторонніх очей клавіша, призначена для миттєвого сповіщення про небезпеку. Цей метод оповіщення ефективно використовується, але недоліком її є те, що для повідомлення про екстрену ситуацію потрібно непомітно натиснути на неї, що не завжди є можливим.

Охоронна сигналізація - це система, яка розрахована на попередження несанкціонованого доступу до приміщення і використовується для захисту об'єкта від вторгнення, викрадання майна або інших небезпечних подій. Сигналізація спрацьовує, за умови, коли давачі реагують на певний сигнал (рух, звук, температуру, концентрацію газів та ін.), але, коли загрожує небезпека безпосередньо людям в приміщенні від інших осіб, то така система відреагувати не здатна. В таких випадках добре себе зарекомендували системи відеонагляду.

Система відеонагляду - це система, яка використовує відеокамери для відстеження подій, що відбуваються. Сучасні системи відеонагляду, використовуючи технології штучного інтелекту, мають можливість попередити, а в подальшому і допомогти виявити зловмисників. Відеокамери можуть розпізнати різні об'єкти, а також здійснювати ідентифікацію особи, порівнюючи її з своєю базою. Такі системи здійснюють передавання відеозображення в реальному часі на пульт спостереження. Але про екстрену ситуацію, викликану погрозою чи нападом безпосередньо на людину, такі системи не інформують миттєво, натомість все залежить від реагування працівника охорони, який слідкує за моніторами.

Тому актуальним завданням є вдосконалення системи відеонагляду функцією ідентифікації екстрених ситуацій з миттєвим оповіщенням про небезпеку. Ідентифікацію небезпечних ситуацій можна реалізувати, використовуючи штучний інтелект, а саме, розпізнаючи позу людини, яка вказує на небезпеку. Такою позою, яку система сприйматиме як екстрену ситуацію, запропоновано вважати позу «руки вгору».

Оболонін О. **МОДЕЛЮВАННЯ РЕЖИМІВ РОБОТИ АСИНХРОННОГО ДВИГУНА ПРИ НЕСИМЕТРІЇ НАПРУГИ МЕРЕЖІ**	**99**
Терявський П., Невідомський М. **ЗАСТОСУВАННЯ ПОЛІМЕРНИХ ІЗОЛЯТОРІВ НА ВИСОКОВОЛЬТНИХ ЛІНІЯХ ЕЛЕКТРОПЕРЕДАЧІ**	**101**
Хариш П. **АНАЛІЗ ТА УПРАВЛІННЯ ЯКІСТЮ ЕЛЕКТРИЧНОЇ ЕНЕРГІЇ**	**103**
Якимчук С. **ДОСЛІДЖЕННЯ ПЕРЕХІДНИХ ПРОЦЕСІВ В ЕЛЕКТРИЧНОМУ КОЛІ З СВІТЛОДІОДОМ**	**105**
Хомишин М **СУЧАСНІ ТЕХНОЛОГІЇ SDR РАДІОЗВ'ЯЗКУ**	**106**
Анистюк Д., Маєвський Т. **ЛОКАТОРИ ЕЛЕМЕНТІВ ПРИ ТЕСТУВАННІ ВЕБ-ІНТЕРФЕЙСІВ**	**108**
Анистюк Д., Маєвський Т. **ТЕСТУВАННЯ ГРАФІЧНИХ ІНТЕРФЕЙСІВ ВЕБ-ЗАСТОСУНКІВ**	**109**
Болож О. **ОГЛЯД SERVERLESS АРХІТЕКТУРИ ТА ЇЇ ПЕРЕВАГИ**	**110**
Букатка С., Тимощук В. **ХЕШ-АЛГОРИТМ ШИФРУВАННЯ ПАРОЛІВ КОРИСТУВАЧІВ ОС LINUX**	**112**
Буковська А. **МЕТОДОЛОГІЯ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ СЕРЕДОВИЩА PLEXSYS ТА ЇЇ ВПЛИВ НА ПРИЙНЯТТЯ КОРИСТУВАЧЕМ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**	**114**
Величко Д. **АНАЛІЗ СИСТЕМ ВИЯВЛЕННЯ ЕКСТРЕНИХ СИТУАЦІЙ**	**116**
Вербіцький Р., Жураковський С. **ЯК ШТУЧНИЙ ІНТЕЛЕКТ ДОПОМАГАЄ МАРКЕТОЛОГАМ У РОБОТІ ІЗ ВІЗУАЛОМ**	**117**
Вербіцький І. **РОЗРОБКА SPA-ЗАСТОСУНКІВ НА MERN-СТЕК**	**118**
Вивюрка А. **ОГЛЯД ШІ CHATSONIC**	**120**
Воробець І. **ВИКОРИСТАННЯ МОДЕЛЕЙ ARIMA ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ІЗ ВЛАСТИВІСТЮ ЦИКЛІЧНОСТІ**	**122**
Гайдук В. **НЕБЕЗПЕКА ШТУЧНОГО ІНТЕЛЕКТУ ТА АЛГОРИТМІВ**	**124**

ДОДАТОК В.

Перелік елементів

Поз. позначення	*Найменування*	*Кіл.*	*Примітка*
	Модулі		
J1	ESP32	1	
U1	OV7670	1	
PWR1	PL2303HX-USB-TTL-ADAPTER	1	
	Резистори		
R1, R2	AXIAL-0.3 4,7 кОм	2	
R3, R4	AXIAL-0.3 10 кОм	2	
R5, R6	AXIAL-0.3 270 Ом	2	
	Конденсатори		
C1, C2, C3	RAD-0.2 10 мкФ	3	
	Роз’єми		
P1	HDR-1X2/2.54	1	
P2, P5, P6	HDR-2X1/2.54	3	
P3	HDR-2X1/2.54 BTN1	1	
P4	HDR-2X1/2.54 BTN2	1	

					КС КРБ 123.035.00.00 ПЕ	
Змн.	*Арк.*	*№ докум.*	*Підпис*	*Дата*		
Розробив		*Величко Д.В.*			*Комп’ютеризована система відеонагляду з функцією ідентифікації екстрених ситуацій* *Перелік елементів*	*Літ.* / *Арк.* / *Акрушів*
Перевірив		*Луцик Н.С.*				
Рецензент		*Приймак М.В.*				*ТНТУ, каф. КС, гр. СІс-41*
Н. Контр.		*Тиш Є.В.*				
Зав. каф.		*Осухівська Г.М.*				

ДОДАТОК Д.

Лістинг коду програми

TF.ino

```
#include <Arduino.h>
#include <WiFi.h>
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "Arduino.h"
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"
#include "page.h"
#define PART_BOUNDARY "123456789000000000000987654321"


static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length:
%u\r\n\r\n";
httpd_handle_t camera_httpd = NULL;
httpd_handle_t stream_httpd = NULL;


const char* ssid = "OPPO Reno4 Pro 5G";
const char* password = "d023f1f81e67";


static esp_err_t capture_handler(httpd_req_t *req){
    Serial.println("Capture image");
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    fb = esp_camera_fb_get();
    if (!fb) {
```

```
        Serial.println("Camera capture failed");
        httpd_resp_send_500(req);
        return ESP_FAIL;
    }
    httpd_resp_set_type(req, "image/jpeg");
    httpd_resp_set_hdr(req, "Content-Disposition", "inline;
filename=capture.jpg");
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");

    res = httpd_resp_send(req, (const char *)fb->buf, fb->len);
    esp_camera_fb_return(fb);
    return res;
}
static esp_err_t page_handler(httpd_req_t *req) {
  Serial.println("Main PAGE START");
    httpd_resp_set_type(req, "text/html");
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
    //httpd_resp_send(req, page, sizeof(page));
    httpd_resp_send(req, page, HTTPD_RESP_USE_STRLEN);
  Serial.println("Main PAGE StoP");
}
static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
        return res;
    }
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
    while(true){
```

```
        fb = esp_camera_fb_get();
        if (!fb) {
            Serial.println("Camera capture failed");
            res = ESP_FAIL;
        } else {

                if(fb->format != PIXFORMAT_JPEG){
                    bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf,
&_jpg_buf_len);
                    esp_camera_fb_return(fb);
                    fb = NULL;
                    if(!jpeg_converted){
                        Serial.println("JPEG compression failed");
                        res = ESP_FAIL;
                    }
                } else {
                    _jpg_buf_len = fb->len;
                    _jpg_buf = fb->buf;
                }
             }
        if(res == ESP_OK){
            res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
        }
        if(res == ESP_OK){
            size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART,
_jpg_buf_len);
            res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
        }
        if(res == ESP_OK){
            res = httpd_resp_send_chunk(req, (const char *)_jpg_buf,
_jpg_buf_len);
        }
        if(fb){
            esp_camera_fb_return(fb);
```

```
            fb = NULL;
            _jpg_buf = NULL;
        } else if(_jpg_buf){
            free(_jpg_buf);
            _jpg_buf = NULL;
        }
        if(res != ESP_OK){
            break;
        }
    }
    return res;
}
void startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    httpd_uri_t index_uri = {
        .uri       = "/",
        .method    = HTTP_GET,
        .handler   = stream_handler,
        .user_ctx  = NULL
    };
    httpd_uri_t page_uri = {
        .uri       = "/ts",
        .method    = HTTP_GET,
        .handler   = page_handler,
        .user_ctx  = NULL
    };
    httpd_uri_t capture_uri = {
        .uri       = "/capture",
        .method    = HTTP_GET,
        .handler   = capture_handler,
        .user_ctx  = NULL
    };

    Serial.printf("Starting web server on port: '%d'\n", config.server_port);
```

```
    if (httpd_start(&camera_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(camera_httpd, &capture_uri);
        httpd_register_uri_handler(camera_httpd, &page_uri);
    }
    // start stream using another webserver
    config.server_port += 1;
    config.ctrl_port += 1;
    Serial.printf("Starting stream server on port: '%d'\n", config.server_port);
    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(stream_httpd, &index_uri);
    }
}
void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
```

```
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                      for larger pre-allocated frame buffer.
  if(psramFound()){
    config.frame_size = FRAMESIZE_QVGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_QVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }
#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif
  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV2640_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);
```

```
#if defined(CAMERA_MODEL_M5STACK_WIDE)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  startCameraServer();
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}
void loop() {
  // put your main code here, to run repeatedly:
  delay(10);
}
```

camera_pins.h

```
#if defined(CAMERA_MODEL_WROVER_KIT)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    21
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27

#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      19
#define Y4_GPIO_NUM      18
#define Y3_GPIO_NUM       5
#define Y2_GPIO_NUM       4
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

#elif defined(CAMERA_MODEL_ESP_EYE)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    4
#define SIOD_GPIO_NUM    18
#define SIOC_GPIO_NUM    23

#define Y9_GPIO_NUM      36
#define Y8_GPIO_NUM      37
#define Y7_GPIO_NUM      38
#define Y6_GPIO_NUM      39
#define Y5_GPIO_NUM      35
```

```
#define Y4_GPIO_NUM      14
#define Y3_GPIO_NUM      13
#define Y2_GPIO_NUM      34
#define VSYNC_GPIO_NUM   5
#define HREF_GPIO_NUM    27
#define PCLK_GPIO_NUM    25

#elif defined(CAMERA_MODEL_M5STACK_PSRAM)
#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     25
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       32
#define VSYNC_GPIO_NUM    22
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_M5STACK_WIDE)
#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     22
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
```

```
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       32
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

#elif defined(CAMERA_MODEL_TTGO_T_JOURNAL)
#define PWDN_GPIO_NUM      0
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
```

```
#define SIOD_GPIO_NUM     25
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       17
#define VSYNC_GPIO_NUM    22
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21

#else
#error "Camera model not selected"
#endif
```

page.h

```
static const char page[] = R"rawliteral(
<!DOCTYPE html>
<html lang="en">

 <head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>ESP32-CAM TensorflowJS</title>
   <style>
   body {
   font-family: "PT Sans", sans-serif;
   background-color: #dde2e1;
   margin: 0;
```

```
    color: #636060;
    line-height: 1.6;
    }
    a {
    text-decoration: none;
    color: #ccc;
    }
    h2 {
    display: block;
    font-size: 1.17em;
    margin-block-start: 1em;
    margin-block-end: 1em;
    margin-inline-start: 0px;
    margin-inline-end: 0px;
    font-weight: bold;
    }
    .container {
    max-width: 1180px;
    text-align: center;
    margin: 0 auto;
    padding: 0 3rem;
    }
    .btn {
    padding: 1rem;
    color: #fff;
    display: inline-block;
    background: red;
    margin-bottom: 1rem;
    }

    </style>
    <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"> </script>
     <script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/mobilenet">
</script>
```

```
    <script language="javascript">
        function classifyImg() {
           const img = document.getElementById("img1");
           const r = document.getElementById("results");
           r.innerHTML = "";

           console.log("Classify...");
           img.crossorigin = " ";
           img.onload = function() {
                console.log("Wait to load..");
                mobilenet.load().then(model => {
                // Classify the image.
                model.classify(img).then(predictions => {
                  for (i in predictions) {
                    r.innerHTML = r.innerHTML + "<b>" + predictions[i].className +
"</b> - " + predictions[i].probability + "<br/>";
                    img.onload = null;
                    img.src = "http://192.168.1.4:81";
                  }
                });
               });
           }
           img.src = "http://192.168.1.4/capture?t=" + Math.random();
        }
    </script>
   </head>

   <body>
     <div class="container">
     <h2>TensorflowJS with ESP32-CAM</h2>
     <section>
       <img id="img1" width="320" height="200" src="http://192.168.1.4:81"
crossorigin style="border:1px solid red"/>
       <div id="results"/>
```

```
      </section>
      <section>
        <a href="#" class="btn" onclick="classifyImg()">Classify the image</a>
       </section>
       <section id="i" />
   </div>
   </body>
</html>
)rawliteral";
```