

ЧИГІНЬ ВАСИЛЬ

Національний університет «Львівська політехніка»

<https://orcid.org/0000-0003-1593-6832>e-mail: vchygin@gmail.com

ПАЗИНЮК МИХАЙЛО

Національний університет «Львівська політехніка»

e-mail: pazyniuk.m@gmail.com

АВТОНОМНА ЕКСПЕРИМЕНТАЛЬНА СИСТЕМА І ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗВУКОВОГО ВИЯВЛЕННЯ ОБ'ЄКТІВ ТА ON-LINE СПОВІЩЕННЯ

Створили експериментальну автономну систему і програмне забезпечення для звукового виявлення об'єктів з надсиланням on-line сповіщення при їх виявленні. Використали міні-комп'ютер ASUS TS-10 і мікрофон типу SyncoMic-M3. Програмне забезпечення для аналізу звуку в реальному часі сформували на основі власної програми, написаної мовою C++ з використанням застосунку відкритого вихідного коду FrequencyAnalyzer. Використали розклад звукової хвилі у ряд Фур'є за алгоритмом швидкого перетворення (FFT). При цьому вперше ввели такі функції, як граничне значення сумарної інтенсивності звуку, при якому надсилається сповіщення, та значення тривалості сигналу для відсікання випадкових звуків. Склали програму мовою Python для надсилання сповіщення при виявленні об'єкта з використанням бібліотеки Flask для отримання запитів від програми аналізу звуку. При цьому використали бібліотеку python-telegram-bot для автоматизації надсилання сповіщень у телеграм-чат. Python-застосунок розгорнули у хмарному середовищі Heroku, що дозволяє зручно керувати застосунком, його версією та запитами на відправлення повідомлення. В основі програмного забезпечення для аналізу звуку лежить застосування алгоритму швидкого перетворення Фур'є (FFT). Цей алгоритм діє як математичний наріжний камінь, ефективно перетворюючи складні звукові хвилі на зрозумілі представлення рядів Фур'є. Це перетворення дозволяє глибше зрозуміти вхідне аудіо, дозволяючи системі розпізнавати ключові моделі та аномалії. Створена експериментальна система та програмне забезпечення є свідченням інновацій у виявленні об'єктів за допомогою звукового аналізу. Використовуючи необхідне апаратне забезпечення, ретельне програмування та інтеграцію, система забезпечує надійне та ефективне рішення. Паралельно з програмним забезпеченням для аналізу звуку було розроблено додаткову програму Python для миттєвого сповіщення про виявлення об'єкта. Використовуючи бібліотеку Flask, ця програма організовує безперебійний зв'язок між програмним забезпеченням аналізу звуку та зовнішніми системами. Ця інтеграція гарантує, що події виявлення викликають негайне сповіщення, уможлижуючи реагування у реальному часі.

Ключові слова: експериментальна установка, програмне забезпечення, звукове виявлення об'єкта, Flask бібліотека, хмарне середовище Heroku, Python-застосунок.

CHYHIN VASYL, PAZYNIUK MYHAILO

Lviv Polytechnic National University

AUTONOMOUS EXPERIMENTAL SYSTEM AND SOFTWARE FOR SOUND OBJECT DETECTION AND NOTIFICATION

An experimental system and software have been created to detect the object with sending the notification about its detection. ASUS TS-10 Mini PC and Synco Mic-M3 type microphone were used. A real-time sound analysis software was created using a C++ programming language with a usage of open-source application named 'Frequency Analyzer' was developed. Fast Fourier Transform algorithm was used to expand a sound wave to Fourier series. Herewith such functions as entering threshold sound intensity value, with a notification sending, and signal duration value to cut off unnecessary sounds were introduced. A python-application was created to send notification when the object is detected using Flask library to receive the requests from Sound Analysis application. Herewith a python-telegram-bot library was used to automate sending messages to the telegram chat. Python-application was deployed on the Heroku cloud environment, which allows to comfortably manage the application, its version, and requests to send the message. At the heart of the sound analysis software lies the ingenious application of the Fast Fourier Transform (FFT) algorithm. This algorithm acts as a mathematical cornerstone, effectively converting intricate sound waves into comprehensible Fourier series representations. This transformation allows for a deeper understanding of the audio input, enabling the system to discern key patterns and anomalies. The created experimental system and software stand as a testament to innovation in object detection through sound analysis. By leveraging cutting-edge hardware, meticulous programming, and strategic integrations, the system provides a robust and efficient solution. In parallel to the sound analysis software, a complementary Python application was ingeniously devised to facilitate instant notifications upon object detection. Leveraging the Flask library, this application orchestrates seamless communication between the sound analysis software and external systems. This integration ensures that detection events trigger immediate notifications, enabling real-time awareness and response.

Keywords: computer control model, software, object sound detection, Flask library, Heroku cloud environment, Python application.

Вступ

На даний час особливо актуальною є задача звукового виявлення об'єктів та формування on-line сповіщень при цьому. Зокрема, створення такої системи є необхідним для попередження про появу несанкціонованих літальних чи наземних апаратів різного типу.

Аналіз останніх досліджень і публікацій

В українській літературі та інтернет сайтах не виявлено описання систем і програмних забезпечень,

які дозволяють виявляти об'єкти за допомогою їх звукових характеристик та надсилати сповіщення. Аналіз іноземних публікацій свідчить про те, що на сьогоднішній день приділяється особлива увага до автоматизованих систем виявлення та розпізнавання об'єктів [1, 2]. В таких системах застосовуються різні фізичні принципи виявлення – електромагнітні, акустичні, оптичні і т.п. Основним недоліком активних систем виявлення та розпізнавання об'єктів є їх вразливість до зовнішнього виявлення. В той же час, пасивні системи можуть приховано і тривало працювати в режимі реального часу.

Окремі дослідження проводилися в області аналізу звуку, зокрема, при створенні застосунку FrequencyAnalyzer для швидкого розкладу звукової хвилі у ряд Фур'є [3]. Цей застосунок потребує подальшої модернізації для можливості виявлення об'єктів в реальному часі. Перевагою застосунку FrequencyAnalyzer є відкритий вихідний код алгоритму швидкого перетворення Фур'є і можливість введення оптимальних параметрів опрацювання звуку об'єктів. При аналізі публікації, зв'язаних з цим алгоритмом, виявлено, що існує декілька методів його програмної реалізації. Зокрема, вони досліджуються в роботі [4], де описані методи їх програмної реалізації (Cooley-Tukey algorithm, Prime-Factor algorithm, Rader's FFT algorithm). При складанні програми звукового аналізу і виявлення об'єктів обрали метод Кулі-Тьюкі (Cooley-Tukey) [5] через його легкість в реалізації і відносно невеликій швидкості обчислення. Зокрема, такий алгоритм дозволяє виразити дискретне перетворення Фур'є довжини, рівній довільному складеному числу N , через певну кількість перетворень меншої довжини за допомогою рекурсії, понижуючи таким чином швидкість обчислення до $O(N \log N)$.

У публікації [6] описуються та порівнюються популярні на даний момент бібліотеки (фреймворки) для мови Python, які призначені для роботи з веб-сервісами. Зокрема, вони поділяються на три типи: фул-стек, мікросервісні та асинхронні. Кожен з типів має свої переваги та недоліки та свою сферу застосування. Оскільки мета фул-стек програм – створення повноцінних великих за обсягом додатків, а мета асинхронних – швидке опрацювання великої кількості інформації, обрали мікросервісний тип. Метою мікросервісного типу є створення невеликих застосунків, які можуть швидко опрацьовувати певну кількість інформації для відповідних потреб. Серед наведених в публікації мікросервісних фреймворків (Bottle, CherryPy, Dash, Flask, MorePath тощо), слід відзначити такі переваги бібліотеки Flask, як гнучкість, швидкодія, легкість у використанні. Крім того, для створення веб-застосунку він дозволяє використати лише один файл, в якому написана вся логіка згідно з документацією Flask [7]. Враховуючи ці переваги фреймворку Flask, обрали його в роботі як основу для створення застосунку.

У публікації [8] порівнюють сервіси, які дозволяють розгорнути веб-застосунки на одному з них для безперешкодного виконання запитів на застосунок. Зокрема, порівнюється сервіс Heroku та платформа PythonAnywhere. Проаналізувавши переваги та недоліки кожного з сервісів, обрали хмарне середовище Heroku. Серед переваг PythonAnywhere можна відзначити легке розгортання Python-застосунку, оскільки цей сервіс працює лише з застосунками, що написані мовою Python. Проте, недоліками PythonAnywhere є відсутність багатьох додаткових можливостей, що полегшують роботу із застосунками, які наявні в Heroku. Серед переваг Heroku можна відзначити легкість у налаштуванні та користуванні сервером, адже весь процес розгортання веб-застосунку здійснюється в декілька кліків без необхідності написання коду. При цьому процеси керування додатком, його модифікацією і версіями є швидкими і простими. Аналіз публікації [9] показав, що процес встановлення python-додатку на хмарне середовище Heroku є простим та інтуїтивно зрозумілим. Для цього слід створити новий проект на власній сторінці сайту Heroku, обравши ім'я проекту та сайту, на якому буде розташовуватись застосунок. Після створення проекту, необхідно розгорнути Flask-застосунок на новоствореному проекті. Завдяки легкості Flask-застосунку для повноцінного функціонування веб-додатку достатньо розгорнути лише один файл, в якому написана вся логіка веб-додатку. Тому при складанні програми передавання сповіщень використали хмарне середовище Heroku.

У публікації [10] порівнюються бібліотеки для мови Python, які дозволяють робити інтеграцію з сервісами Telegram-месенджера. Серед бібліотек, що порівнювались (python-telegram-bot, pyTelegramBotAPI, Telethon, aiogram), звернули увагу на python-telegram-bot. До переваг цієї бібліотеки слід віднести легкість у налаштуванні і надсиланні сповіщень, швидкодію роботи та простоту документації, що дозволяє створювати застосунки різної складності з використанням Telegram-месенджера. Саме тому у роботі для надсилання сповіщень обрали python-telegram-bot.

Метою роботи є створення експериментальної системи і програмного забезпечення для виявлення об'єктів за їх звуковими сигналами з можливістю надсилання сповіщень про його наявність і мінімізація її розмірів.

Методика дослідження та експериментальна система

Основою для створення експериментальної системи і програмного забезпечення дослідження процесів виявлення об'єкта за його звуковими характеристиками з надсиланням сповіщення служили звуковловлювач з кардіоїдним мікрофоном і концентратором параболоїдного типу, міні-комп'ютер типу ASUSTS-10, а також відкритий код FrequencyAnalyzer [4]. Генератором звуку гармонічного типу служив безпілотний літальний апарат (БПЛА) типу Фантом 3. Відкритий вихідний код FrequencyAnalyzer удосконалили з метою порівняння сумарної інтенсивності звуків віддалених об'єктів з наперед заданим рівнем звукового сигналу порядку фоновому у певному діапазоні частот і надсилання сповіщення у

Telegram-месенджер при її перевищенні протягом наперед заданого часу.

Фотографія на рисунку 1 відображає основні елементи запропонованої експериментальної системи: 1 – комп'ютер ASUS TS-10, 2 – мікрофон SyncoMic-M3, 3 – розгалужувач для навушників і мікрофона, 4 – вихід HDMI для під'єднання до монітора і відлагодження системи, 5 – вхід живлення Micro-USB, 6 – 2 USB входи для мишки і клавіатури. Завдяки мінімальній кількості основних елементів системи досягається мінімізація розмірів експериментальної установки в цілому.

Комп'ютер ASUS TS-10 – це сучасний персональний комп'ютер, що працює з операційною системою Windows 10. Завдяки його характеристикам, а саме 4-ядерному процесору Intel, що працює на частоті 1.44 Гц, має 2 Гб оперативної пам'яті та 32 Гбайт постійної пам'яті, він має значну перевагу над іншими комп'ютерами такого типу. У порівнянні з комп'ютером Raspberry PI, ASUSTS-10 є більш гнучким у налаштуванні і користуванні завдяки операційній системі, дозволяє запускати програми, написані різними мовами програмування, проводити швидкі обчислення в реальному часі завдяки високій швидкодії та потужності, не потребує додаткових вкладень у систему у вигляді допоміжних модулів. Комп'ютер ASUS TS-10 має USB 2.0 та USB 3.0 порти, HDMI порт, 3.5 мм лінійний вхід для навушників та Micro-USB порт для живлення. Він працює від блоку живлення із потужністю 18 Ватт. Завдяки своїй енергоефективності, ASUS TS-10 може працювати від зовнішнього пристрою живлення (Powerbank) із напругою 5-9 В та струмом 2А. Незначні розміри цього міні-комп'ютера дозволили мінімізувати розміри кінцевої установки. Його структура дозволила обійтися без додаткової зовнішньої звукової карти.

Оскільки комп'ютер не має вбудованого мікрофона, використали кардіоїдний зовнішній мікрофон типу Synco Mic-M3. Мікрофон такого типу дозволяє якісно зчитувати і опрацьовувати звукові сигнали завдяки своїм характеристикам і функціям підсилення та фільтру низьких частот, що дозволяє також усувати шкідливі сигнали (шум, вітер). Для під'єднання мікрофона до комп'ютера використали розгалужувач 3, що дозволяє під'єднати навушники і мікрофон одночасно.



Рис. 1. Фотографія комплексу «Комп'ютер-мікрофон»: 1 – комп'ютер ASUSTS-10, 2 – мікрофон SyncoMic-M3, 3 – розгалужувач лінійного входу, 4 – вихід HDMI, 5 – вхід живлення Micro-USB, 6 – 2 USB входи

Програмне забезпечення системи звуковловлення і надсилання сповіщення

Склали програму для аналізу звуку мовою програмування C++ і використали комп'ютер ASUS TS-10 для відлагодження програми. Основою програмного забезпечення став проект з відкритим вихідним кодом під назвою “FrequencyAnalyzer” [4]. Ця програма працює за алгоритмом швидкого перетворення Фур'є і дозволяє опрацьовувати та аналізувати звукові характеристики вхідного сигналу у реальному часі. Наявність вихідного коду дозволяє модифікувати програму, розширюючи її функціонал. У програму додали додаткові функції аналізу звукового сигналу та його опрацювання, зокрема, поля для введення граничної тривалості звуку і граничного значення інтенсивності сигналу, при досягненні яких відправляється сповіщення. Для зручності надсилання запитів на веб-сервер встановили бібліотеку, яка опрацьовує надсилання таких запитів. Для цього використали бібліотеку з відкритим вихідним кодом “HttpRequest.hpp”.

Якщо виміряна сумарна інтенсивність сигналу у певному діапазоні частот є більшою, ніж задана у вікні вводу даних, що зображене 7 на рис.2, починається відлік таймера. При виконанні такої умови програма на кожному кроці ітерації 10 перевіряє, чи сумарна інтенсивність є більшою, ніж задана гранична інтенсивність. Якщо така умова виконується, то таймер продовжує виконувати відлік. Якщо значення таймера є більшим, ніж значення, що вказане у вікні вводу даних 10, виконується функція надсилання запиту на Flask веб-сервер. Останній, в свою чергу, опрацьовує запит і надсилає сповіщення в телеграм-месенджер, яке повідомляє користувача про наявність об'єкта.

Користувачський інтерфейс програми складається з трьох вікон (рис. 2):

а) вікно введення даних, до якого додали власні функції

- б) вікно з представленням звуку у вигляді залежності його інтенсивності від частоти,
в) вікно із сумарною інтенсивністю звуку в реальному часі, як критерію для посилення сповіщення.

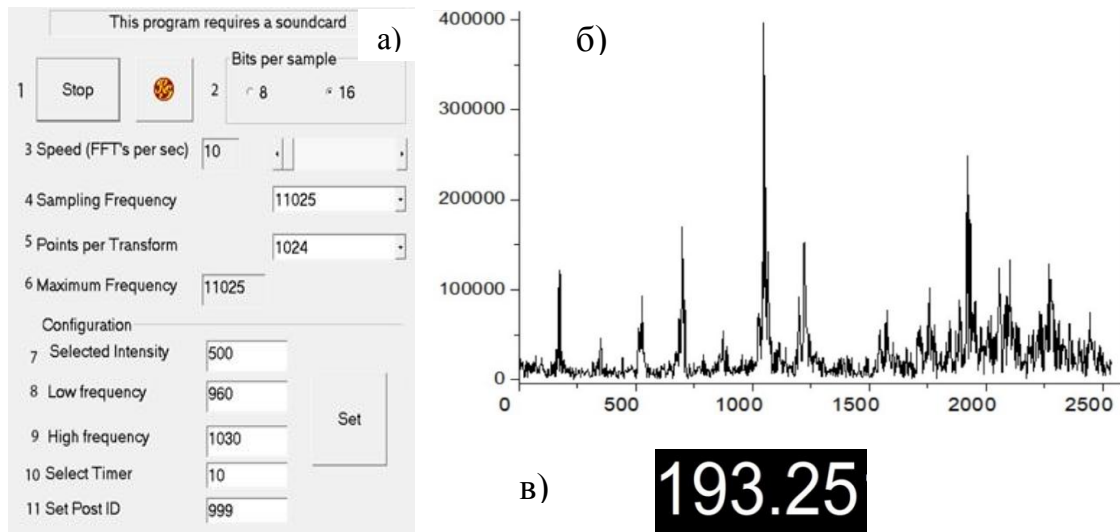


Рис. 2. Користувачський інтерфейс програми:

- а) вікно введення даних, б) вікно з представленням звуку у вигляді залежності його інтенсивності від частоти, в) вікно із сумарною інтенсивністю звуку в реальному часі.

1 – кнопка пуску/припинення роботи програми, 2 – розрядність розкладу звуку у ряд Фур'є, 3 – швидкість алгоритму перетворення Фур'є, 4 – частота дискретизації, 5 – кількість точок на одиничне перетворення звукового ряду, 6 – максимальна частота звуку для опрацювання програмою, 7 – граничне значення інтенсивності сигналу, 8 – мінімальна частота захопленого звуку, 9 – максимальна частота звуку, що приймається мікрофоном, 10 – кількість ітерацій, що задають час тривання сигналу до моменту передавання сповіщення, 11 – номер поста, на якому розташована установка звуковловлення.

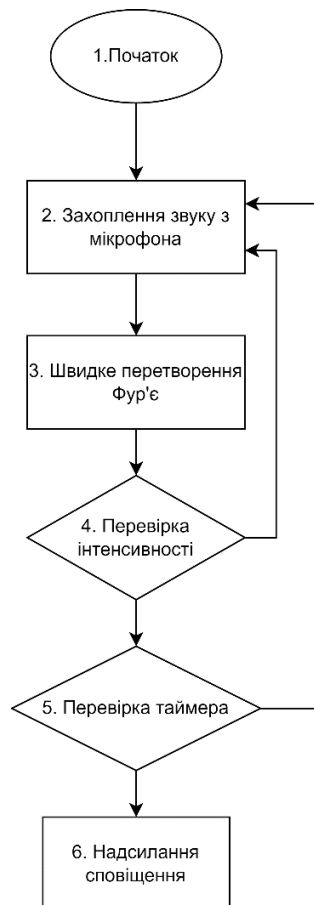


Рис. 3. Алгоритм програми для аналізу звукового сигналу:

- 1 - початок роботи програми, 2 - процес зчитування звуку з мікрофона і підготовка до його опрацювання алгоритмом Швидкого Перетворення Фур'є.

1. Процес швидкого перетворення Фур'є, обчислення суми інтенсивностей в заданому користувачем діапазоні частот.

2. Перевірка суми інтенсивностей, опрацьовані в кроці 3. Якщо таке значення є більшим ніж вказане

користувачем в вікні вводу, перехід до кроку 5.

3. Початок відліку таймера; якщо значення суми інтенсивностей на кожній ітерації залишається більшим, ніж введене користувачем значення у вікні вводу, таймер продовжує свою роботу. Якщо кількість ітерацій є більшою, ніж значення, введене користувачем у вікні вводу, перехід до кроку 6.

4. Надсилання запиту на Flask-веб сервер, запуск python-застосунку, що надсилає сповіщення в telegram-чат.

Алгоритм програми зображено на рис. 3. Для оптимізації програми і надсилання сповіщень, створили застосунок мовою програмування Python. Використовуючи фреймворк під назвою “Flask”, створили веб-сервер, який може отримувати запити від програми для аналізу звуку і в подальшому надсилати сповіщення користувачу. Для надсилання сповіщення обрали месенджер Telegram і бібліотеку для Python програми, що дозволяє надсилати сповіщення в чат. Надсилання запитів на сервер розгорнули на хмарній платформі Heroku, яка дозволяє зберігати сервер, налаштовувати його і користуватися ним. Обрали безкоштовну ліцензію, що покриває потреби експериментальної системи. У момент часу, коли виникає необхідність у сповіщенні користувача про виявлення об’єкта, надсилається запит на відповідну адресу, яка вказується при створенні проекту на Heroku сторінці. При отриманні запиту, Flask-застосунок зчитує параметри, передані на адресу, опрацьовує їх та надсилає сповіщення у телеграм-чат.

Перевагою надсилання сповіщень на месенджер є можливість отримувати сповіщення на будь-якому пристрої у довільний час при наявності якісного сигналу для виходу в інтернет. Також надсилання сповіщень у месенджер дозволяє надсилати їх відповідній групі людей, використовуючи можливість чату або групи у месенджері.

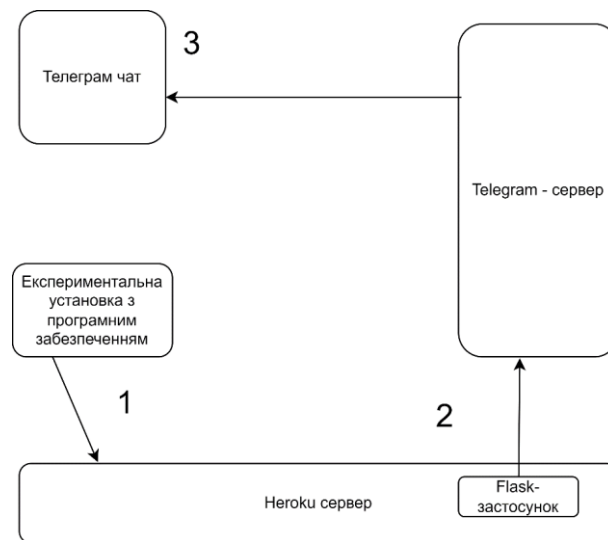


Рис. 4. Схема установки з програмним забезпеченням

1. Надсилання запиту при виявленні об’єкта програмою для аналізу частот на Flask-веб-сервер, що розгорнутий на хмарному середовищі Heroku.
2. Опрацювання Flask-застосунком отриманих параметрів, створення повідомлення і створення запиту на Telegram-сервер про відправку сповіщення.
3. Опрацювання запиту, надісланого Flask-застосунком та надсилання сповіщення в чат.

Результати дослідження та їх аналіз

При дослідженнях запускали програму, яка працює відповідно до алгоритму на рис. 3 і відправляє сповіщення за схемою на рис. 4. Попередньо встановили відповідне значення граничної інтенсивності сигналу та вибрали бажане значення таймера, при якому відсилається сповіщення, щоб уникнути небажаних та випадкових звуків. Підписались на відповідний канал в telegram-месенджері, на якому встановлений користувач-бот, через якого надсилаються сповіщення для кінцевих користувачів.

Як видно з рис. 4, для тестування експериментальної установки не потрібно встановлення ніяких додаткових програм або бібліотек, оскільки вся логіка опрацювання звукового сигналу знаходиться в exe-файлі програми, а надсилання сповіщень – на сервері, що знаходиться у хмарному середовищі.

При забезпеченні якісного під’єднання до мережі Wi-Fi, протестували програму, використовуючи генерацію звуків з різною частотою та різною тривалістю сигналу. З рис. 5 можна побачити, що сповіщення надсилаються успішно, інформуючи користувача про інтенсивність сигналу, яка була у момент надсилання сигналу та час надсилання, що свідчить про коректну роботу програми.

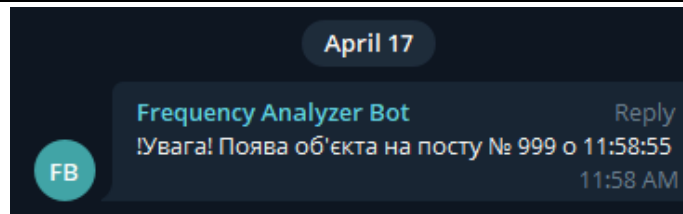


Рис. 5. Скріншот Telegram-чату з надсиланням сповіщення про виявлення об'єкта

Висновки

Підсумовуючи результати експериментального випробування системи і програми виявлення об'єкта з надсиланням сповіщень, можна стверджувати про те, що запропонована система і програмне забезпечення є достатніми для виявлення об'єкта у просторі за звуком, опрацюванням звукової інформації про нього та сповіщенням користувача, надаючи швидко і достатню інформацію. Отримані результати виявлення об'єкта за допомогою програми FrequencyAnalyzer, разом з аналізом та обробкою звукового сигналу та відправленням сповіщення при відповідних значеннях свідчать про можливість продовження роботи у напрямку дослідження звукових характеристик об'єктів, зокрема, при більш детальному аналізі гармонік звукових коливань для встановлення типу об'єкта.

References

1. Vrba M., Saska M. Marker-Less Micro Aerial Vehicle Detection and Localization Using Convolutional Neural Networks. IEEE Robotics and Automation Letters, vol. 5, no. 2. 2020. doi: 10.1109/LRA.2020.2972819.
2. Bouguettaya A., Zarzour H., Kechida A., Taberkit A. Vehicle Detection From UAV Imagery With Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 11. 2022. doi: 10.1109/TNNLS.2021.3080276.
3. Fourier Transform. URL: https://uk.wikipedia.org/wiki/Перетворення_Фур'є.
4. The Real Time Frequency Analyzer. <http://www.icodeguru.com/CPP/CppInAction/freq.html>.
5. Алгоритм Кулі-Тьюкі. https://uk.wikipedia.org/wiki/Алгоритм_Кулі_Тьюкі.
6. Top Python Frameworks. <https://www.monocubed.com/blog/top-python-frameworks/>.
7. Flask Documentation. <https://flask.palletsprojects.com/en/2.0.x/>.
8. Heroku vs PythonAnywhere. <https://stackshare.io/stackups/heroku-vs-pythonanywhere>.
9. How to Deploy Python App on Heroku. <https://www.geeksforgeeks.org/deploy-python-flask-app-on-heroku/>.
10. Python Telegram Bot Usage. <https://kandi.openweaver.com/collections/python/python-telegram-bot>.
11. Heroku Documentation. <https://www.heroku.com/>.
12. Python Telegram Bot Documentation. <https://github.com/python-telegram-bot/python-telegram-bot>.