

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
Факультет електроніки та інформаційних технологій

Кафедра електроніки,  
загальної та прикладної фізики

Кваліфікаційна робота магістра

**СИСТЕМИ АНАЛІЗУ ДАНИХ З ЕЛЕМЕНТАМИ ШТУЧНОГО  
ІНТЕЛЕКТУ НА БАЗІ МІКРОПРОЦЕСОРНИХ ПЛАТФОРМ**

спеціальності 171 Електроніка

Здобувач вищої освіти гр. ЕП.м-01н

В. В. Сімагін

Науковий керівник,  
канд. фіз.-мат. наук, ст.викл.

К. В. Тищенко

Завідувач кафедри ЕЗПФ  
д-р фіз.-мат. наук, професор

І. Ю. Проценко

Суми 2022

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра електроніки, загальної та прикладної фізики  
Спеціальність 171 – Електроніка, освітньо-наукова програма  
«Електронні інформаційні системи»

ЗАТВЕРДЖУЮ  
Зав. кафедри ЕЗПФ  
І.Ю. Проценко  
«02» травня 2022 року

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА**

**Сімагіна Вадима Володимировича**

Тема роботи Системи аналізу даних з елементами штучного інтелекту на базі мікропроцесорних платформ

затверджена наказом по університету від «18» квітня 2022 р., № 0270-VI

2. Термін здачі студентом закінченої роботи 17 травня 2022 року

3. Вихідні дані до роботи (актуальність, мета)

Сьогодні розгортання системи аналізу даних може бути виконано на базі одноплатних комп'ютерів, котрі мають ряд унікальних характеристик, необхідних в цій області: низька вартість, хороша автономність роботи, достатня розрахункова потужність, універсальність. Пристрої такого класу мають операційну систему загального користування, що дозволяє використовувати широкий спектр сучасних програмних засобів та забезпечує загальну захищеність пристрою від несанкціонованого доступу. Широко затребуваними сьогодні є системи аналізу потокового відео, наприклад з камер спостереження. Такі системи виконують велику кількість паралельних розрахунків. Саме тому при аналізі таких даних доцільним є застосування програмних засобів, які мають ознаки штучного інтелекту і здатні до самонавчання.

Метою кваліфікаційної роботи магістра є огляд сучасних систем і технологій для проектування і розробки систем аналізу даних з елементами штучного інтелекту з використанням мікропроцесорних платформ та розробка системи аналізу потокового відео в режимі реального часу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що належить їх розробити)

1. Використання елементів штучного інтелекту в системах аналізу даних.

2. Огляд мікропроцесорних платформ для систем з штучним інтелектом.

3. Підбір компонентів та розробка схеми пристрою для аналізу даних з елементами ШІ.

4. Розробка програмного забезпечення з елементами ШІ для аналізу потокового аудіо та відео.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайди № 1, 2. Актуальність і мета роботи.

Слайди № 3-4. Поняття штучного інтелекту, підходи і напрямки дослідження.

Класифікація та принципи роботи ШІ.

Слайди № 5-7. Мікропроцесорні платформи для систем зі штучним інтелектом.

Слайди № 8, 9. Операційні системи для Raspberry Pi.

Слайди № 10, 11. Розробка схеми пристрою для аналізу даних.

Слайди № 12-14. Розробка та налаштування програмного забезпечення.

Слайди № 15-16. Висновки, публікації.

6. Дата видачі завдання 02.05.2022 р.

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз літературних даних	до 02.05.2022 р.	<i>вик.</i>
2.	Проведення експерименту, моделювання, розрахунків, обробка результатів	до 12.05.2022 р.	<i>вик.</i>
4.	Оформлення тексту кваліфікаційної роботи.	до 17.05.2022 р.	<i>вик.</i>
5.	Попередній захист роботи	18.05.2022 р., онлайн	<i>вик.</i>
6.	Захист роботи в екзаменаційній комісії	24.05.2022 р., онлайн	<i>вик.</i>

Здобувач вищої освіти

Сімагін В.В.

Науковий керівник

Тищенко К.В.

## РЕФЕРАТ

Об'єктом дослідження кваліфікаційної роботи магістра є сучасні мікропроцесорні платформи і технології для проектування та розробки систем аналізу даних з елементами штучного інтелекту для аналізу потокового аудіо і відео в режимі реального часу.

Мета роботи полягає в розробці пристрою на базі одноплатного комп'ютера Raspberry Pi для збору аудіо та відео даних, а також розробка програмного забезпечення для їх обробки і аналізу із застосуванням елементів штучного інтелекту.

У роботі розглянуто історію створення, проблеми, підходи та напрямки розвитку штучного інтелекту. Розглянуто апаратні платформи Raspberry Pi та NVIDIA Jetson Nano, які можуть бути використані при розробці електронних систем інтелектуального збору і обробки даних.

Розроблено принципову схему електронного пристрою та програмне забезпечення, які дозволяють реалізувати дві функціональні можливості з самонавчанням, які демонструють елементи штучного інтелекту, а саме розпізнавання голосу та візуальних елементів у потоковому відео, а також керує електричними серводвигунами за допомогою голосових команд.

Робота викладена на 49 сторінках, зокрема містить 25 рисунків, 1 таблицю, список цитованої літератури із 21 джерела.

**КЛЮЧОВІ СЛОВА:** ШТУЧНИЙ ІНТЕЛЕКТ, РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, RASPBERRY PI, IBM, NODE JS, NVIDIA JETSON NANO, СИСТЕМА АНАЛІЗУ ДАНИХ, КАСКАДИ НААР

## ЗМІСТ

ВСТУП .....	6
РОЗДІЛ 1 МІКРОПРОЦЕСОРНІ СИСТЕМИ АНАЛІЗУ ДАНИХ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ .....	7
1.1 Використання елементів штучного інтелекту в системах аналізу даних	7
1.1.1 Поняття штучного інтелекту .....	7
1.1.2 Історія розвитку штучного інтелекту .....	8
1.1.3 Проблеми штучного інтелекту .....	9
1.1.4 Підходи та напрямки у дослідженнях ШІ .....	11
1.1.5 Принцип роботи штучного інтелекту .....	14
1.2 Мікропроцесорні платформи для систем з ШІ .....	16
1.2.1 Огляд платформи Raspberry Pi .....	17
1.2.2 Огляд платформи NVIDIA Jetson Nano .....	21
РОЗДІЛ 2 МЕТОДИКА ТА ТЕХНІКА ЕКСПЕРИМЕНТУ .....	25
2.1 Апаратне забезпечення для розробки системи аналізу даних.....	25
2.2 Підготовка пристрою до роботи.....	29
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРИСТРОЮ ОБРОБКИ ДАНИХ НА БАЗІ RASPBERRY PI.....	33
3.1 Налаштування програмного забезпечення для розпізнання голосових команд .....	33
3.2 Налаштування програмного забезпечення для розпізнання елементів зображення.....	39
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48

## ВСТУП

Розробка портативних систем аналізу даних сьогодні, переважно, виконується із застосуванням електронних обчислювальних платформ на базі мікропроцесорів. Для реалізації інтелектуальної обробки даних доцільним є застосування елементів штучного інтелекту (ШІ) у програмній частині обчислювального комплексу. Виконання програми з ШІ вимагає значних розрахункових потужностей, і як наслідок, розвинутої апаратної складової кінцевого пристрою.

Сьогодні розгортання системи аналізу даних може бути виконано на базі одноплатних комп'ютерів (напр. Raspberry Pi, Jetson Nano), котрі мають ряд унікальних характеристик, необхідних в цій області: відносно низька вартість, хороша автономність роботи, достатня розрахункова потужність, універсальність. Пристрої такого класу мають операційну систему загального користування (здебільшого Linux), що дозволяє використовувати широкий спектр сучасних програмних засобів та забезпечує загальну захищеність пристрою від несанкціонованого доступу.

Широко затребуваними сьогодні є системи аналізу потокового відео, наприклад з камер спостереження. Такі системи виконують велику кількість паралельних розрахунків, і мікропроцесор досить погано підходить для даних цілей. Більш продуктивної роботи при аналізі відеопотоку можна досягти виконуючи розрахунки за допомогою графічних процесорів.

Мета роботи полягає в розробці пристрою на базі одноплатного комп'ютера Raspberry Pi для збору аудіо та відео даних, а також розробка програмного забезпечення для їх обробки і аналізу із застосуванням елементів штучного інтелекту.

Результати кваліфікаційної роботи були представлені і обговорені на Міжнародній науково-методичній конференції «Фізика, електроніка, електротехніка ФЕЕ-2022» [21].

# РОЗДІЛ 1

## МІКРОПРОЦЕСОРНІ СИСТЕМИ АНАЛІЗУ ДАНИХ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ

### 1.1 Використання елементів штучного інтелекту в системах аналізу даних

**Штучний інтелект** (ШІ, англ. Artificial intelligence) – наука та технологія створення інтелектуальних систем та комп'ютерних програм [3]. ШІ пов'язані з подібним завданням використання комп'ютерів розуміння людського інтелекту, але обов'язково обмежується біологічно правдоподібними методами.

#### 1.1.1 Поняття штучного інтелекту

На 2021 рік дослідники використали таку класифікацію типів ШІ [2]:

– Artificial Super Intelligence (ASI) – гіпотетичний ШІ, який зможе не тільки відтворити максимум здібностей людини, але навіть перевершити його. Прихильники ідеї створення ASI вважають, що він набуде сили проникнення в думки і почуття людини з тим, щоб підпорядкувати їх своїй волі.

– Сильний, що залишається теж гіпотетичним, або загальний ШІ (Artificial General Intelligence, AGI) за ступенем розумності стоїть на щабель нижче ASI, адепти цього типу AI обмежені у своїх переконаннях можливістю створення машин, здатних як мінімум виконувати ті ж дії, що і людина.

– Слабкий, або вузький ШІ (Artificial Narrow Intelligence, ANI) дозволяє побачити в поведінці машин слабкі натяки на розум (тому його називають слабким). Він призначений для виконання лише певного вузького кола задач. У разі ANI неможливо жодна невіддільна людині автономна поведінка чи

самостійний розвиток. Системи, забезпечені ANI, можуть існувати тільки в тій формі, в якій вони були створені людиною і навіть теоретично не можуть вийти з-під контролю.

На початку 1980-х років. вчені в галузі теорії обчислень Барр та Файгенбаум запропонували таке визначення штучного інтелекту (ШІ) [3]: *«Штучний інтелект – це область інформатики, яка займається розробкою інтелектуальних комп'ютерних систем, тобто систем, які мають можливості, які ми традиційно пов'язуємо з людським розумом, – розуміння мови, навчання, здатність розмірковувати, вирішувати проблеми тощо.»*

Пізніше до ШІ стали відносити ряд алгоритмів і програмних систем, характерною властивістю яких є те, що вони можуть вирішувати деякі завдання так, як це робила б людина, що розмірковує над їх вирішенням.

Основні властивості ШІ - це розуміння мови, навчання і здатність мислити і, що важливо, діяти.

### **1.1.2 Історія розвитку штучного інтелекту**

Термін «штучний інтелект» з'явився у 1956 році, але справжньої популярності технологія ШІ досягла лише сьогодні на тлі збільшення обсягів даних, удосконалення алгоритмів, оптимізації обчислювальних потужностей та засобів зберігання даних.

Перші дослідження в галузі ШІ, що розпочались у 50-х роках минулого століття і були спрямовані на вирішення проблем та розробку систем символічних обчислень. У 60-х роках цей напрямок привернув інтерес Міністерства оборони США: американські військові почали навчати комп'ютери імітувати розумову діяльність людини. Наприклад, Управління перспективних дослідницьких проектів Міністерства оборони США (DARPA) виконало в 70-х роках низку проектів створення віртуальних вуличних карт [7]. І фахівцям DARPA вдалося створити інтелектуальних особистих помічників у 2003 році задовго до того, як з'явилися Siri, Alexa та Cortana.



Ці роботи стали основою для принципів автоматизації та формальної логіки міркувань, які використовуються в сучасних комп'ютерах, зокрема, в системах для підтримки прийняття рішень та розумних пошукових системах, покликаних доповнювати та примножувати можливості людини.

Хоча в науково-фантастичних фільмах і романах ШІ найчастіше зображають у вигляді людиноподібних роботів, що захоплюють владу над світом, на даному етапі розвитку технології ШІ далеко не такі розумні. Навпаки, розвиток штучного інтелекту дозволяє цим технологіям давати реальну користь в різноманітних галузях науки, техніки та повсякденних задачах [1].

Загалом розвиток штучного інтелекту умовно можна розділити на три етапи [3]:

1950-1970-ті рр. – Нейромережі. Перші розробки в галузі нейромереж викликали ажіотаж у зв'язку з можливістю створення "мислячих" машин.

1980-2010-ті рр. – Машинне навчання. Стають популярними технології машинного навчання.

Теперішній час – Глибоке навчання. Прориви у сфері глибокого навчання сприяли розквіту технологій ШІ.

### **1.1.3 Проблеми штучного інтелекту**

16 лютого 2022 року з'явилася інформація про те, що колишній виконавчий директор Google Ерік Шмідт (Eric Schmidt) оголосив про створення благодійного фонду із загальним капіталом 125 млн доларів, який сприятиме дослідженням у сфері штучного інтелекту. Йдеться насамперед про дослідження, спрямовані на вирішення кардинальних проблем, що виявляються при використанні технологій штучного інтелекту, у тому числі упередженості, можливості заподіяння шкоди та зловживань. У списку значаться також геополітичні конфлікти та наукова обмеженість самої технології.

В основі всього того, що є практикою ШІ (машинний переклад, розпізнавання мови, обробка текстів природними мовами, комп'ютерний зір, автоматизація водіння автомобілів і багато іншого) лежить глибинне навчання. Це підмножина машинного навчання, що відрізняється використанням моделей нейронних мереж, про які можна сказати, що вони імітують роботу мозку, тому їх з натяжкою можна віднести до ШІ. Будь-яка модель нейронної мережі навчається на великих наборах даних. Таким чином вона знаходить деякі «навички», але те, як вона ними користується – для розробників залишається не ясним, що в кінцевому рахунку стає однією з найважливіших проблем для багатьох додатків глибинного навчання[7]. Причина в тому, що така модель працює з образами формально, без розуміння того, що вона робить. Чи є така система ШІ і чи можна довіряти системам, побудованим на основі машинного навчання? Тому помітно загострилася увага до явища, що отримало назву AI bias. Його можна перекласти як «необ'єктивність ШІ» або «упередженість ШІ»[2].

Головне обмеження ШІ полягає в тому, що навчання можливе лише на основі даних. Це означає, що будь-які неточності даних позначаються на результатах. А нові рівні прогнозування чи аналізу необхідно додавати окремо.

Системи ШІ призначені для виконання чітко визначених завдань. Система, призначена для гри в покер, не може розкладати пасьянси або грати в шахи. Система, налаштована виявлення шахрайства, не зможе водити машину чи надавати правову допомогу. Іншими словами, ці системи характеризуються дуже тонкою спеціалізацією. Вони призначені для виконання одного конкретного завдання, і їм далеко до багатозадачності людини.

Крім того, самонавчальні системи не є автономними. Образи технологій ШІ, які ми бачимо на екранах телевізорів та кінотеатрів, як і раніше, є елементами фантастики. Проте комп'ютери, здатні аналізувати складні дані для освоєння та вдосконалення конкретних навичок, вже не рідкість.

### 1.1.4 Підходи та напрямки у дослідженнях ШІ

Єдиної відповіді на питання, чим займається штучний інтелект, немає. Майже кожен дослідник, який займається питаннями ШІ, відштовхується від будь-якого визначення, розглядаючи у його світлі досягнення тієї сфери, в якій він проводить дослідження.

У філософії не вирішено питання про природу та статус людського інтелекту. Немає і точного критерію досягнення комп'ютерами «розумності», хоча на перших етапах розвитку систем штучного інтелекту було запропоновано низку гіпотез, наприклад, тест Тьюрінга або гіпотеза Ньюелла Саймона. Тому, незважаючи на наявність безлічі підходів як до розуміння завдань ШІ, так і створення інтелектуальних інформаційних систем, можна виділити два основні підходи до розробки ШІ [7]:

- семіотичний (англ. Top-Down AI), низхідний – створення експертних систем, баз знань і систем логічного висновку, що імітують високорівневі психічні процеси: мислення, міркування, мовлення, емоції, творчість тощо;
- біологічний (англ. Bottom-Up AI), висхідний – вивчення нейронних мереж та еволюційних обчислень, що моделюють інтелектуальну поведінку на основі біологічних елементів, а також створення відповідних обчислювальних систем, таких як нейрокомп'ютер або біокомп'ютер.

**Тест Тьюрінга та інтуїтивний підхід.** Емпіричний тест був запропонований Аланом Тьюрінгом у статті «Обчислювальні машини і розум» [5] (Computing Machinery and Intelligence), опублікованій в 1950 році у філософському журналі «Mind». Метою даного тесту є визначення можливості штучного мислення, наближеного до людського. Стандартна інтерпретація цього тесту звучить так: «Людина взаємодіє з одним комп'ютером та однією людиною. На підставі відповіді на запитання він повинен визначити, з ким він розмовляє: з людиною чи комп'ютерною програмою. Завдання комп'ютерної програми – ввести людину в оману, змусивши зробити неправильний вибір. Усі учасники тесту не бачать один одного.

Найзагальніший підхід передбачає, що ШІ буде здатний проявляти поведінку, яка не відрізняється від людської, причому в нормальних ситуаціях [5]. Ця ідея є узагальненням підходу тесту Тьюринга, який стверджує, що машина стане розумною тоді, коли буде здатна підтримувати розмову зі звичайною людиною, і той не зможе зрозуміти, що говорить з машиною.

Також часто пропонують ще один підхід: ШІ виникне тоді, коли машина буде здатна відчувати та творити. Проте такий підхід навряд чи витримує критику за більш детального розгляду. Наприклад, нескладно створити механізм, який оцінюватиме деякі параметри зовнішнього або внутрішнього середовища і реагуватиме на їх несприятливі значення. Про таку систему можна сказати, що вона має почуття («біль» — реакція на спрацьовування датчика удару, «голод» — реакція на низький заряд акумулятора, тощо).

**Символьний підхід.** Історично символічний підхід був першим в епоху цифрових машин, оскільки саме після створення Лісп, першої мови символічних обчислень, у його автора виникла впевненість у можливості практично розпочати реалізацію цими засобами інтелекту. Символьний підхід дозволяє оперувати слабоформалізованими уявленнями та його смислами.

Успішність та ефективність вирішення нових завдань залежить від уміння виділяти лише суттєву інформацію, що потребує гнучкості в методах абстрагування. Тоді як звичайна програма встановлює один свій спосіб інтерпретації даних, через що її робота і виглядає упередженою та суто механічною. Інтелектуальне завдання у цьому випадку вирішує лише людина, аналітик чи програміст, не вміючи довірити цьому машині. В результаті створюється єдина модель абстрагування, система конструктивних сутностей та алгоритмів. А гнучкість та універсальність виливається у значні витрати ресурсів для не типових завдань, тобто система від інтелекту повертається до грубої сили.

Основна особливість символічних обчислень – створення нових правил у процесі виконання програми. Тоді як можливості не інтелектуальних систем завершуються якраз перед здатністю хоча б позначати труднощі, що знову

виникають. Тим не менш ці проблеми не вирішуються і нарешті комп'ютер не вдосконалює такі можливості самостійно.

Недоліком символного підходу і те, що такі відкриті можливості сприймаються не підготовленими як відсутність інструментів. Цю скоріше культурну проблему частково вирішує логічне програмування.

**Логічний підхід** до створення систем штучного інтелекту ґрунтується на моделюванні міркувань. Теоретичною основою є логіка. Такий підхід може бути проілюстрований застосуванням цих цілей мови та системи логічного програмування Пролог. Програми, записані мовою Пролог, представляють набори фактів і правил логічного висновку без жорсткого завдання алгоритму як послідовності дій, які призводять до необхідного результату.

**Агентно-орієнтований підхід.** Останній підхід, що розвивається з початку 1990-х років, називається агентно-орієнтованим підходом або підходом, заснованим на використанні інтелектуальних (раціональних) агентів. Відповідно до цього підходу, інтелект – це обчислювальна частина (грубо кажучи, планування) здатності досягати поставлених перед інтелектуальною машиною цілей. Сама така машина буде інтелектуальним агентом, що сприймає навколишній світ за допомогою датчиків, і здатна впливати на об'єкти в навколишньому середовищі за допомогою виконавчих механізмів.

Цей підхід акцентує увагу на тих методах та алгоритмах, які допоможуть інтелектуальному агенту виживати у навколишньому середовищі під час виконання його завдання. Так, тут значно ретельніше вивчаються алгоритми пошуку шляху та прийняття рішень.

**Гібридний підхід** передбачає, що тільки синергетична комбінація нейронних та символних моделей досягає повного спектру когнітивних та обчислювальних можливостей. Наприклад, експертні правила висновків можуть генеруватися нейронними мережами, а породжуючі правила отримують за допомогою статистичного навчання. Прихильники цього

підходу вважають, що гібридні інформаційні системи будуть значно сильнішими, ніж сума різних концепцій окремо.

### 1.1.5 Принцип роботи штучного інтелекту

Принцип роботи ШІ полягає у поєднанні великого обсягу даних з можливостями швидкої, ітеративної обробки та інтелектуальними алгоритмами, що дозволяє програмам автоматично навчатися на базі закономірностей та ознак, що містяться у даних. ШІ є комплексною дисципліною з безліччю теорій, методик і технологій. Її головними напрямками є такі:

– **Машинне навчання** – це область знань, що досліджує алгоритми, які навчаються на даних з метою знайти закономірності. У ньому використовуються методи нейромереж, статистики, дослідження операцій тощо. виявлення прихованої корисної інформації даних; при цьому явно не програмується інструкції, що вказують, де шукати дані та як робити висновки.

– **Нейромережі** – це один з методів машинного навчання. Це математична модель, і навіть її програмне чи апаратне втілення, побудована за принципом організації та функціонування біологічних нейронних мереж – мереж нервових клітин живого організму.

– **В глибокому навчанні** використовуються складні нейромережі з безліччю нейронів та шарів. Для навчання цих глибоких нейромереж, а також для виявлення складних закономірностей у великих масивах даних застосовуються підвищені обчислювальні потужності та вдосконалені методики. Поширені області застосування: розпізнавання зображень та мовлення.

– **Когнітивні обчислення** – напрямок ШІ, завданням якого є забезпечення процесу природної взаємодії людини з комп'ютером, аналогічного взаємодії між людьми. Кінцева мета ШІ та когнітивних обчислень — імітація

когнітивних процесів людини комп'ютером завдяки інтерпретації зображень та мовлення з видачею відповідної реакції у відповідь.

– **Комп'ютерний зір** спирається на розпізнавання шаблонів та на глибоке навчання для розпізнавання зображень та відео. Машини вже вміють обробляти, аналізувати та розуміти зображення, а також знімати фото чи відео та інтерпретувати навколишнє оточення.

– **Обробка природної мови** – це здатність комп'ютерів аналізувати, розуміти та синтезувати людську мову, включаючи усне мовлення. Зараз ми вже можемо керувати комп'ютерами за допомогою звичайної мови, яка використовується у повсякденному побуті. Наприклад, використовуючи Siri або Google Assistant.

Технології, котрі забезпечують функціонування ІІІ:

– **графічні процесори** – надають обчислювальні потужності, необхідні для ітеративної обробки даних. Для навчання нейромереж необхідні «великі дані» та обчислювальні ресурси.

– **інтернет речей** збирає величезні обсяги даних від підключених пристроїв. Більшість цих даних не проаналізовано. Автоматизація моделей за допомогою ІІІ дозволить використовувати більше таких даних.

– розробляються та по-новому комбінуються більш **досконалі алгоритми**, які дозволяють швидше аналізувати більший обсяг даних одразу на кількох рівнях. Така інтелектуальна обробка – ключ до виявлення та прогнозування малоймовірних подій, розуміння складних систем та оптимізації унікальних сценаріїв.

– **API (програмні інтерфейси додатків)** являють собою пакети коду, що переносяться, завдяки яким функціонал ІІІ може бути інтегрований в існуючі продукти і пакети програм. За допомогою API можна додати функцію розпізнавання зображень в домашню систему безпеки або питання-відповіді для опису даних, створення титрів і заголовків, виявлення в даних цікавих закономірностей та іншої корисної інформації [7].

Тобто, на сьогодні, мета ШІ – забезпечення роботи програмних продуктів, здатних до аналізу вхідних даних та інтерпретації отриманих результатів. Штучний інтелект – засіб, що забезпечує інтуїтивніший процес взаємодії людини з програмами та допомогу при прийнятті рішень у рамках певних завдань.

## 1.2 Мікропроцесорні платформи для систем з ШІ

Для розгортання обчислювальної системи, яка виконує обробку великих масивів даних із застосуванням програмних засобів з елементами штучного інтелекту добре підходять одноплатні комп'ютери, які набули сьогодні широкого застосування у найрізноманітніших областях.

Одноплатний комп'ютер – це платформа, до складу якої входить мікропроцесор, пам'ять та базові периферійні пристрої, і все це розміщується на одній платі. За розрахунковою потужністю такі пристрої прирівнюються до сучасних смартфонів, але мають ряд переваг у порівнянні із ними. Одноплатний комп'ютер не розміщується в компактному корпусі, тому проблема охолодження центрального процесора та графічних ядер може бути вирішена традиційним способом – встановленням системи активного чи пасивного охолодження [4]. Також до такого пристрою можна підключати периферійні пристрої за допомогою інтегрованих портів введення/виведення.

І головною перевагою є те, що на них може бути розгорнута операційна система загального призначення (здебільшого Linux), що дозволяє використовувати стандартні засоби програмування а також системи криптографічного та іншого захисту[3]. В роботі ми розглянемо два популярних одноплатних комп'ютера, які активно використовуються при розробці систем аналізу даних з елементами штучного інтелекту, а саме Raspberry Pi та NVIDIA Jetson Nano.



## 1.2.1 Огляд платформи Raspberry Pi

Raspberry Pi – це мініатюрний одноплатний комп'ютер, котрий може поміститися на долоні людини. Незважаючи на свої мініатюрні розміри, плата має достатньо високу продуктивність, що дозволяє вийти на один рівень обчислювальної потужності зі персональними комп'ютерами. Спочатку Raspberry Pi була розроблена як навчальний посібник з інформатики. Але сама ідея виявилася настільки вдалою, що за кілька років міні-комп'ютер став популярним у дуже широких колах. З часом Raspberry Pi пережила кілька модифікацій, кожна з яких відрізнялася від попередника покращеними параметрами. Такий підхід дозволив регулювати вартість виробу в залежності від потреб користувача, що також позитивно позначилося на популярності пристрою. Уся лінійка Raspberry Pi застосовує процесори з APM-архітектурою, яка зарекомендувала себе з найкращого боку.

На платі мікрокомп'ютера Raspberry Pi (рис. 1.1) розміщуються наступні компоненти: Центральний процесор ARM, Ethernet порт, порти USB, WiFi та Bluetooth модулі, порт аналогового аудіовиходу, HDMI вихід, роз'єм DSI дисплея, роз'єм для CSI камери та 40-выводний GPIO (порти введення/виведення з підтримкою протоколів UART, I2C, SPI, PWM та ін.).

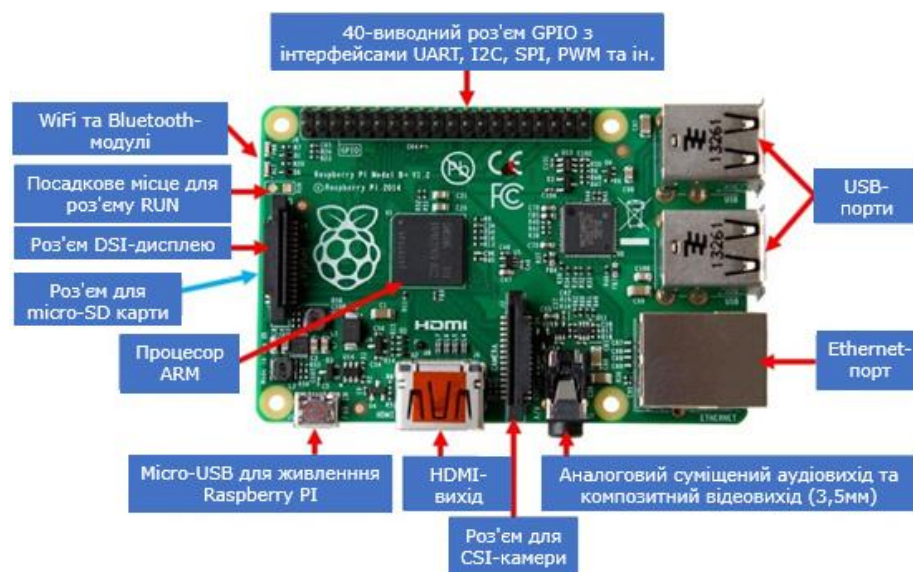


Рисунок 1.1 – Одноплатний мікрокомп'ютер Raspberry Pi. Адаптовано з роботи [4]

На сьогоднішній день існує 11 модифікацій комп'ютера Raspberry Pi [6]. Останні версії оснащені бездротовими Wi-Fi та Bluetooth модулями, що розширюють межі застосування міні-ПК в області Ethernet-технологій. В таблиці 1.1 наведено порівняння різних модифікацій Raspberry Pi, а також відображені особливості кожної модифікації із зазначенням деяких технічних характеристик. Як видно з наведеної таблиці, навіть наймолодша модель в лінійці має цілком серйозні характеристики, враховуючи те, що це одноплатний комп'ютер розміром трохи більше кредитної картки.

Таблиця 1.1

## Порівняння технічних характеристик Raspberry Pi різних модифікацій

Модель	Тактова частота	К-ть ядер	Об'єм ОЗП	К-ть GPIO	К-ть USB	Підтримка Ethernet	Підтримка WiFi	Підтримка Bluetooth
B	700 МГц	1	512 Мб	26	2	Так	Ні	Ні
A	700 МГц	1	256 Мб	26	1	Ні	Ні	Ні
B+	700 МГц	1	512 Мб	40	4	Так	Ні	Ні
A+	700 МГц	1	256 Мб	40	1	Ні	Ні	Ні
2B	900 МГц	4	1 Гб	40	4	Так	Ні	Ні
Zero	1 ГГц	1	512 Мб	40	1	Ні	Ні	Ні
3B	1,2 ГГц	4	1 Гб	40	4	Так	Так	4.1
Zero W	1 ГГц	1	512 Мб	40	1	Ні	Так	4.0
3B+	1,4 ГГц	4	1 Гб	40	4	Так	Так	4.2
3A+	1,4 ГГц	4	512 Мб	40	1	Ні	Так	4.2
4B	1,5 ГГц	4	1, 2, 4 Гб	40	4	Так	Так	5.0

Хоча Raspberry Pi зовні може нагадати нам Arduino, він все-таки використовує кардинально інший спосіб функціонування. Ця плата, як і стандартний ПК, працює під управлінням однієї зі спеціалізованих

операційних систем. Залежно від сфери застосування або особистих симпатій, кожен може вибрати для себе свою. Нижче наведено перелік найпопулярніших «операцій» для Raspberry Pi з їх коротким описом.

Raspbian – операційна система в 2015 році була представлена як основна для Raspberry Pi. Вона максимально оптимізована для процесорів з ARM-архітектурою і досить активно продовжує розвиватися. До складу дистрибутива входять програма комп'ютерної алгебри Mathematica, модифікована версія Minecraft PI та урізана версія Chrome.

Debian – операційна система з відкритим вихідним кодом. У стандартний дистрибутив включені: середовище робочого столу GNOME з набором найпопулярніших програм, таких як Firefox, LibreOffice, Evolution та інший набір для роботи з мультимедіа.

Ubuntu – система заснована на Debian GNU/Linux. За популярністю Ubuntu посідає перше місце серед дистрибутивів Linux, призначених для веб-серверів.

Fedora – операційна система заснована на дистрибутиві Linux компанії Red Hat.

Gentoo Linux – один із популярних дистрибутивів GNU/Linux. У системі передбачено можливість максимальної оптимізації під конкретне апаратне рішення. Алгоритм управління пакетами дає можливість легко реалізувати як робочу станцію, так і сервер.

RISC OS – Операційна система спеціально розроблялася для процесорів з архітектурою ARM. Особливості ядра RISC OS дозволяють системі проводити прискорений запуск за рахунок зберігання даних у ПЗУ. Такий підхід також допомагає захистити дані при різноманітних збоях та впливу шкідливого ПЗ.

У мережі Інтернет, крім перерахованих операційних систем, можна знайти ще безліч модифікацій для найрізноманітніших призначень.

Основною відмінністю Raspberry Pi від звичайних комп'ютерів, є наявність програмованих портів введення-виведення GPIO. За допомогою них

можна керувати різними пристроями та приймати телеметрію з різноманітних датчиків. Користувачеві доступна можливість управління цими виводами, а це означає, що до Raspberry Pi можна підключати дисплеї, кнопки, датчики, реле та інші електронні модулі, якими можна керувати на свій розсуд.

Зовні GPIO виконаний у вигляді дворядної штирьової колодки, розташованої на краю плати. Ранні моделі, такі як і А містять 26 виводів [6],



а

	3V3	1	2	5V	
SDA1 I2C	GPIO2	3	4	5V	
SDL1 I2C	GPIO3	5	6	GND	
	GPIO4	7	8	GPIO14	UART0_TX
	GND	9	10	GPIO15	UART0_RX
	GPIO17	11	12	GPIO18	PCM_CLK
	GPIO27	13	14	GND	
	GPIO22	15	16	GPIO23	
	3V3	17	18	GPIO24	
SPI0_MOSI	GPIO10	19	20	GND	
SPI0_MISO	GPIO9	21	22	GPIO25	
SPI0_SCLK	GPIO11	23	24	GPIO8	SPI0_CE0_N
	GND	25	26	GPIO7	SPI0_CE1_N
I2C ID EEPROM	ID_SD	27	28	ID_SC	I2C ID EEPROM
	GPIO5	29	30	GND	
	GPIO6	31	32	GPIO12	
	GPIO13	33	34	GND	
	GPIO19	35	36	GPIO16	
	GPIO26	37	38	GPIO20	
	GND	39	40	GPIO21	

б

Рис. 1.2 – Виводи GPIO плати Raspberry Pi 3B+ (а) та їх розпіновка (б).

Адаптовано з роботи [6]

а більш сучасні – 40. На рисунку 1.2 (а) показаний зовнішній вигляд портів загального призначення для плати Raspberry Pi 3B+ із зазначенням нумерації виводів.

Але щоб повноцінно використовувати GPIO, знати їх нумерацію недостатньо. Необхідно точно розуміти, де розташований той чи інший вивод, як він називається і за що він відповідає. На рис. 1.2 (б) наведено повне розпінування роз'єму GPIO для модифікацій Raspberry Pi A+ та B+ і вище.

За класикою, програми для Raspberry Pi пишуться скриптовою мовою програмування Python. Особливість її в тому, що для запуску програми не потрібний компілятор. Скрипт запускається і починає роботу відразу, але його необхідно зберегти до файлу із подальшим завантаженням у плату. Ще однією важливою особливістю програми Python є суворе дотримання табуляцій під час написання коду програм [18].

### **1.2.2 Огляд платформи NVIDIA Jetson Nano**

Платформа Jetson Nano (рис. 1.3), так само, як і Raspberry Pi, представлена у вигляді одноплатного комп'ютера зі схожими габаритами, та великою мірою сумісна з її периферійними пристроями. Даний пристрій спеціально заточений для роботи з додатками, котрі використовують у своїй роботі елементи штучного інтелекту.

Jetson Nano поставляється у двох версіях:

- комплект для розробників, виробників та ентузіастів;
- готовий модуль для компаній, які бажають створити системи для масового ринку.

Jetson Nano підтримує сенсори високої роздільної здатності, може паралельно обробляти інформацію з великої кількості датчиків і запускати кілька нейромереж одночасно [10]. Він також підтримує безліч популярних фреймворків ШІ, що дозволяє розробникам інтегрувати свої улюблені моделі та фреймворки.

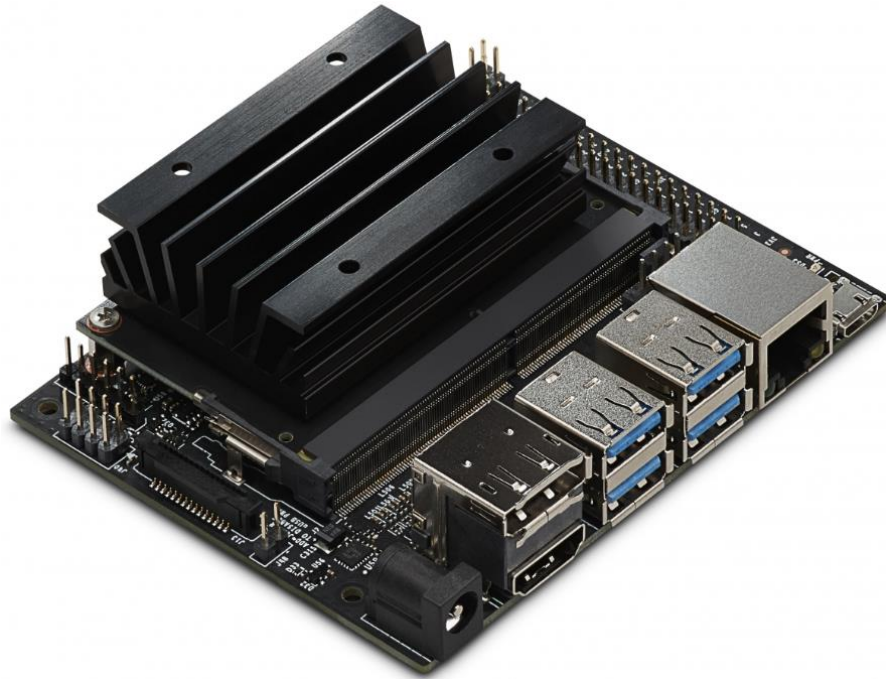


Рис. 1.3 – Зовнішній вигляд платформи Jetson Nano

На думку NVIDIA, дешевий одноплатник «ідеально підходить для підприємств, стартапів та дослідників», які раніше не могли собі дозволити купити дорожчі плати. Таким чином, платформа Jetson значно розширює свою аудиторію, а прискорювач ШІ фактично стає мало не товаром народного вжитку. Як каже Nvidia, плата «приносить силу сучасного ШІ на недорогу платформу, стимулюючи нову хвилю інновацій від виробників, винахідників, розробників та студентів. Вони можуть створювати проекти ШІ, які раніше були неможливі, і виводити існуючі проекти на новий рівень – мобільні роботи та дрони, цифрові помічники, автоматизовані прилади та багато іншого»[10].

Технічні характеристики платформи Jetson Nano:

- Графічний процесор: Nvidia на архітектурі Maxwell зі 128 ядрами CUDA;
- Процесор: чотириядерний ARM Cortex-A57 MPCore;
- Відео: 4K на 30 кадрах за секунду (формат H.264/H.265) та 4K на 60 кадрах за секунду (формат H.264/H.265) для кодування та декодування, відповідно;
- Відеовихід: HDMI 2.0 чи DP1.2, eDP 1.4, DSI (1×2), два одночасно;

- Оперативна пам'ять: 4 ГБ LPDDR4 64-біт; 25,6 ГБ/сек;
- Флеш-пам'ять: 16 ГБ eMMC;
- Камера: 12 ліній (3×4 або 4×2) MIPI CSI-2 DPHY 1.1 (1,5 Гбіт/с), 12х (модуль) та 1х (набір розробника);
- Роз'єми: 1×1/2/4 PCIE, 1×USB 3.0, 3×USB 2.0;
- Введення/виведення: 1×SDIO / 2×SPI / 6×I2C / 2×I2S / піни GPIO;
- Мережа: Gigabit Ethernet;
- Підтримка ОС: Linux для Tegra;
- Розмір модуля: 69,5×45 мм;
- Розмір комплекту розробника: 100×80 мм;
- Підключення: 260-піновий конектор.

Виходячи з технічних характеристик Jetson Nano варто відзначити його особливості:

– Перше – 128 ядерний графічний процесор GPU, відповідно, на платі можна запускати GPU-орієнтовані завдання, на зразок CUDA (підтримується та встановлений з «коробки») або Tensorflow. Основний процесор 4-х ядерний, заснований на технології ARM. Пам'ять 4 Гб, спільна між CPU та GPU.

– Друге – це сумісність із Raspberry Pi. Плата має 40-піновий роз'єм з різними інтерфейсами (I2C, SPI та ін), також є роз'єм камери, який також сумісний з Raspberry Pi. Можна припустити, що велика кількість вже наявних аксесуарів (екрани, плати управління моторами та ін.) будуть працювати.

– Третє – на платі є 2 відеовиходи, Gigabit Ethernet і USB 3.0, тобто, Jetson Nano в цілому навіть трохи більш функціональна, ніж Raspberry Pi. Живлення 5В може братися як по Micro USB, так і через окремий роз'єм, який рекомендується для ресурсомістких завдань. Як і в Raspberry Pi, програмне забезпечення вантажиться з SD-карти, образ якої потрібно попередньо записати. Загалом, за ідеологією, плата цілком схожа на Raspberry Pi, що й замислювалося в NVIDIA.

Конструктивно пристрій складається з двох модулів – власне модуля Jetson Nano, і нижньої плати з конекторами, з'єднання цих модулів здійснюється через спеціальний роз'єм.

Загалом Jetson Nano не претендує на заміну Raspberry PI, але для ресурсомістких завдань вона більше підходить, оскільки графічний процесор краще справляється із математичними розрахунками, при яких необхідно виконувати одночасно велику кількість паралельних потоків.



## РОЗДІЛ 2 МЕТОДИКА ТА ТЕХНІКА ЕКСПЕРИМЕНТУ

### 2.1 Апаратне забезпечення для розробки системи аналізу даних

В рамках магістерської роботи нами була спроектована система аналізу даних з елементами штучного інтелекту. Проектування системи було проведено з використанням апаратної платформи Raspberry Pi на базі мікропроцесора. Нами було обрано для реалізації дві функціональні можливості з самонавчанням [9], які демонструють елементи штучного інтелекту, а саме розпізнавання голосу та розпізнавання візуальних елементів у потоковому відео.

Проект базується на Raspberry Pi 3B (рис. 2.1), так як цей пристрій має характеристики, які задовільняють поставленим задачам, а саме:

- процесор: ARM Cortex-A53, 64-х бітний, частота 1,2 ГГц, 4-х ядерний;
- ОЗП: 1 Гб;
- GPIO: 40 пінів;
- USB: 4 шт., 2.0;
- Ethernet: RJ45 Ethernet;
- WiFi: 802.11n;
- Bluetooth: 4.1.

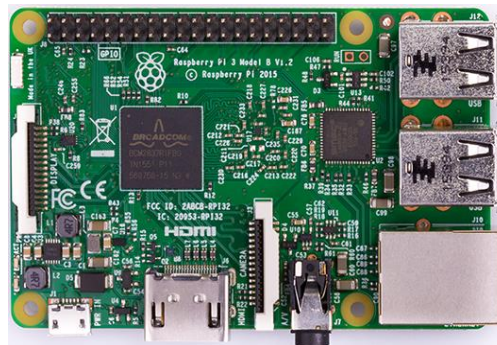


Рис. 2.1 – Raspberry Pi 3B

Також нам необхідно використовувати камеру для отримання відеопотоку в режимі реального часу. Проаналізувавши асортимент камер, сумісних з Raspberry Pi 3B, було розглянуто 2 камери: Raspі Cam V1 і Raspі Cam V2.

Raspі Cam V1.3 (рис. 2.2 а) має п'ятимегапіксельний сенсор, автоматичний фокус, записує відео в трьох режимах: 1080р – 30 кад/сек., 720р – 60 кад/сек., VGA – 60-90 кад/сек., роздільна здатність фото – 2592x1944 пікселів. Повністю підтримує останню версію Raspbian.

Raspі Cam V2 (рис. 2.2 б) – оновлена версія попередньої камери. Ця версія має покращений восьмимегапіксельний сенсор, якість запису відео ідентична, роздільна здатність фото покращена – 3280x2464 пікселів.

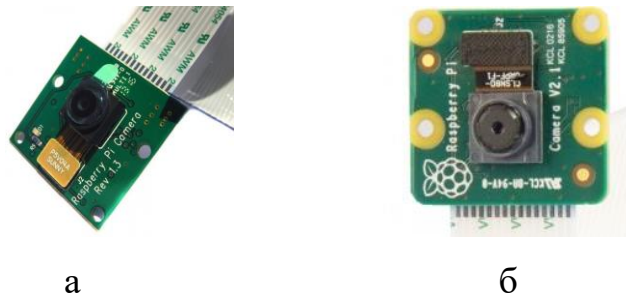


Рис. 2.2 – Модулі камери Raspі Cam V1.3 (а) та Raspі Cam V2 (б)

Проаналізувавши технічні характеристики обох камер, була обрана RaspіCam V1.3, бо вона не поступається якості запису відео, дешевша і якість фото досить чітка.

Для забезпечення стабільності роботи мікрокомп'ютера необхідно підтримувати її нормальний тепловий режим роботи. Щоб плата не перегрівалася, можна використати радіатори пасивного охолодження, краще алюмінієві. Вони кріпляться до плати (Рисунок 2.3) за допомогою теплопровідної клейкої стрічки або термоклею.



Рисунок 2.3 – Радіатори для Raspberry Pi 3B

Для того, щоб пристрій мав змогу отримувати аудіодані, необхідний мікрофон [13]. Проаналізувавши асортимент мікрофонів, було розглянуто два мікрофони: SunFounder USB міні (рис. 2.4 а) та Elecrow модель E (рис. 2.4 б). Обидва мікрофони мають достатні характеристики, однак був обраний перший, оскільки він значно менший по габаритам і дешевший.



а



б

Рис. 2.4 – Мікрофон Sun Founder USB міні (а) та Elecrow модель E (б)

Також важливим компонентом є SD-карта, на якій буде інстальована операційна система. Була обрана SD-карта фірми Team Group об'ємом 16 Гб з наступними характеристиками:

- швидкість читання від/до: 17 Мб/с до 20 Мб/с;
- швидкість запису від/до: 10 Мб/с до 14 Мб/с;
- клас: 10;
- тип: micro-SD.

Також необхідно обрати серводвигун. Проаналізувавши асортимент доступних серводвигунів, було розглянуто такі моделі: Tower Pro 9g SG90 (рис. 2.5 а), Futaba S3004 (рис. 2.5 б) та Power HD 1711MG (рис. 2.5 в).

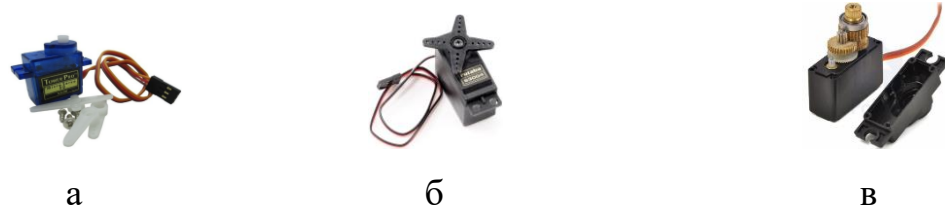


Рис. 2.5 – Серводвигуни Tower Pro 9g SG90 (а), Futaba S3004 (б) та Power HD 1711MG (в)

Серводвигун Tower Pro 9g SG90 – це серводвигун, який здатний повертати вихідний вал в задане положення (на заданий кут) і утримувати його. Він має наступні характеристики:

- вага: 9 грам;
- габарити: 21.5 мм х 11.8 мм х 22.7 мм;
- напруга живлення: 4.8 ... 6 В;
- момент (зусилля): 1.2 кг\*см при напрузі 4.7 В;
- час повороту на 60°: 0.12 сек при напрузі 4.7 В;
- робоча температура: -30°С...+60°С;
- довжина кабелю: 23 см;
- матеріал шестерні: нейлон;
- механіка: аналогова.

Серводвигун Futaba S3004 – аналоговий двигун, використовується для моделей літаків, машин та суден. Головна особливість серед бюджетних конкурентів – наявність одного підшипника коливання на головній шестерні зверху. Він має наступні характеристики:

- вага: 37 грам;
- габарити: 35 мм х 20 мм х 45 мм;
- напруга живлення: 4.8 ... 6 В;

- момент (зусилля): 3.2 кг\*см / 4.1 кг\*см при напрузі 4.8 В / 6 В;
- час повороту на 60°: 0.23 сек / 0.19 сек при напрузі 4.8 В / 6 В;
- робоча температура: -30°C...+60°C;
- довжина кабелю: 23 см;
- матеріал шестерні: пластик;
- механіка: аналогова.

Серводвигун Power HD 1711MG – потужний, швидкісний, аналоговий серводвигун з металевими шестернями в редукторі. Забезпечує додатковий високий обертаючий момент. Він має наступні характеристики:

- вага: 17.5 грам;
- габарити: 29.5 мм x 11.6 мм x 30.2 мм;
- напруга живлення: 6 В;
- момент (зусилля): 3.5 кг\*см при напрузі 6 В;
- час повороту на 60°: 0.11 сек при напрузі 6 В;
- робоча температура: -30°C...+60°C;
- довжина кабелю: 28 см;
- матеріал шестерні: метал;
- механіка: аналогова.

Проаналізувавши три серводвигуни, найкращим виявився Tower Pro 9g SG90, бо він має досить швидку реакцію, мало споживає енергії, компактний і відмінно підходить для поставлених у роботі задач.

## **2.2 Підготовка пристрою до роботи**

Для того, щоб почати налагодження пристрою, необхідний сам пристрій (Raspberry Pi 3B), модуль камери Raspi Cam V1.3, SD-карта на 16 Гб, мікрофон SunFounder USB, кабель Micro USB, кабель HDMI, монітор, клавіатура та мишка.

Далі необхідно завантажити операційну систему для Raspberry. На офіційному сайті доступно 2 основні ОС: NOOBS (програмне забезпечення, яке допомагає інсталювати бажану ОС) та Raspbian.

Raspbian – це офіційна операційна система для Raspberry Pi, вона розроблена спеціально для цього пристрою і має все необхідне програмне забезпечення. Raspbian основана на ARM версії Debian 8 Jessie і містить такі програми за замовчуванням – офісний пакет LibreOffice, веб-браузер, поштовий клієнт, легке оточення робочого столу, а також деякі інструменти для навчання програмуванню [16]. Нами було обрано дистрибутив Raspbian, оскільки він має всі необхідні інструменти для роботи з міні-комп'ютером. Після завантаження образ ОС необхідно розархівувати і записати на SD-карту об'ємом не менше 8 Гб. Записати можна програмою Win32DiskImager.

Далі підключаємо необхідні компоненти: серводвигун, мікрофон, динамік до Raspberry Pi 3B. Світлодіод підключаємо за допомогою джамперів: перша ніжка порожня, друга до 6-го піну (заземлення), третя до 1-го (живлення), четверта до 12-го піну (дані). Серводвигун підключаємо так: жовтий дріт (дані) до 26-го піну, червоний (живлення) до 2-го піну, коричневий (заземлення) до 14-го піну [6] (рис. 2.6).

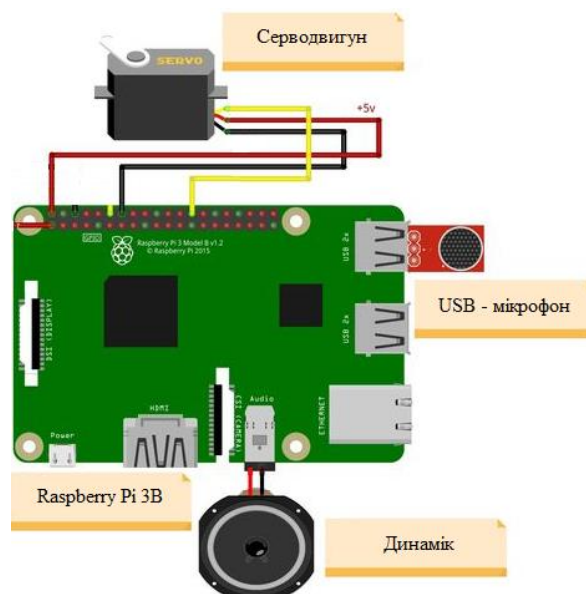


Рис. 2.6 – Принципова схема пристрою на базі Raspberry Pi.

Адаптовано з роботи [4]

Після цього вставляємо SD-карту з образом системи в Raspberry, під'єднуємо монітор за допомогою HDMI-кабелю, клавіатуру, мишку, модуль камери до роз'єму, який знаходиться між HDMI та Mini-Jack роз'ємів, підключаємо до джерела живлення і плата автоматично вмикається. Після завантаження системи потрібно налаштувати її [4]. Щоб запустити майстра налаштувань, запускаємо термінал поєднанням клавіш Ctrl+Alt+T, далі вводим команду `sudo raspi-config` (`sudo` – права суперкористувача), і відкривається майстер налаштування (рис. 2.7)[9].

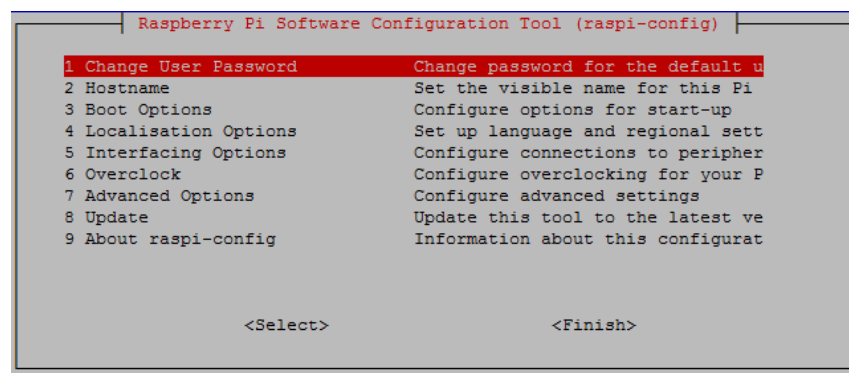


Рис. 2.7 – Майстер налаштування ОС Raspbian

Спочатку задамо пароль для обікового запису користувача. Для цього обираємо перший пункт, двічі вводимо пароль і нас повідомить про успішну зміну паролю. Далі потрібно збільшити кореневий розділ на весь об'єм карти пам'яті, для цього обираємо обираємо Advanced Options – Expand Filesystem і натискаємо ОК. Потім змінюємо ім'я пристрою, далі в розділі Interfacing Options обираємо перший пункт Camera – натискаємо Так. В наступному пункті SSH також натискаємо «Так» і перезавантажуємо систему [4].

SSH – мережевий протокол, який дозволяє керувати віддалено операційною системою. З його допомогою набагато легше керувати комп'ютерами.

Raspberry Pi 3В набагато зручніше керувати віддалено за допомогою програми PuTTY термінал. PuTTY – це клієнт для різноманітних протоколів

віддаленого доступу (SSH, Telnet, rlogin та ін.). Керувати можна як звичайним комп'ютером, так і сервером.

Для початку підключаємо Raspberry Pi до WiFi, в терміналі вводимо команду `ifconfig` і бачимо IP-адресу нашого пристрою (рис. 2.8), далі інсталуємо SSH-клієнт PuTTY на ПК, запускаємо клієнт, у вкладці Session в полі IP address вводимо IP-адресу, порт залишаємо 22 і бажано зберегти дану сесію: в полі Saved Sessions вводимо ім'я сесії і натискаємо Save. Далі натискаємо Load, запускається сесія, вводимо ім'я користувача (Pi) і пароль [16].

```
wlan0      Link encap:Ethernet  HWaddr b8:27:eb:7c:21:89
           inet addr:10.12.20.247  Bcast:10.12.20.255  Mask:255.255.255.0
           inet6 addr: fe80::7eae:2adf:4291:c07/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:9282 errors:0 dropped:703 overruns:0 frame:0
           TX packets:1339 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1482509 (1.4 MiB)  TX bytes:375523 (366.7 KiB)

pi@raspberrypi$
```

Рис. 2.8 – IP-адреса Raspberry Pi 3B

Тепер необхідно оновити систему [4]. Для цього в терміналі набираємо `sudo apt update`, натискаємо Enter, з'явиться повідомлення про пакети, які можна оновити, вводимо команду `sudo apt upgrade` і починається процес оновлення системи (Рисунок 2.9). На цьому початкове налаштування пристрою завершено.

```
Игн http://mirrordirector.raspbian.org jessie/main Translation-ru_RU
Игн http://archive.raspberrypi.org jessie/ui Translation-ru_RU
Игн http://mirrordirector.raspbian.org jessie/main Translation-ru
Игн http://mirrordirector.raspbian.org jessie/main Translation-en
Игн http://archive.raspberrypi.org jessie/ui Translation-ru
Игн http://mirrordirector.raspbian.org jessie/non-free Translation-ru_RU
Игн http://mirrordirector.raspbian.org jessie/non-free Translation-ru
Игн http://archive.raspberrypi.org jessie/ui Translation-en
Игн http://mirrordirector.raspbian.org jessie/non-free Translation-en
Игн http://mirrordirector.raspbian.org jessie/rpi Translation-ru_RU
Игн http://mirrordirector.raspbian.org jessie/rpi Translation-ru
Игн http://mirrordirector.raspbian.org jessie/rpi Translation-en
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
All packages are up to date.
pi@raspberrypi$ sudo apt upgrade
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Расчёт обновлений... Готово
обновлено 0, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 п
акетов не обновлено.
pi@raspberrypi$
```

Рис. 2.9 – Завершення процесу оновлення програмного забезпечення



## РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРИСТРОЮ ОБРОБКИ ДАНИХ НА БАЗІ RASPBERRY PI

### 3.1 Налаштування програмного забезпечення для розпізнання голосових команд

Для того, щоб пристрій міг сприймати, аналізувати та відповідати на команди, можна скористатись загальнодоступними програмними сервісами з відкритим програмним доступом. На даний момент активно використовуються та розвиваються сервіси такого класу від компаній Google та IBM Watsons [17].

Компанія Google розробляють сервіси розпізнавання осіб, перетворення мови в текст, визначення думок і емоцій в текстах – інструменти, які можуть стати доступними розробникам найближчим часом для Raspberry Pi. На даний момент Google завершує розробку великого спектру інструментів для машинного навчання, Інтернету речей, переносних пристроїв, робототехніки і домашньої автоматизації, перетворення природньої мови в текст, розпізнавання емоцій, аналізу пристроїв та інші технології в області машинного навчання і штучного інтелекту [20].

Розпізнавання мови – найвідоміша із технологій штучного інтелекту Google: вона є основою Google Assistant, віртуального голосового асистенту, доступного на смартфонах, планшетах, телевізорах і бездротовому динаміку Google Home [13]. Але в розумних пристроях на одноплатних комп'ютерах Raspberry Pi поки ще не доступна.

Сервіси IBM Watson представляють собою когнітивну систему, котра здатна розуміти, робити висновки і навчатися. Вона також дозволяє працювати у різних галузях та напрямках науки і техніки. Наприклад, передбачати появу епідемій або виникнення вогнищ, природних катастроф у різних регіонах, вести моніторинг стану атмосфери великих міст, оптимізувати бізнес-процеси,

дідзаватися, які товари будуть в тренді найближчим часом. Зараз на основі цієї хмарної API-платформи розробляються різні програми, сервіси і процеси. У Watson на даний момент 28 API і в основі всього цього лежить понад 50 різних технологій [13].

Також у процесі створення системи з елементами штучного інтелекту нам знадобиться Node.js – серверна платформа, розроблена Райаном Далем у 2009 році для роботи з JavaScript. JavaScript виконує дії на стороні клієнта, а Node – на сервері. За допомогою Node можна писати повноцінні програми. Node вмiє працювати із зовнішніми бібліотеками, викликати команди із коду на JavaScript і виконувати роль веб-серверу. Основною перевагою є простота масштабування. При одночасному підключенні до серверу тисяч користувачів, Node працює асинхронно, тобто ставить пріоритети і розподіляє ресурси грамотніше [16]. Java виділяє на кожне підключення окремий потік.

Так як Watson має багато сервісів, було обрано саме його. Пристрій здатний виконувати голосові команди, спілкуватися. Для цього необхідні наступні сервіси: Watson Speech to Text API (аналізує мову людини і перетворює в текст), Watson Conversation (оброблює текст і дає відповідь), Watson Text to Speech (перетворює текст в аудіо) (рис. 3.1). Тобто на голосовий запит бот відповідає своїм голосом [13]. За допомогою функцій розробника, можна редагувати, додавати власні фрази, створювати діалоги.

Для початку вмикаємо Raspberry Pi 3B, по SSH підключаємося до неї, вводимо логін і пароль. Далі встановлюємо Node.js і npm (пакетний менеджер) (ці пакети необхідні для запуску коду) наступними командами:

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo
-E bash -
sudo apt-get install -y nodejs
sudo apt-get install alsa-base alsa-utils libasound2-
dev
```

Далі додаємо необхідний репозиторій і завантажуємо 2 папки з файлами конфігурації, переходимо в папку Conversation та Tjwave і інсталуємо усі залежності (Рис. 3.2).

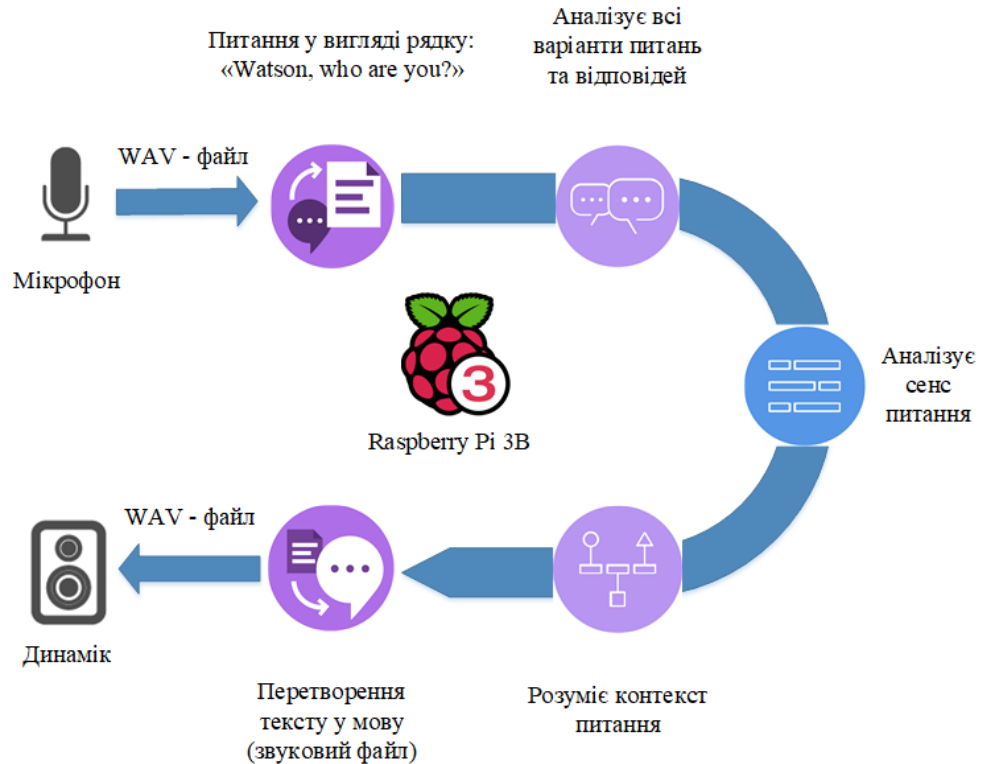


Рис. 3.1 – Структурна схема розпізнавання мови системою Watson

```

solr-client@0.7.0
duplexer@0.1.1
httperror@0.2.3
hnp@0.0.1
json-bigint@0.1.4
  bignumber.js@1.1.1
JSONStream@1.0.7
  jsonparse@1.3.1
  through@2.3.8
string@3.3.3
vcap_services@0.3.4
websocket@1.0.24
  debug@2.6.8
  ms@2.0.0
  typedarray-to-buffer@3.1.2
  yaeti@0.0.6
winston@2.3.1
  async@1.0.0
  colors@1.0.3
  cycle@1.0.3
  eyes@0.1.8
  stack-trace@0.0.10
pi@raspberrypiscpp:~/tjbot/recipes/conversation $

```

Рис. 3.2 – Створення залежностей програмних пакетів та їх успішна інсталяція

Далі необхідно зареєструватися на сайті Bluemix, де будемо використовувати сервіси Watsons. Обліковий запис реєструється умовно-безкоштовно і протягом 30-ти днів можна використовувати до 10 сервісів. Після реєстрації підтверджуємо обліковий запис за допомогою електронної пошти, після цього буде наданий доступ до списку сервісів Watsons (рис. 3.3).

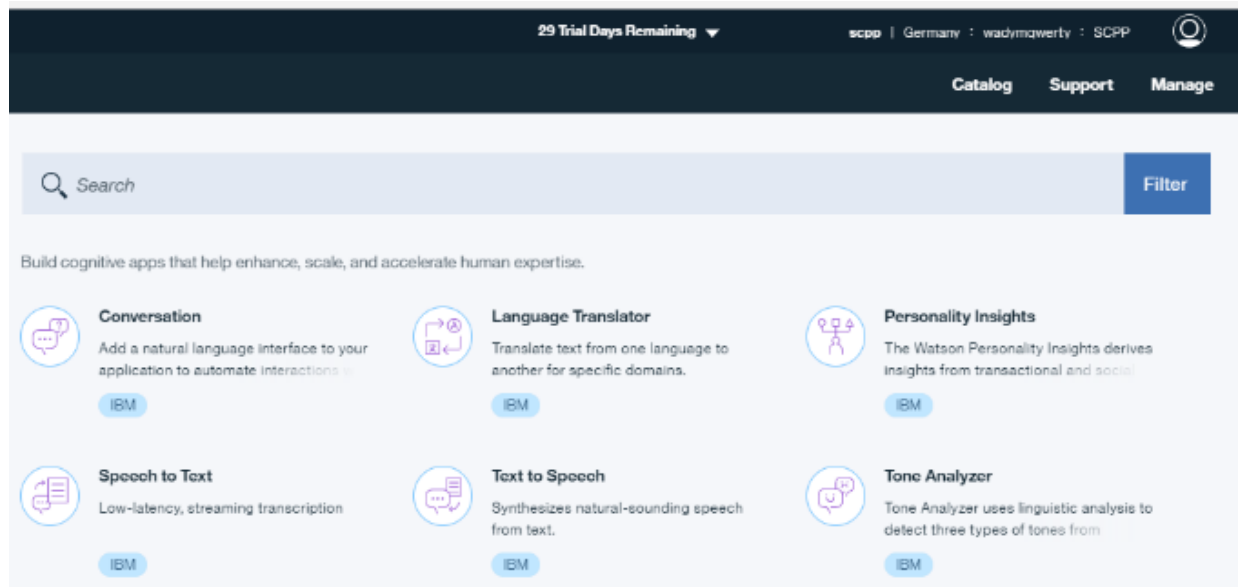


Рис.3.3 – Список доступних після реєстрації сервісів Watsons

Для реалізації необхідного функціоналу у рамках кваліфікаційної роботи потрібні наступні сервіси: Watson Speech to Text (мова людини в текст), Watson Conversation (діалог) і Watson Text to Speech (текст в мову). Потрібно скопіювати облікові дані кожного сервісу у файл config.js. Для цього створюємо нові дані у 3-х сервісах і копіюємо їх (username і password). Відкриваємо файл config.js командою `sudo nano config.js` і копіюємо в нього дані у відповідний рядок (рис. 3.4).

```

pi@raspberrypiscpp:~/tjbot/recipes/conversation $ sudo nano config.js
GNU nano 2.2.6 File: config.js

// User-specific configuration
exports.conversationWorkspaceId = ''; // replace with the workspace identifier $

// Create the credentials object for export
exports.credentials = {};

// Watson Conversation
// https://www.ibm.com/watson/developercloud/conversation.html
exports.credentials.conversation = {
  password: '',
  username: ''
};

// Watson Speech to Text
// https://www.ibm.com/watson/developercloud/speech-to-text.html
exports.credentials.speech_to_text = {
  password: '',
  username: ''
};

[ Read 26 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos

```

Рис. 3.4 – Додавання сервісів у конфігураційний файл

Далі потрібно навчити бота, що і коли говорити. Для цього використовуємо Watson Conversation. Для цього необхідно імпортувати зразок діалогу з носія в Raspberry Pi 3B. Спочатку скопіюємо файл workspace-sample.json за допомогою програми FileZilla.

Далі в менеджері сервісів, обираємо Conversation – у вкладці зліва обираємо Manage – Launch Tool – import - обираємо діалоговий конфігураційний файл workspace-sample.json.

Після імпортування файлу для його перегляду клікаємо по ньому і бачимо основні модулі діалогу. Тепер необхідно скопіювати облікові дані конфігураційного файлу: натискаємо на три крапки – View details, копіюємо Workspace ID в файл конфігурації config.js (рис. 3.5).

```

// User-specific configuration
exports.conversationWorkspaceId = '8337116d-ea46-47f5-9dde-49f2865c4eed';

```

Рис. 3.5 – Номер облікового запису Workspace ID

Після налаштування потрібно протестувати роботу серводвигуна і сервісів. Для цього необхідно перейти в папку `tjwave` командою `cd tjbot/recipes/tjwave` і запустити файл `wavetest.js` командою `sudo node wavetest`, відбудеться тестування серводвигуна.

Залишилось протестувати сервіси та їх інтеграцію з компонентами [8]. Для цього в папці `conversation` запускаємо файл `conversation.js`, починається ініціалізація пристроїв та сервісів, необхідних для роботи. У кінцевому варіанті система голосового керування працюватиме відповідно до схеми показаної на рис. 3.6. Система за допомогою мікрофона отримує голосове повідомлення, далі проходить його інтелектуальний аналіз за посередництва сервісів IBM Watsons, після чого приймається рішення про виконання необхідних операцій, у нашому випадку керування положенням серводвигуна і звітування про виконання команди у голосовому вигляді.

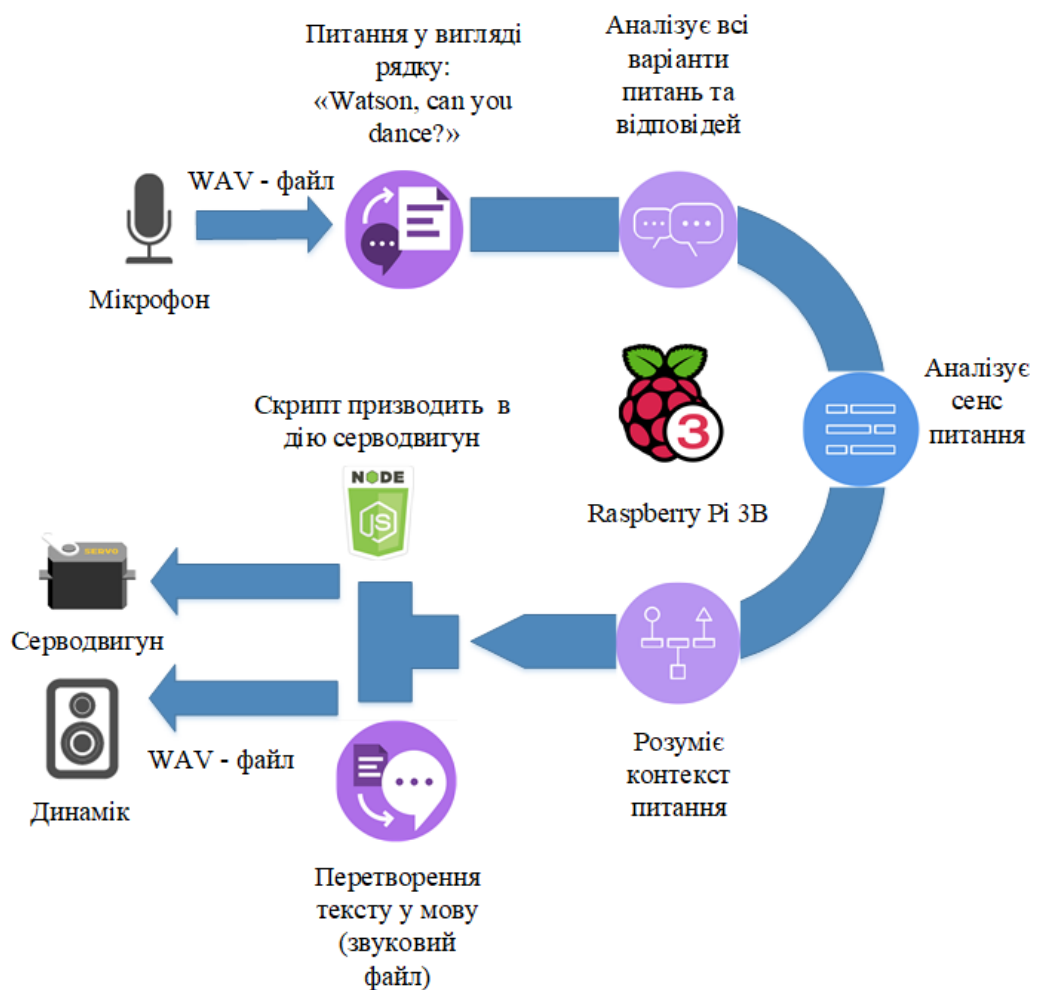


Рис. 3.6 – Структурна схема виконання голосових команд

### 3.2 Налаштування програмного забезпечення для розпізнання елементів зображення

Наступним етапом нашої роботи є налаштування системи, котра може розпізнавати візуальні образи. Нами було реалізовано систему, котра в режимі реально часу може розпізнавати обличчя та очі людини, і у подальшому здійснювати їх аналіз. Схематично даний процес показано на рис. 3.7.

Камеру RaspіCam можна використати як IP-камеру, тобто реалізувати відеопотік і переглядати зображення через веб-браузер як з комп'ютеру так і з телефонів та планшетів Android, iPad та iPhone. Для цього використовують потокові протоколи. Найпопулярніші з них:

- HLS – обраний компанією Apple і має широку підтримку на мобільних пристроях Apple але не на інших;
- Fragmented MP4 – підтримується Adobe та Microsoft, але вимагає плагіни до браузера від цих компаній для відтворення на комп'ютері, тому комп'ютери з ОС Windows і Mac можуть це зробити, а Linux і мобільні пристрої – ні;

IP камери використовують потоковий протокол Motion JPEG (MJPEG). Motion JPEG – це потік окремих JPEG фотографій, одна за одною. Більшість сучасних браузерів можуть програвати MJPEG потоки. Зворотною стороною MJPEG потоків є те, що вони не настільки ефективні, як H.264, який значно підвищує якість і зменшує розмір шляхом кодування тільки відмінності одного кадру від іншого. З MJPEG кожен кадр кодується для всього зображення JPEG.

Краще використовувати MJPG-стрімер з вільним MJPG потоковим сервером, він зручний для запуску в будь-якому місці потокового серверу для камери RaspіCam.

Для початку необхідно встановити бібліотеки, які використовує MJPG-стрімер командою `sudo apt-get install libjpeg62-dev` (рис. 3.8).

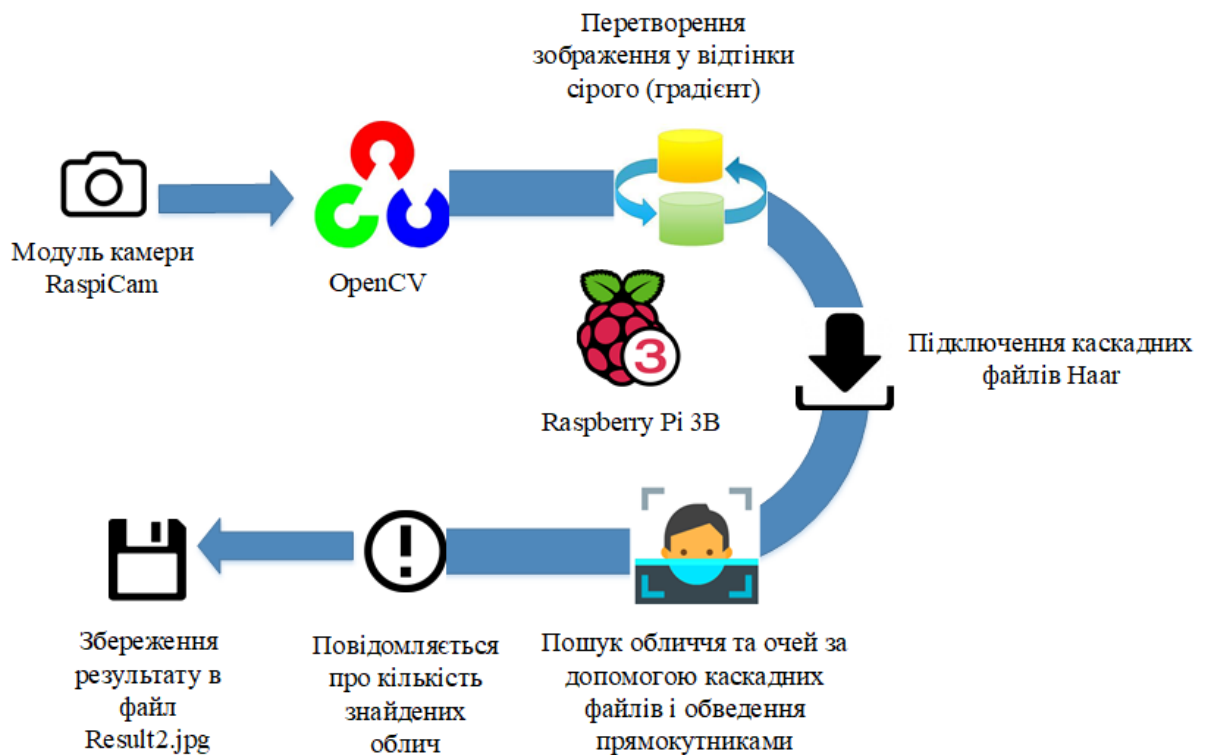


Рис. 3.7 – Структурна схема роботи програмного забезпечення при розпізнаванні обличчя

```
pi@raspberrypiscpp:~$ sudo apt-get install libjpeg-dev
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
libilmbase-dev libjbig-dev liblzma-dev libopencv-calib3d-dev
libopencv-core-dev libopencv-features2d-dev libopencv-flann-dev
libopencv-gpu-dev libopencv-gpu2.4 libopencv-imgproc-dev libopencv-ml-dev
libopencv-ocl2.4 libopencv-photo-dev libopencv-stitching-dev
libopencv-stitching2.4 libopencv-superres2.4 libopencv-ts-dev libopencv-ts2.4
libopencv-video-dev libopencv-videostab2.4 libopencv2.4-java libopencv2.4-jni
libopenexr-dev libtiffxx5 opencv-data
```

Рисунок 3.8 – Установка бібліотек для коректної роботи MJPG

Далі потрібно інстальювати утиліту CMake. CMake – кроссплатформенна автоматизована система збирання проєктів. Далі необхідно завантажити mjpg-streamer з плагіном raspicam з сайту GitHub, переходимо у директорію cd/mjpg-streamer/mjpg-streamer-experimental та компілюємо вихідні файли за допомогою команди make clean all.

Провівши ці етапи налаштування системи ми можемо розпочати трансляцію відеопотоку з камери. Для цього необхідно ввести наступну



команду: `LD_LIBRARY_PATH=/opt/mjpg-streamer/ /opt/mjpg-streamer/mjpg_streamer -i "input_raspicam.so -fps 15 -q 50 -x 640 -y 480" -o "output_http.so -p 9000 -w /opt/mjpg-streamer/www" &` (рис. 3.9 а), де параметри:

- `LD_LIBRARY_PATH` – задає шлях до динамічних бібліотек в поточній директорії. Це необхідно для того, щоб програма мала змогу знайти всі необхідні плагіни для роботи;

- `i` встановлює вхідний плагін. Плагін, який ми використовуємо, називається `input_file.so`. Цей плагін переглядає директорію `i` в будь-який час виявляє записаний в неї файл JPEG, щоб передати цей файл в потік. Папка для перегляду задається як аргумент `-f`;

- `o` встановлює вихідний плагін. Ми використовуємо потоковий плагін HTTP, який запускає веб-сервер, до якого можемо підключитися, щоб подивитися відео. Кореневої каталог веб задається як аргумент `-w`. Ми будемо використовувати веб-сторінки за замовчуванням, які приходять відразу з додатком, але вони можуть бути змінені і налаштовані в міру необхідності.

- `fps` встановлює кількість кадрів в секунду,

- `q` встановлює роздільну здатність зображення.

Нами було обрано параметри 15 кадрів за секунду та роздільну здатність 640 на 480 для того, щоб зменшити навантаження на Raspberry Pi 3B.

Запустивши трансляцію, запускаємо браузер та переходимо за наступною адресою: 10.12.20.247:9000 (рис. 3.9 б)

Щоб мінімізувати введення команд кожний раз для запуску трансляції, створив 2 bash-скрипти з розширенням `.sh` (для Windows – дистрибутивів - `.bat`) для старту (рис. 3.10 а) та завершення (рис. 3.10 б) роботи трансляції.

Bash – це командний інтерпретатор, працюючи, зчитує команди з текстового файлу, який називається скриптом.

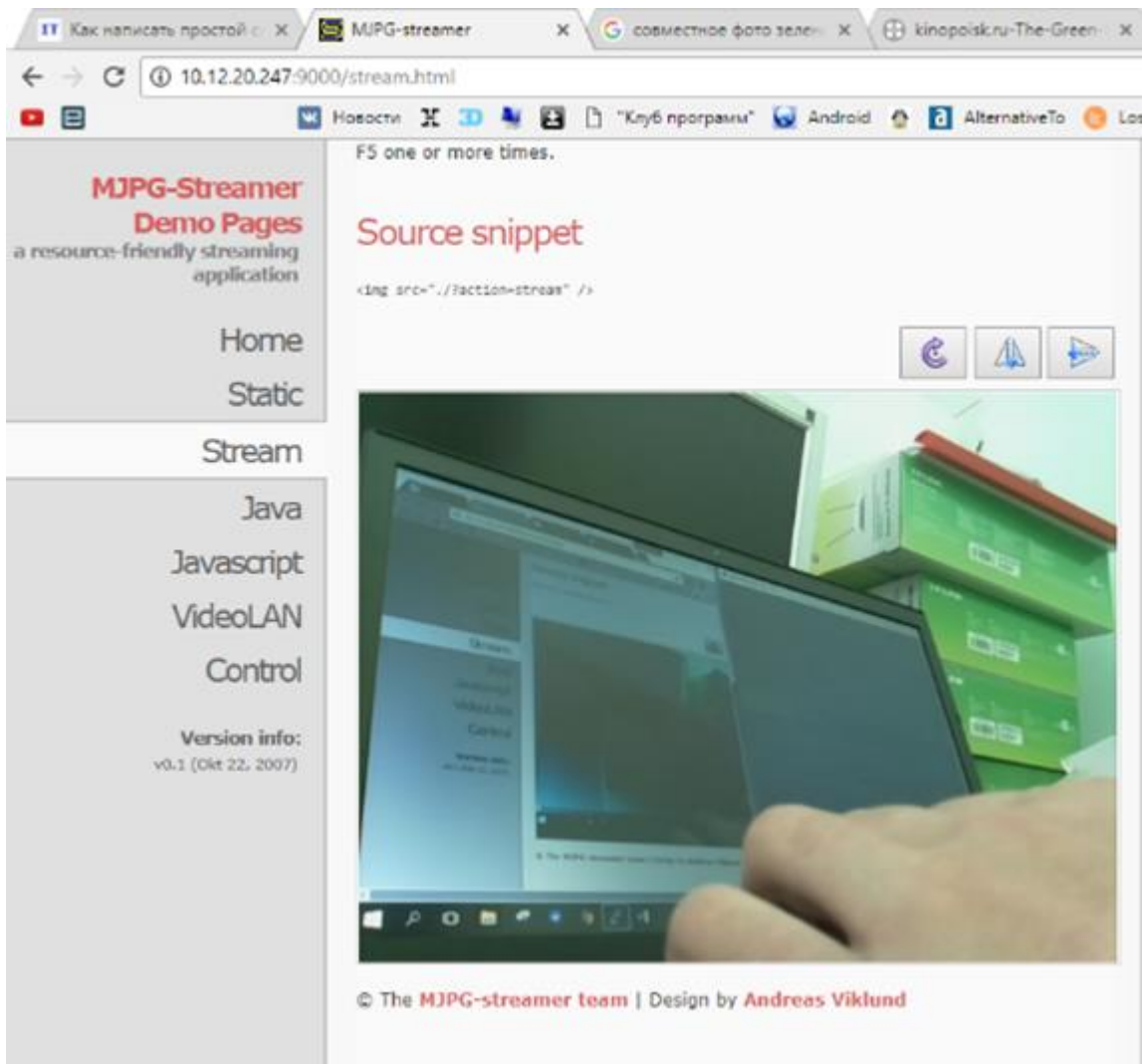
```

pi@raspberrypiscpp:~/mjpg-streamer/mjpg-streamer-experimental $ MJP
G Streamer Version.: 2.0
i: fps.....: 15
i: resolution.....: 640 x 480
i: camera parameters.....:

Sharpness 0, Contrast 0, Brightness 50
Saturation 0, ISO 0, Video Stabilisation No, Exposure compensation 0
Exposure Mode 'auto', AWB Mode 'auto', Image Effect 'none'
Metering Mode 'average', Colour Effect Enabled No with U = 128, V = 128
Rotation 0, hflip No, vflip No
ROI x 0.000000, y 0.000000, w 1.000000 h 1.000000
o: www-folder-path.....: /opt/mjpg-streamer/www/
o: HTTP TCP port.....: 9000
o: HTTP Listen Address..: (null)
o: username:password....: disabled
o: commands.....: enabled
i: Starting Camera
Encoder Buffer Size 81920

```

a



б

Рис. 3.9 – Запуск трансляції відеопотоку (а) та веб трансляції (б)

<pre>#!/bin/bash if pgrep mjpg_streamer &gt; /dev/null then echo "mjpg_streamer already running" else LD_LIBRARY_PATH=/opt/mjpg-streamer/ /opt/mjpg-streamer/mjpg_streamer -i "input_raspicam.so -fps 15 -q 50 -x 640 -y 480" -o "output_http.so -p 9000 -w /opt/mjpg-streamer/www" &gt; /dev/null 2&gt;&amp;1&amp; echo "mjpg_streamer started" fi</pre>	<pre>#!/bin/bash if pgrep mjpg_streamer then kill \$(pgrep mjpg_streamer) &gt; /dev/null 2&gt;&amp;1 echo "mjpg_streamer stopped" else echo "mjpg_streamer not running" fi</pre>
а	б

Рис. 3.10 – Скрипт для запуску (а) та завершення (б) трансляції

Для розпізнавання обличчя за допомогою камери Raspicam нами були використані мова програмування Python, програмне забезпечення OpenCV та додаткові спеціалізовані бібліотеки.

OpenCV (Open Computer Vision) – бібліотека комп'ютерного зору з відкритим вихідним кодом, що надає набір типів даних і чисельних алгоритмів для обробки зображень алгоритмами комп'ютерного зору. Ця бібліотека дуже популярна за рахунок своєї відкритості і можливості безкоштовно і вільно використовувати як в учбових, так і комерційних цілях [16].

Python це – об'єктно-орієнтована мова програмування загального призначення, націлена в першу чергу на підвищення продуктивності самого програміста, ніж коду, який він пише. За рахунок простоти коду, подальший супровід програм, написаних на Python, стає легше і приємніше в порівнянні з Java або C ++.

Виявлення обличчя та очей реалізовано за допомогою каскадів Хаара (рис. 3.11). Ознаки Хаара – ознаки цифрового зображення, використовувані в розпізнаванні образів. Вони використовувалися в першому детекторі обличчя, що працює в реальному часі.

Щоб налаштувати розпізнавання обличчя, необхідно встановити встановити метапакет build-essential – це пакет, в я кому знаходяться посилення на всі необхідні пакети для роботи системи і програм. Далі необхідно інстальювати утиліту CMake та необхідні бібліотеки [8].

В операційній системі Raspbian за замовчуванням вже інстальована мова програмування Python, тому потрібно встановити додаткові бібліотеки для

Python і NumPy. NumPy – це бібліотека мови Python, яка додає підтримку великих масивів і матриць, разом із великою бібліотекою високорівневих і дуже швидких математичних функцій для різноманітних операцій. Встановити необхідні бібліотеки за допомогою команди: `sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev`.

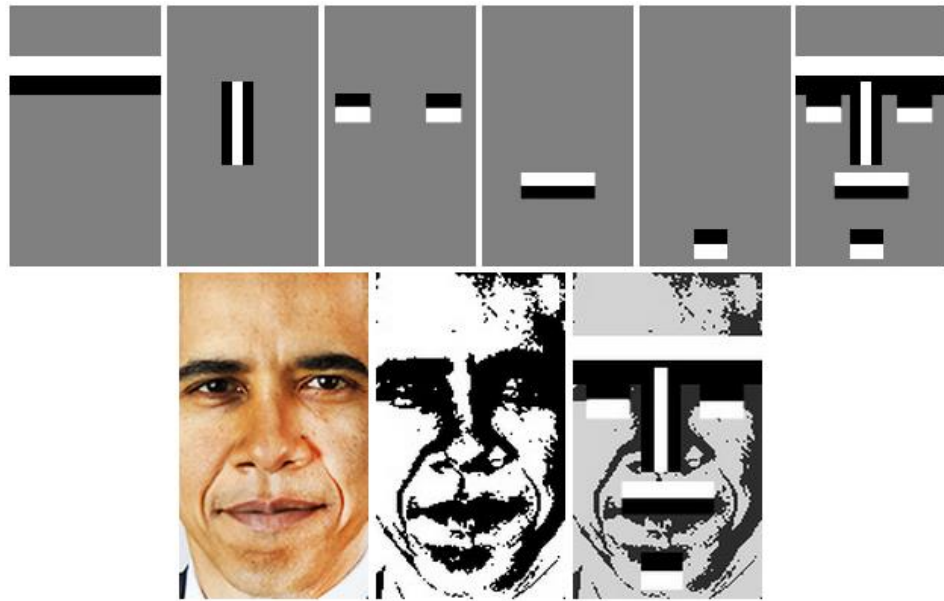


Рис. 3.11 – Виявлення обличчя за допомогою методу каскадів Хаара

Далі потрібно встановити бібліотеки, необхідні для сумісної роботи Python та OpenCV командою `sudo apt-get install python-opencv`. Остання бібліотека `python-matplotlib` – бібліотека двовимірної графіки для мови Python, за допомогою якої можна створювати якісні двовимірні графіки та високоякісні рисунки різних форматів. Matplotlib є модуль-пакетом для Python, встановити можна командою `sudo apt-get install python-matplotlib`.

Далі перевіряємо, чи включений інтерфейс камери. Для цього вводимо команду `sudo raspi-config`, в меню обираємо `Interfacing Options` – `Camera` – майстер задає питання, чи бажаємо ми увімкнути камеру – обираємо `Yes` і перезавантажуємо систему (рис. 3.12).

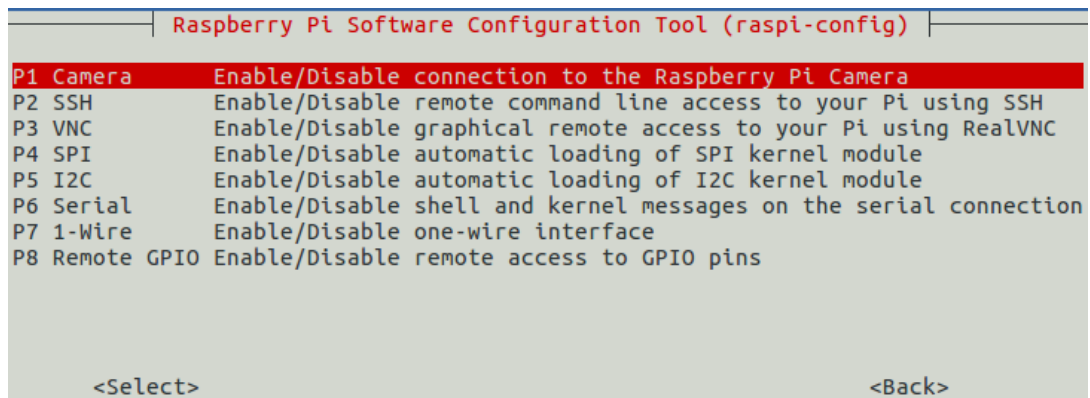


Рис. 3.12 – Налаштування камери Raspicam

```

import io          #Потоки введення-виведення
import picamera   #Задіюється модуль камери Raspicam
import cv2        #Імпортується бібліотека CV(Комп'ютерний зір)
import numpy      #Імпортується бібліотека NumPy
#Створення потоку даних таким чином, щоб фото не зберігалось у файл (збереження потім)
stream = io.BytesIO()
#Отримуємо зображення (Встановлено низьку роздільну здатність, щоб швидкість фотографування була швидша)
#Тут можна встановити необхідні параметри (наприклад, повернути зображення і т.д.)
with picamera.PiCamera() as camera:
    camera.resolution = (320, 240)
    camera.capture(stream, format='jpeg')
    #Конвертація зображення в масив numpy
    buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8)
    #Створює OpenCV зображення
    img = cv2.imdecode(buff, 1)
    #Тепер завантажується спеціальний "каскадний" файл для виявлення обличчя та очей
    face_cascade = cv2.CascadeClassifier('/usr/share/opencv/haarcascades/haarcascade_frontalface_alt.xml')
    eye_cascade = cv2.CascadeClassifier('/usr/share/opencv/haarcascades/haarcascade_eye.xml')
    #Тепер перетворює зображення у відтінки сірого
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    #Перегляд обличчя на зображенні за допомогою завантаженого каскадного файлу
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    print "Found "+str(len(faces))+ " face(s) "
    #Малюється прямокутник навколо кожного обличчя та очей
    for (x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

        eyes = eye_cascade.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color, (ex,ey), (ex+ew,ey+eh), (0,255,0), 2)

#Зберігається результат з назвою result2
cv2.imwrite('result2.jpg', img)

```

Рис. 3.13 – Вміст файлу facedetection1.py для пошуку обличчя та очей

Тепер необхідно протестувати камеру, для цього в терміналі вводимо команду `sudo raspistill -o filename.jpg`, фото збережеться в поточній директорії.

Далі створюємо файл з кодом Python (рис 3.13), зберігаємо з

роширенням `.py`, і запускаємо командою `sudo python facedetection1`, Алгоритм роботи програми наступний: камера робить знімок, виконує пошук обличчя, повідомляє користувача про результат та зберігає рисунок з розпізнаним обличчям. Результат зберігається у папці Face з іменем `result2.jpg`.

У результаті тестування програмного забезпечення було проведено пошук у потоковому відео обличчя (рис. 3.14 а), потім обличчя і очей (рис. 3.14 б) та декількох облич (рис. 3.14 в).

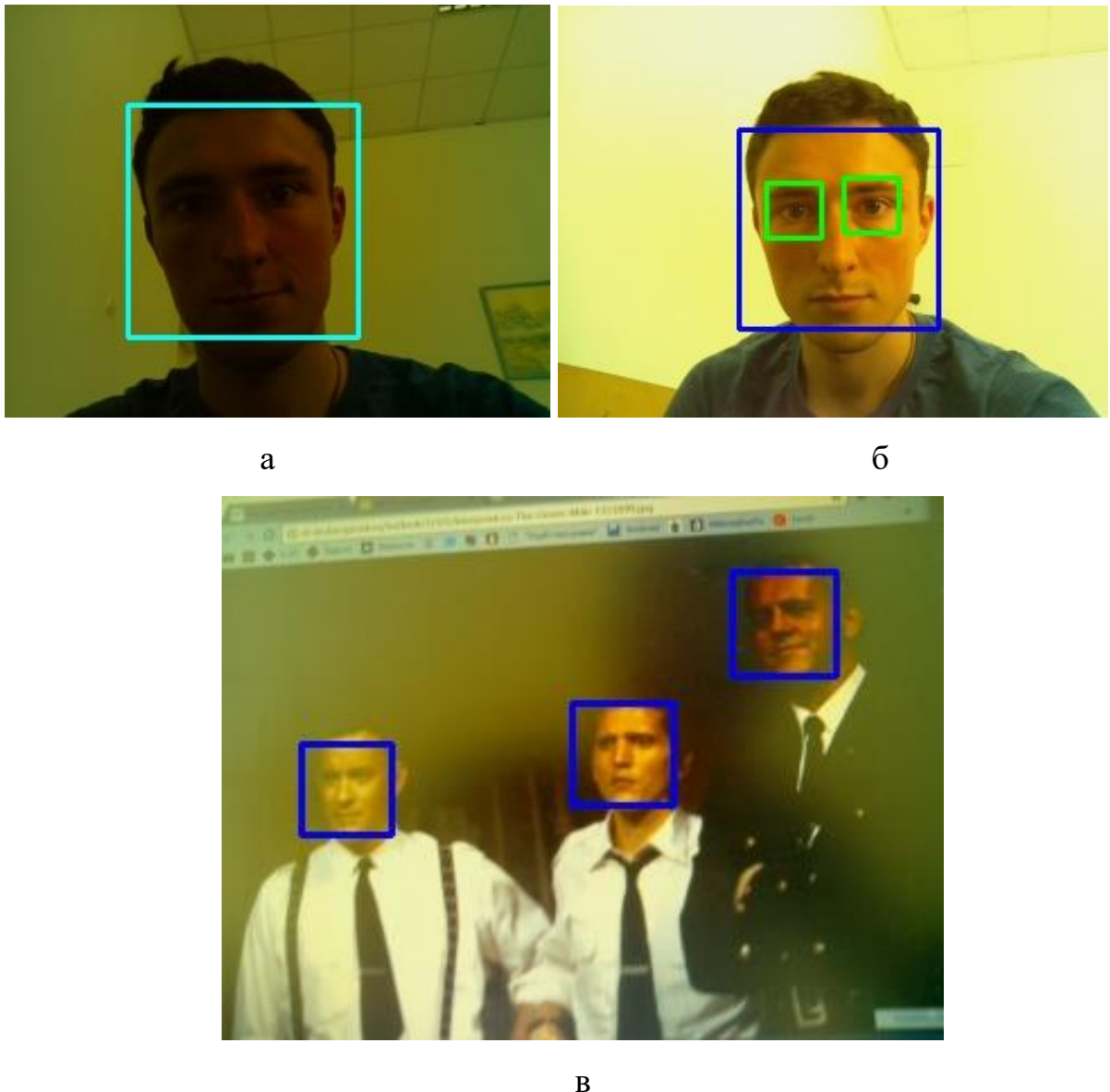


Рис. 3.14 – Результат пошуку обличчя (а), обличчя та очей (б) та декількох облич (в)

## ВИСНОВКИ

1. Проведено аналіз літературних джерел стосовно використання елементів штучного інтелекту в програмному забезпеченні електронних пристроїв обробки і аналізу даних. Розглянуто історію створення ШІ, принципи роботи, обмеження та напрямки у дослідження в області штучного інтелекту

2. Показано, що для проектування і розробки електронного пристрою аналізу даних, котрий демонструє ознаки штучного інтелекту можуть бути використані одноплатні мікропроцесорні платформи, такі, як Raspberry Pi та NVIDIA Jetson Nano.

3. Показано, що для реалізації можливості аналізу «живої» мови та потокового відео добре підходить платформа Raspberry Pi, оскільки її потужності достатньо для реалізації таких задач. У той же час, використання платформи NVIDIA Jetson Nano економічно не виправдане, оскільки вона має більшу вартість, а її розрахункова потужність буде використовуватись не повною мірою.

4. Виконано підбір периферійного обладнання для Raspberry Pi для електронної системи аналізу даних за ознакою ціна/якість. Розроблено принципову схему системи аналізу даних з елементами штучного інтелекту з використанням апаратної платформи Raspberry Pi.

5. Розроблено програмне забезпечення для реалізації двох функціональних можливостей із самонавчанням, які демонструють елементи штучного інтелекту, а саме розпізнавання голосу та розпізнавання візуальних елементів у потоковому відео. Виконано налаштування та протестовано системи аналізу мови і розпізнавання обличчя в режимі реального часу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Робототехніка в Україні: розробки і перспективи: [Електронний ресурс] – Режим доступу: <https://dou.ua/lenta/articles/robotics-in-ukraine/https://geekelectronics.org/raspberry-pi/raspberry-pi-pervyj-zapusk.html>
2. Види штучного інтелекту: [Електронний ресурс] – Режим доступу: <https://uk.education-wiki.com/2698347-types-of-artificial-intelligence>
3. Верескун М.В. Навчання машин та штучний інтелект / М.В. Верескун – Маріуполь: ДВНЗ «ПДТУ», 2019. – 84 с.
4. Raspberry Pi — перший запуск: [Електронний ресурс] – Режим доступу: <https://geekelectronics.org/raspberry-pi/raspberry-pi-pervyj-zapusk.html>
5. Машина Тюрінга: тест і теорія: [Електронний ресурс] – Режим доступу: [https://studopedia.com.ua/1\\_140872\\_mashina-tyuringa.html](https://studopedia.com.ua/1_140872_mashina-tyuringa.html)
6. Використання портів GPIO в Raspberry Pi. Частина 1: [Електронний ресурс] – Режим доступу: <http://eah.me/raspberry-pi-gpio/>
7. Троцько В.В. Методи штучного інтелекту: навчально-методичний і практичний посібник / В.В. Троцько. - К.: Університет "КРОК", 2020. – 86 с.
8. Srihari K., Ramesh R., Udayakumar E., Dhiman G. An Innovative Approach for Face Recognition Using Raspberry Pi. Artificial Intelligence Evolution. – 2020. – V. 1(2). – P.63 – 144.
9. Imran Uddin Md., Shahriar Alamgir Md., Mahabubur Rahman Md., Bhuiyan M. AI Traffic Control System Based on Deepstream and IoT Using NVIDIA Jetson Nano. – 2021. – P. 115 – 119.
10. Vijitkunsawat W., Chantngarm P. Comparison of Machine Learning Algorithm's on Self-Driving Car Navigation using Nvidia Jetson Nano. – 2020. – P.201 – 204.
11. Chen L., Chen P., Lin Z. Artificial Intelligence in Education: A Review. – 2020. P.75264 – 75278.
12. Kumar Jain A. Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino. – 2018. – P.1630 – 1635.



13. Tae-Kook K. Short Research on Voice Control System Based on Artificial Intelligence Assistant. – 2020. – P.236 – 242.
14. Varghese L., Deepak G. IoT Analytics Approach for Weather Forecasting using Raspberry Pi 3 Model B+. – 2019. – P. 415 – 419.
15. Muck Y., Homam P. «Iot Based Weather Station Using Raspberry Pi 3», International Journal of Engineering & Technology, – 2018. – 15 p.
16. Dewangan D., Sahu S.. Deep Learning Based Speed Bump Detection Model for Intelligent Vehicle System using Raspberry Pi. – 2020. – 79 p.
17. Mohan H. M., Anitha S., Chai R.. Edge Artificial Intelligence:Real-Time Noninvasive Techniquefor Vital Signs of Myocardial Infarction Recognition Using Jetson Nano. – 2021. – 119 p.
18. Zhan C., Lu Y. Study on artificial intelligence: The state of the art and future prospects // IEEMA Engineer Infinite Conference – 2021. – DOI: 10.1109/ETECHNXT.2021.8495294.
19. Kurniawan A. Introduction to NVIDIA Jetson Nano. In IoT Projects with NVIDIA Jetson Nano. – 2021. – 142 p.
20. Mazzia V., Salvetti F. Real-Time Apple Detection System Using Embedded Systems With Hardware Accelerators: An Edge AI Application. 2020. – P. 9102 – 9114.
21. Сімагін В.В., Тищенко К.В. Реалізація мікроконтролерних систем аналізу даних з елементами штучного інтелекту / Матеріали Міжнародної науково-технічної конференції студентів та молодих вчених «Фізика, електроніка, електротехніка ФЕЕ-2022». – Суми: СумДУ, 2022. – С.53.