Чаку О.C.

Національний університет "Львівська політехніка",

кафедра електронних обчислювальних машин

**Програмний сервіс розпізнавання типу сміття на базі мобільних обчислювальних приладів з графічним вводом даних**

*© Чаку О.C., 2019*

**Розглянуто проблему визначення типу сміття, пошуку і побудування маршруту до найближчих місць утилізації відповідного типу. Аналіз графічного зображення реалізований за допомогою нейронної мережі, побудування маршруту – з використанням GPS та** **акселерометру. В якості бази імплементації вибрана платформа iOS, через наявність сенсор глибини в останніх моделях iPhone для визначення контуру об’єкту.**

**Ключові слова: нейронна мережа, аналіз зображень, сенсор глибини, GPS, акселерометр.**

**The problem of determining the type of garbage is really actual nowadays, as well as finding and constructing a route to the nearest disposal sites of the appropriate garbage type. Graphical image analysis is performed using a neural network, route construction – using GPS and an accelerometer.**  **The iOS platform was chosen as the implementation base because of the depth sensor in the latest iPhone models that helps to determine the shape of the object.**

**Keywords: neural network, image analysis, depth sensor, GPS, accelerometer.**

**Вступ.** В сучасно світі мобільні обчислювальні пристрої досягли дуже високого рівня у свої фотографічних здібностях. Високий рівень вихідних зображень дозволяє достовірно передавати колір та деталізацію об’єкту, включно з найдрібнішими текстами на ньому. Також деякі виробники додають ще окремі сенсори для отримання різних оптичних ефектів на фотографіях – це або лазерне фокусування, або RGB-сенсори для більш точної передачі кольорів, або різні спроби отримати портретні зображення з відокремленим фоном як на дзеркальних фотокамерах.

Окрім цього, стрімко зростає й обчислювальна потужність, кількість оперативної пам’яті та об’єм накопичувача у таких мобільних пристроїв. А через велику зацікавленість кінцевих користувачів у оброці фотографій перед публікацією у соціальних мережах та запуску та проводження часу в дорозі у мобільних іграх – виробники процесорів для мобільних обчислювальних пристроїв приділяють ще більшу увагу розробці і покращення графічного ядра.

Незважаючи на стрімкий розвиток мобільних пристроїв, паралельно у світі все збільшується і збільшується кількість неутилізованого сміття. Це сміття не тільки загромаджує вільне місце, але й отрує нашу планету – тому варто об’єднати розвиток з користю для нашого ж дому.

**Стан проблеми.** Критичність проблеми утилізації сміття дуже різна в залежності від країни – десь сам уряд приймає якихось активних кроків щодо пришвидшення цих процесів, десь все тримається тільки на ініціативі активістів. Саме такий другий стан в нас у Україні, немає єдиного, загальнодержавного процесу з роздільними смітниками коли житлових будинків та публічних місцях, спонуканням громадян до дотримання правил сортування та покаранням за хаотичні сміттєзвалища у непризначених для того місцях.

Візьмемо в якості конкретного прикладу наше місто Львів. У Львові є певна кількість або маленьких приватних переробних заводів для певного типу, або певних активістів які пересилають це кудись, або заводів які використовують сміття як сировину (для прикладу скло для пляшок). Проблема полягає у тому що пересічний громадян окрім того що з великою вірогідністю не знає де розташовані оці всі маленькі пункти прийому, так зазвичай ще й не до кінця впевнений який саме тип сміття в нього хто його приймає. Оця необхідність самостійно шукати інформацію по кожному з об’єктів самостійно відбиває і так невелику мотивацію в людей займатись сортуванням і утилізацією.

**Постановка задачі.**

Базуємось на німецький класифікації типів сміття з 7-ма різними категоріями. Розробити мобільних додаток якій використовуючи камеру мобільного обчислювального пристрою та нейронну мережу визначає відповідний тип, після чого пропонує маршрути до найближчих пунктів утилізації.

**Розв’язання задачі.**

Програмна система складається з кількох основних модулів які відповідаю за різні етапи обробки вхідних даних та подальших дій – структура та взаємозв’язок зображені на (рис. 1). Система включає в себе модуль роботи з камерою пристрою, модуль аналізу даних, а також модулі необідні для роботи користувацьким інтерфейсом і навігацією.

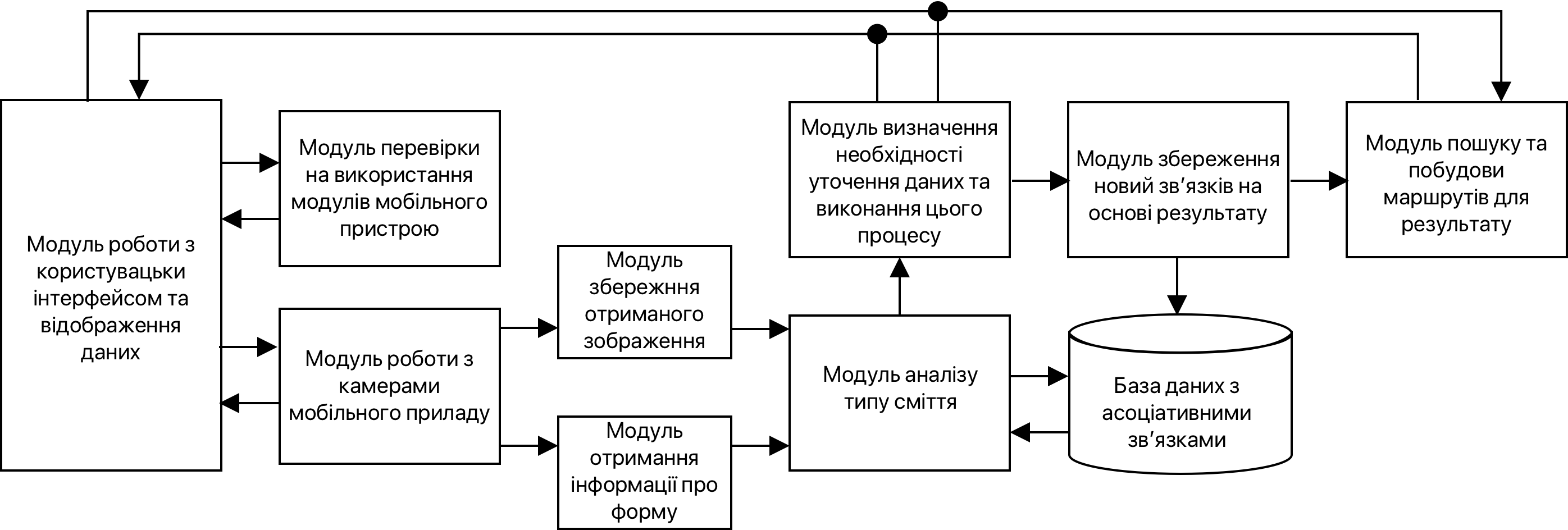


Рис. 1. Структурна модулів програмної системи та їх взаємозв’язків.

Алгоритм роботи програмного модулю аналізу графічних даних використовує вихідні дані модуля обробки цифрових даних які включають безпосередньо зображення об’єкту вирізане з фону та форму об’єкта у математичному вигляді з її хеш-сумою для швидкого порівняння. На етапі аналізу нейронна мережа використовуючи збережені асоціативні правила з бази даних намагається за допомогою примітивних рецепторів визначати якій з типів (кожний має відповідне) має найбільше спільного, а які можна відкинути відразу. Вибір налічує 7 різних типів з відповідними кольоровими кодуваннями. На наступному етапі використовуються більш складні перевірки + сторонні програмних інтерфейс якій дозволяє отримувати складові частини предмету для уточнення правильності вибраної категорії. (рис. 2)

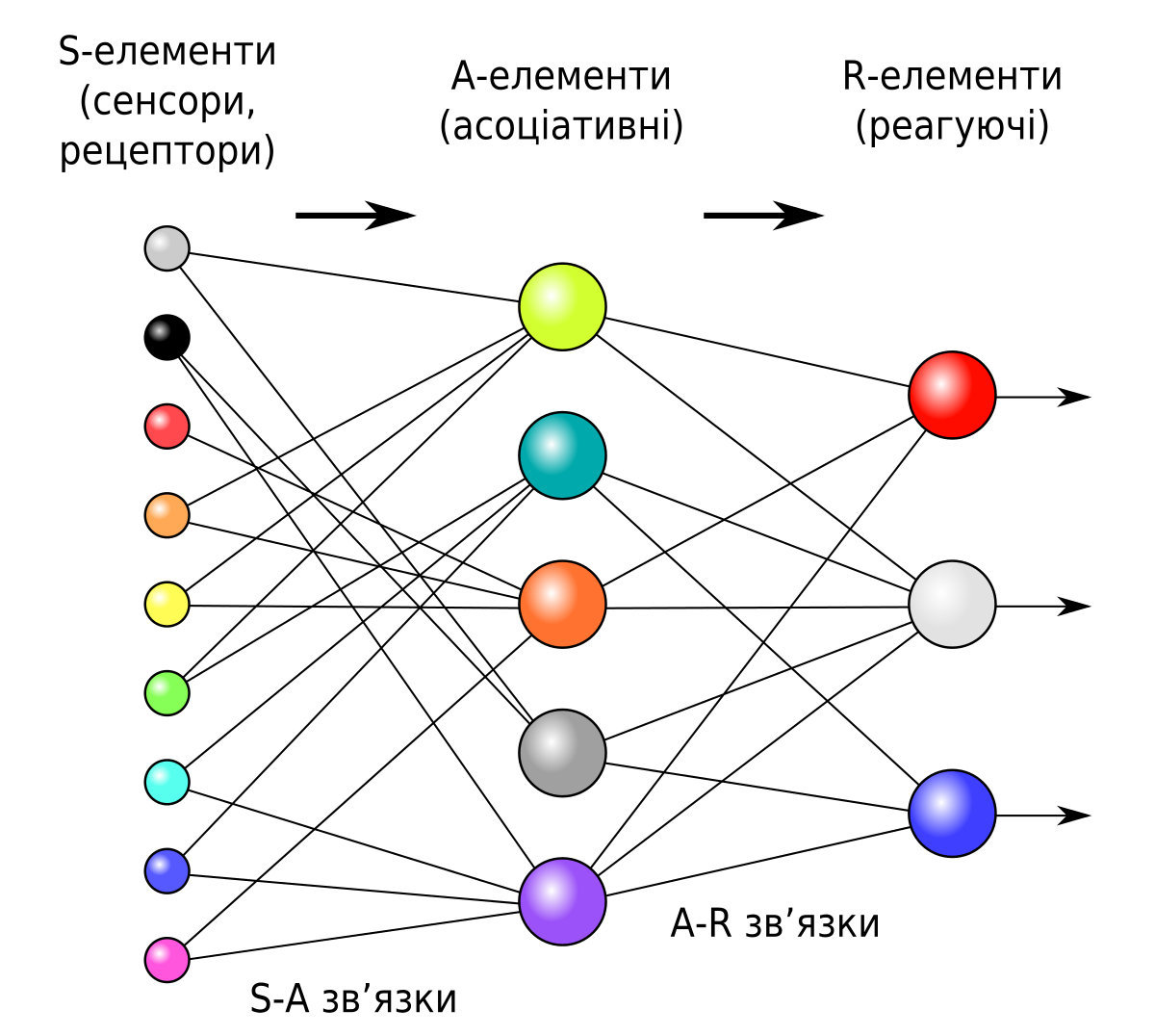


Рис. 2. Алгоритм роботи нейронної мережі використовуючи противні рецептори та більш складні перевірки для визначення кінцевого результату.

Якщо на останньому етапі кількість реагуючих елементів більше ніж 1, то нейронна мережа до кінця не впевнена якісь це тип і просить користувача вибрати один з найбільш релевантних. Після цього система зберігає нову асоціацію в базу даних і буде використовувати її для порівняння.

На наступній схемі знаходяться кроки які будуть відбуватись з кінцевим користувачем під час використання програмної системи. Наступні стани та переходи між екранами залежать від результатів роботи алгоритмів аналізу та рішень користувача де це потрібно. (рис. 3)

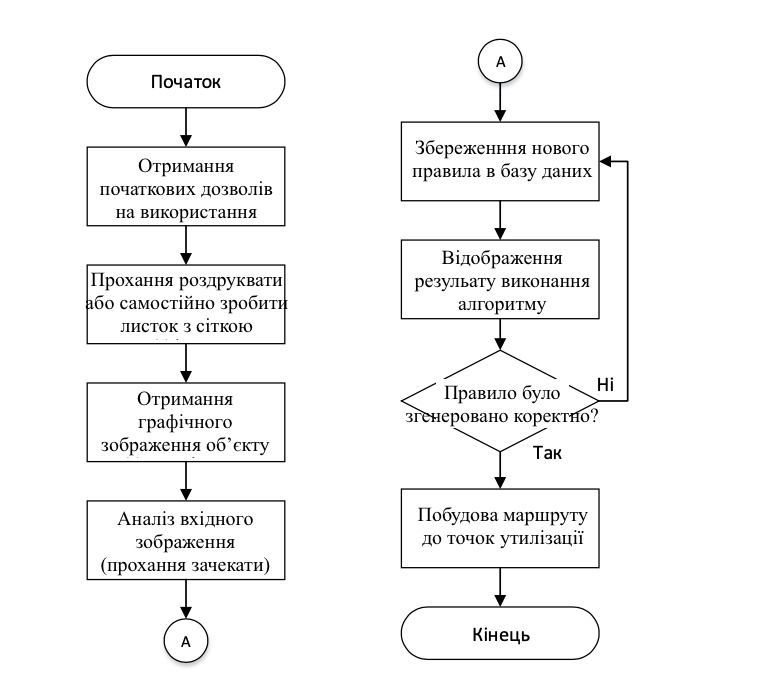


Рис. 3. Алгоритм взаємодії програмної системи з користувачем і відповідні переходи між різними станами (екранами) додатку

**Висновки.**

У даній роботі запропоновано інновацій підхід до сортування сміття. Для його реалізації не потрібно закупляти спеціальне обладнання, а достатньо скачати додаток з магазину. Також розроблений алгоритм може бути реалізований під інші операційні системи, варіюватись буде тільки методи визначення форми об’єкту. Побудована програмна система успішно виконує поставлене завдання та навчається в процесі роботи щодалі покращуючи якість розпізнавання. Вбудовані картографічні та навігаційні здібності дозволяють легко і зручно будувати маршрут до певного місця утилізації та переробки відповідного типу сміття.

**Література.**

1. Морозов А. В. Основні принципи ООП. [Електронний ресурс] / А. В. Морозов. – 2016. – Режим доступу до ресурсу: https://learn.ztu.edu.ua/mod/page/view.php?id=7255.
2. Apple Developer Documentation – 2018 – Apple Inc. – <https://developer.apple.com/documentation>.
3. Themes - Overview - iOS Human Interface Guidelines — 2018 — Apple Inc. – <https://developer.apple.com/ios/human-interface-guidelines/overview/themes/>.
4. MapKit | Apple Developer Documentation – 2018 – Apple Inc. – Режим доступу: <https://developer.apple.com/reference/mapkit>.
5. Encoding and Decoding Custom Types | Apple Developer Documentation – 2018 – Apple Inc. – Режим доступу: <https://developer.apple.com/documentation/foundation/archives_and_serialization/encoding_and_decoding_custom_types>
6. Generics in Swift 4 / Candost Dağdeviren – 2017 – Режим доступу: <https://theswiftpost.co/generics-swift-4/>
7. Overview | Geocoding API | Google Developers – 2018 – Режим доступу: <https://developers.google.com/maps/documentation/geocoding/intro?hl=en>
8. Protocol Based Generic Networking using JSONDecoder and Decodable in Swift 4 / James Rochabrun – 2017 – Режим доступу: <https://medium.com/@jamesrochabrun/protocol-based-generic-networking-using-jsondecoder-and-decodable-in-swift-4-fc9e889e8081>.
9. A deep dive into Grand Central Dispatch in Swift / Jonh Sundell – 2017 – Режим доступу: <https://www.swiftbysundell.com/posts/a-deep-dive-into-grand-central-dispatch-in-swift>
10. Engineering Fast Route Planning Algorithms / Peter Sanders, Dominik Schultes – Universit ̈at Karlsruhe (TH), 2007 – 36 c.
11. Realm: Create reactive mobile apps in a fraction of time – 2018 – Realm – Режим доступу: <https://realm.io/docs/swift/latest/>.
12. Writing a Scalable API Client in Swift 4 / Víctor Pimentel – 2017 – Режим доступу: <https://medium.com/makingtuenti/writing-a-scalable-api-client-in-swift-4-b3c6f7f3f3fb>