

## РОЗДІЛ II. ІНФОРМАЦІЙНО-КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

УДК 004.382:004.042

DOI: 10.25140/2411-5363-2018-4(14)-118-127

Інна Стеценко, Марина Суханюк, Владислав Шишкін

### ІНФОРМАЦІЙНА СИСТЕМА РОЗУМНОГО ВІДЕОРЕЄСТРАТОРА

**Актуальність теми дослідження.** Нині великою популярністю користуються «розумні пристрої». Такі пристрої зазвичай є певною модернізацією вже звичних речей. У цій статті представлена розробка пристрою «розумний відеореєстратор», що підтримує можливість розпізнавання українських автомобільних номерів. Такий засіб може підвищити рівень захищеності на дорогах, що також є актуальним питанням.

**Постановка проблеми.** У процесі розробки комплексу такого рівня з'являється безліч питань, пов'язаних із його архітектурою та роботою з даними, а саме: які методи та алгоритми використовувати для перетворення даних у потрібний формат, передачі, прийому, консолідації даних та зберігання їх у базі даних. Крім того, необхідно взяти за увагу, що в процесі побудови необхідно враховувати фізичні особливості модулів та їхні можливості з обробки та передачі даних (швидкість роботи процесора, розмір пам'яті).

**Аналіз останніх досліджень та публікацій.** Були розглянуті останні технології у сфері обробки даних (бібліотеки для серіалізації та десеріалізації), алгоритми розпізнавання автомобільних номерів та бази даних із можливістю текстового пошуку.

**Виділення недослідженої частини загальної проблеми.** Побудова архітектури та розробка програмних модулів розумного відеореєстратора, вирішення задачі передачі даних до кластера за умови великої завантаженості та переривчастого інтернет-зв'язку, повнотекстовий та частковий пошук автомобільних номерів у базі даних, алгоритм розпізнавання автомобільних номерів у русі.

**Постановка задачі.** Вибір необхідної комбінації методів та алгоритмів для успішної реалізації інформаційної системи розумного відеореєстратора.

**Виклад основного матеріалу.** Опис основних модулів, з яких складається пристрій розумного відеореєстратора, та з якою метою використовується кожний модуль. Представлена схема роботи системи загалом та описаний алгоритм знаходження автомобільних номерів у режимі реального часу YOLO. Розглянуто основні принципи комунікації між серверами та пристроями.

**Висновки відповідно до статті.** Наданий матеріал надає змогу зрозуміти, яким чином може бути побудований такий пристрій, які проблеми можуть з'явитись та як знайти шляхи їх вирішення.

**Ключові слова:** відеореєстратор; розумні пристрої; ASUS Tinker Board; Satellite Based Augmentation System; GPS; NMEA, GPRSD; JSON, серіалізація даних; YOLO.

Рис.: 8. Табл.: 2. Бібл.: 11.

**Актуальність.** Нині великою популярністю користуються «розумні пристрої». Такі пристрої зазвичай є певною модернізацією вже звичних речей. У цій статті представлена розробка пристрою «розумний відеореєстратор», що підтримує можливість розпізнавання українських автомобільних номерів.

За даними УНН, в Україні протягом січня-червня 2017 року було зафіксовано 3563 випадки викрадення автомобілів. Зокрема, 647 автомобілів було викрадено тільки у м. Києві. Ці дані були надані у відповідь інформаційному запиту УНН Національної поліцією України.

Такий засіб може підвищити рівень захисту на дорогах, що також є актуальним питанням.

**Постановка проблеми.** Програмно-апаратний комплекс розумного відеореєстратора являє собою сукупність модулів апаратного забезпечення та методів обробки вхідної відеоінформації, що об'єднані з метою отримання масиву даних для розпізнавання автомобільного номера та передачі його на кластер для подальшої обробки та аналізу. У процесі розробки комплексу такого рівня з'являється безліч питань, пов'язаних із його архітектурою та роботою з даними, а саме: які методи та алгоритми використовувати для перетворення даних у потрібний формат, передачі, прийому, консолідації даних та зберігання їх у базі даних. Крім того, необхідно взяти за увагу, що в процесі побудови пристрою необхідно враховувати фізичні особливості модулів та їхні можливості з обробки та передачі даних (швидкість роботи процесора, розмір пам'яті).

**Аналіз останніх досліджень та публікацій.** Було розглянуто останні дослідження у сфері роботи з даними, що були надані модулем GPS (формат NMEA), процесів, пов'язаних із серіалізацією та десеріалізацією. Проаналізовано останні міжнародні публікації у сфері алгоритмів розпізнавання, а саме: Yolov2, Yolo, Tiny Yolo та CNN, що їх здійснили такі автори, як Kiana Ehsani, Hessam Bagherinezhad, Joseph Redmon, Roozbeh Mottaghi, Ali Farhadi, Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari.

**Постановка завдання.** Метою цієї роботи є опис програмно-апаратного комплексу розумного відеореєстратора із можливістю розпізнавання автомобільних номерів із відеопотоку в режимі реального часу в умовах руху.

**Виділення недосліджених частин загальної проблеми.** Рішення проблеми вибору методів серіалізації та десеріалізації даних, алгоритму розпізнавання автомобільних даних із відеопотоку, побудова архітектури та реалізація програмних модулів розумного відеореєстратора, передача даних за умов великої навантаженості на кластер та використання методів ElasticSearch для пошуку автомобільних номерів, які були отримані за допомогою алгоритму розпізнавання автомобільних номерів, що був реалізований як програмна частина розумного відеореєстратора.

**Виклад основного матеріалу.** На сучасному ринку вже існують пристрої, функціонально схожі на розумний відеореєстратор. Однак головна відмінність таких пристроїв – статичність камери. У пристрої, модель якого представлена далі, важливим досягненням є те, що він являється програмно-апаратним комплексом, який поєднує в собі апаратні частини: датчики та мікрокомп'ютер, які можуть бути об'єднані в мобільний та невеликий за розміром пристрій під назвою «розумний відеореєстратор» та програмну частину – розпізнавання автомобільних номерів (рис. 1).

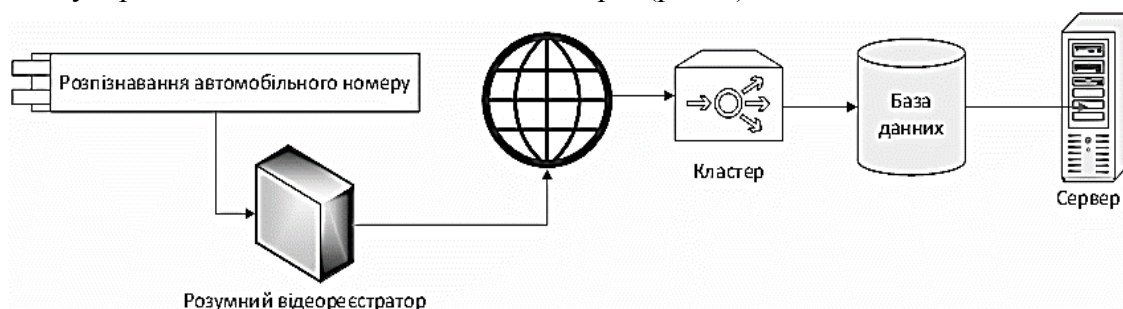


Рис. 1. Апаратно-програмний комплекс розумного відеореєстратора

Далі розглядаються основні модулі, які були використані для побудови моделі розумного відеореєстратора.

#### Основа відеореєстратора.

Основою став одноплатний комп'ютер ASUS Tinker Board. Він має схожу архітектуру, що і в Raspberry Pi. Ця плата підтримує відео в розширенні 4K, має 2 Гб оперативної пам'яті, гігабітний інтернет і процесор Rockchip RK3288 з частотою 1,8 ГГц [1]. Завдяки процесору Rockchip цей мікрокомп'ютер добре пристосований до обробки відеоінформації.

#### Опис компонентів.

Складові модулі розумного відеореєстратора наведені далі.

1. GPS модуль від компанії WaveShare на основі NEO-6M. Цей модуль виконує функціональність, пов'язану з обчисленням координат, швидкості, курсу й інших параметрів. Приймач використовується в системі для отримання координат транспортного засобу. NEO-6M має середню точність та невеликий розмір, що є важливим аспектом. Neo-6M вміє використовувати супутникові системи диференціальної корекції, що збільшує точність визначення положення до 2 м.

2. Модуль камери від компанії Raspberry. Для роботи із Raspberry Pi Camera використовують інфрачервоні (NoIR) та звичайні камери (v2). У цьому проекті використовується звичайна камера. Версія v2 має значні переваги над першою, надає можливість знімати FULL HD відео з кадровою частотою 30 fps та має кращу якість зйомки. Крім того, кількість мегапікселів збільшилась на 3 одиниці.

3. Монітор руху 10 DOF IMU Sensor (C) - модуль десяти ступенів свободи від Waveshare. Цей модуль об'єднує в собі трьохосьовий гіроскоп, трьохосьовий акселерометр і трьохосьовий компас (магнітометр) на одному кристалі MPU9255 та барометричний датчик тиску BMP280. Вибір саме цього датчика зумовлений ціною та якістю збирання. Крім того, наявна можливість додатково доповнювати функціонал відеореєстратора такими можливостями: стабілізація камери, контроль нахилу відеореєстратора і т. ін.

4. Модем LTE. Якість Інтернету відіграє вирішальну роль, оскільки використання розумного відеореєстратора тісно пов'язане з постійним обміном даних між сервером та відеореєстратором. Через постійний рух автомобіля, на якому встановлений відеореєстратор, можуть спостерігатися збої в передачі даних або мережеві ями.

5. СУБД SQLite. Вибір цієї СУБД зумовлений такими факторами:

- невеликий розмір;
- популярність серед портативних пристроїв;
- швидкість;
- надійність.

Крім того, СУБД SQLite підтримує SharedPreferences – це сховище ключів-значень, в якому зберігаються дані під певним ключем, а саме, координати транспортного засобу, на якому встановлений розумний відеореєстратор.

На рис. 2 представлені основні модулі відеореєстратора, їхні складові та як вони взаємодіють один з одним.

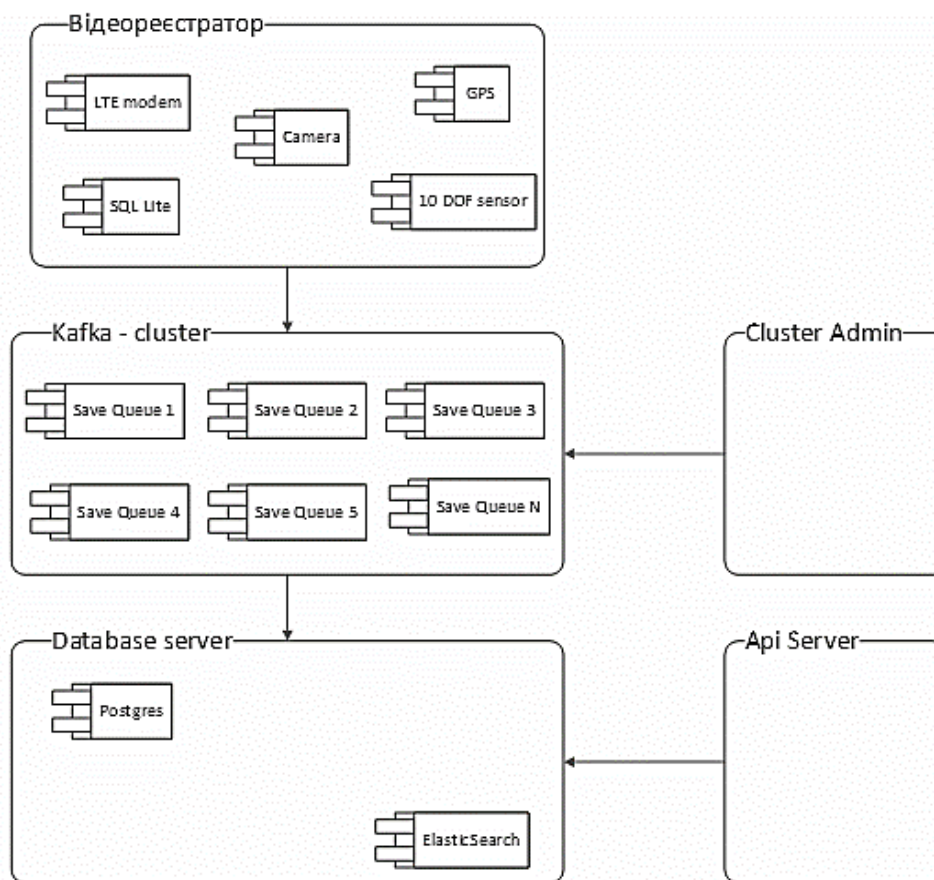


Рис. 2. Схема ключових модулів проекту

*Датчик GPS.*

До Asus Tinker Board був підключений датчик NEO-6M, за допомогою якого відслідковується положення транспортного засобу. Цей пристрій передає дані, які він отримав від супутників у форматі NMEA кожні N секунд. Дуже важливою є інформація про ширину, довготу, висоту та швидкість руху об'єкта. Для підключення датчика до Asus tinker board необхідно з'єднати контакти плати та датчик NEO-6M за такою схемою, що наведена в табл. 1.

Таблиця 1

*Підключення датчика до Asus Tinker Board*

Asus Tinker Board/ Raspberry PI	Waveshare GPS Neo6m
Pin 1 (3.3 V)	VCC
Pin 8 (TX)	RX
Pin 10 (RX)	TX
Pin 6 (GND)	GND

*Робота з форматом NMEA.*

Для роботи з датчиком NEO-6M використовується бібліотека GPSD [2]. Ця бібліотека підтримує виконання таких функцій:

- 1) синтаксичний розбір вхідного потоку даних у форматі NMEA з датчику GPS Neo6m;
- 2) перетворення даних у формат JSON;
- 3) фільтрування даних від непотрібної або неправильної інформації;
- 4) надає можливість передавати дані за допомогою TCP/UDP та інших популярних протоколів.

При роботі з форматом NMEA необхідно враховувати його команди. Наведемо перелік основних команд NMEA 0183 версії 3:

- GPGGA – дані про останнє місцезнаходження пристрою;
- GPGLL – координати: широта, довгота, висота;
- GPGSA – DOP(GPS) активні супутники;
- GPGSV – спостережувані супутники;
- GPWPL – параметри заданої точки;
- GPBOD – азимут однієї точки відносно іншої;
- GPRMB – рекомендований мінімум навігаційних даних для досягнення заданої точки;
- GPRMC – рекомендований мінімум навігаційних даних;
- GPRTE – маршрути;
- HCHDG – дані від компасу [3].

Бібліотека GPSD має певні недоліки та не відповідає всім вимогам проекту, який розробляється. За результатами проведеного тестування навантаження дана бібліотека не є стабільною, якщо встановлена велика кількість з'єднань. Використання TCP/UDP з'єднання не є доречним, оскільки при пересуванні транспортного засобу Інтернет має властивість «пропадати», що призводить до постійного оновлення підключення. Дані знайдених транспортних засобів та положення транспортного засобу передаються одним блоком даних.

Бібліотека GPSD написана на мові C. Для того щоб інтегрувати її в проект, був написаний клієнт на Java, який підключається до відкритого сервера GPSD (є доступним тільки локально), обмінюється інформацією та передає необхідні команди.

*Формат JSON.*

Бібліотека GPSD надає дані у форматі JSON, проте для подальшої роботи необхідно перетворити ці дані в більш зручний для обробки формат. Для забезпечення ефективності обробки даних необхідна висока швидкість серіалізації/десеріалізації та мінімально можливі витрати пам'яті [4].

Було проведено тестування найпопулярніших бібліотек для мови програмування Java GSON [5] та Jackson [6]. У результаті випробувань бібліотека GSON виявилась швидшою, проте її швидкість недостатня для використання в проєкті.

Через низьку ефективність формату JSON для передачі даних між сервером та пристроєм у цьому проєкті використовується бібліотека ProtoBuf від компанії Google [7], яка є бінарною та за результатами тестувань є швидшою за GSON.

Далі наведено результати тестування бібліотек GSON, Jackson и Protobuf за наступними критеріями:

- 1) швидкість серіалізації і десеріалізації для 100 000 ітерацій;
- 2) швидкість серіалізації і десеріалізації для 10 000 000 ітерацій;
- 3) використання CPU;
- 4) динаміка використання пам'яті.

Результати даних випробувань наведені в табл. 2.

Таблиця 2

*Результати випробувань*

Критерій	GSON	Jackson	Protobuf
Швидкість серіалізації й десеріалізації для 100 000 ітерацій, мс	445	13544	210
Швидкість серіалізації і десеріалізації для 10 000 000 ітерацій, мс	3223	91877	903
Використання ЦП, %	12,5	12,6	12,1
Розмір задіяних ресурсів на останній ітерації, Б	39 088 584	59 816 548	49 696 136

*Кластер Kafka.*

Для обміну даними між відеореєстратором та сервером як брокер повідомлень використовується технологія, розроблена компанією LinkedIn під назвою Kafka. Сервіс підтримує такі можливості:

- кластеризація;
- зберігання повідомлень;
- можливість використання різних баз даних;
- кешування;
- ведення журналів.

Для реалізації серверної частини використовується кластер з N серверів Kafka. Це надає можливість розвантажити кластер, який не здатний обробити вхідний потік інформації з усіх пристроїв. За рахунок розподіленої технології передачі даних є можливість додавати сервери до кластера для розвантаження та збалансування черги запису до бази даних.

*Бібліотека YOLO.*

Для виявлення об'єктів у реальному часі використовується бібліотека YOLO, найбільша перевага якої відображена власне в її назві: «You Only Look Once» [8]. Існує декілька версій конфігурацій і наборів даних для машинного навчання, однак найкраще обрати свій варіант, яким є найбільш «компактна» версія продукту – Tiny YOLO. Важливою перевагою Tiny YOLO є те, що вона не втрачає швидкості, навіть працюючи на таких пристроях, як смартфони та Raspberry Pi.

Особливістю моделі є те, що ця модель накладає на зображення сітку, розділяючи його на осередки (рис. 3).



Рис. 3. Накладення сітки на зображення

Першим етапом цього алгоритму являється обробка зображення з метою отримання його стиснутої копії. Наприклад, на вхід подається зображення розмірності  $448 \times 448 \times 3$ . Після пропускання зображення через операції конволюції [10] отримаємо на виході тензор розмірності  $7 \times 7 \times 3$ .

Схематично цей процес зображений на рис. 4.

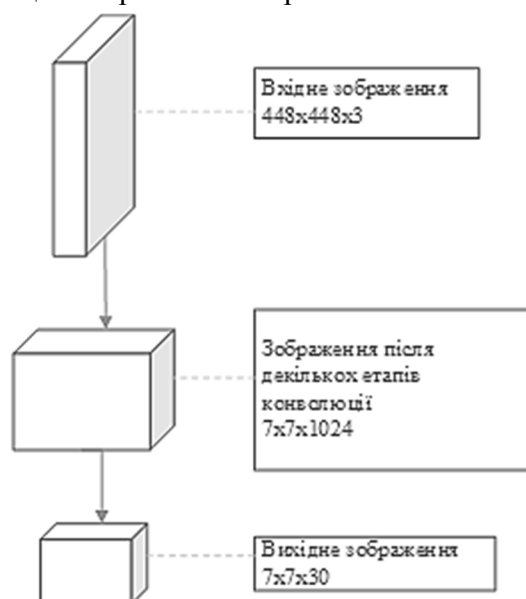


Рис. 4. Конволюція зображення

Після отримання тензора розмірністю  $7 \times 7 \times 30$  продовжується робота саме з ним, та за допомогою нього має бути задіяна процедура знаходження номерного знака.

Отже, сітку розмірністю  $7 \times 7$  було накладено на зображення. Для кожної комірки будується вектор (рис. 5).

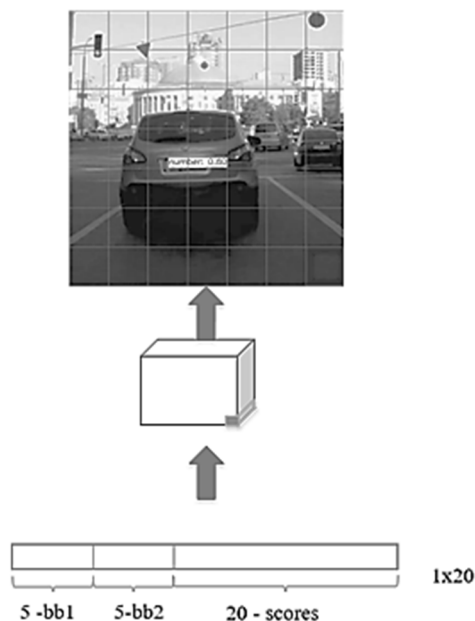


Рис. 5. Побудова вектору для кожної комірки

У результаті було отримано вектор  $20 \times 1$ , який містить оцінки для кожного класу для конкретного bbox-a.

Ця операція повторюється для кожної комірки сітки  $7 \times 7$ . Таким чином була отримана матриця оцінок розмірністю  $20 \times 98$ . Дана матриця оброблюється з метою залишити в ній тільки ті обрамляючі прямокутники, які найвірогідніше розташовані навколо потрібного класу, які й накладаються на вихідне зображення.

Вихідним даними етапу знаходження потрібної області є область зображення, якій відповідає номерний знак автомобіля, що відображений на рис. 6.



Рис. 6. Область номерного знаку автомобіля

Знайдена область потім обробляється за допомогою алгоритму Tesseract [11]. На виході маємо номерний знак у строковому вигляді: «CB2558AK».

*Збереження даних.*

Зберігання даних грає важливу роль в інформаційній системі. У представленій інформаційній системі дані потрапляють до бази даних вже після їх обробки.

У цьому випадку використані дві бази даних: Elasticsearch та Postgres. В Elasticsearch зберігаються дані про місцезнаходження пристрою та знайдені автомобільні номери на цій ділянці дороги. У Postgres зберігаються дані про пристрої, їхні параметри (клас пристрою, параметри пристрою, налаштування, список «чорних» номерів) та дані про користувача.

Далі наведений алгоритм обробки даних про місцезнаходження пристрою.

Крок 1. Отримання даних з датчику GPS у форматі NMEA та отримання даних відеопоток.

Крок 2. Фільтрація та перетворення отриманих даних за допомогою бібліотеки GPST.

Крок 3. Використання алгоритму YOLO для знаходження автомобільних номерів.

Крок 4. Зчитування автомобільного номеру на знайдений області.

Крок 5. Відсіювання некоректно зчитаних автомобільних номерів.

Крок 6. Серіалізація даних за допомогою протоколу Protocol Buffers.

Крок 7. Передача даних до кластеру Kafka.

TECHNICAL SCIENCES AND TECHNOLOGIES

Крок 8. Збереження даних з Kafka до ElasticSearch.

ElasticSearch надає Restfull Арі для збереження даних у свою БД у форматі JSON та надає можливість для повнотекстового пошуку.

На рис. 7 наведений приклад структури даних, які зберігаються в ElasticSearch. На рис. 8 відображена схема бази даних в Postgres.

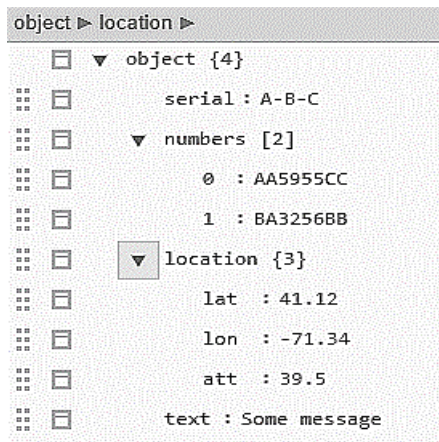


Рис. 7. Структура даних, що відповідає за збереження даних автомобільних номерів та місцезнаходження приладу

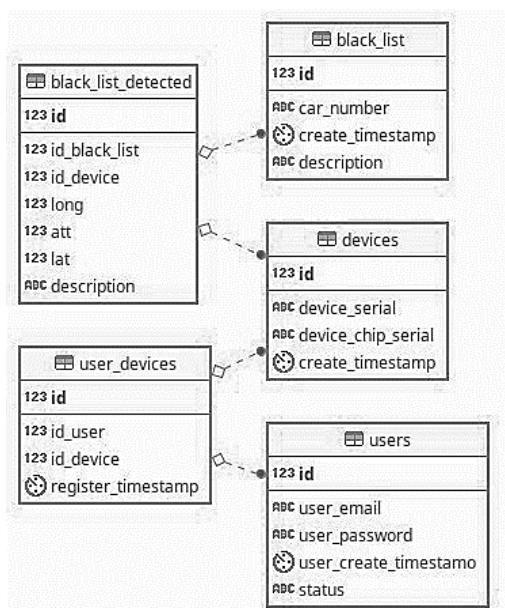


Рис. 8. Схема бази даних, що відповідає за збереження даних про пристрій та користувача

**Висновок відповідно до статті.** У роботі були розглянуті основні модулі для розробки інформаційної технології розумного відеореєстратора: ASUS Tinker Board, датчики, перетворення даних, клієнтський сервер. Описано вибір оптимального формату даних, який забезпечує прийнятну швидкість передачі та прийому. За результатами тестування доведено, що Protobuf має більш високі показники якості в порівнянні з бібліотеками GSON та Jackson.

Наведена архітектура моделі розумного відеореєстратора з описом компонентів. Описані основні засади алгоритму знаходження автомобільного номеру: обробка зображення, накладення сітки, побудова вектору та обрамляючих прямокутників. Наведені вихідні дані алгоритму та описана подальша робота з ними.

Також у роботі наведено опис збереження даних у базах даних ElasticSearch та Postgres.



**Список використаних джерел**

1. Chan J. W. Learn 5 Single Board Computer: Raspberry Pi, Asus Tinkerer Board, Banana PI M2, Pine A 64, Chip, Rock 64 / James Watson Chan., 2018. – 240 с.
2. El-Rabbany A. Introduction to GPS: The Global Positioning System / Ahmed El-Rabbany., 2002. – 176 с.
3. Kaplan E. Understanding GPS/GNSS: Principles and Applications / E. Kaplan, C. J. Hegarty., 2017. – 1002 с.
4. Kleppmann M. Designing Data-Intensive Applications / Martin Kleppmann. – Sebastopol : O'Reilly Media, 2010.
5. Patel S. K. Instant GSON / Sandeep Kumar Patel., 2013. – 60 с.
6. Young T. M. The Jackson Cookbook / Ted M. Young. // LeanPub. – 2013. – С. 1–16.
7. Protocol buffers [Електронний ресурс]. – Режим доступу : <https://developers.google.com/protocol-buffers>.
8. Redmon J. YOLOv3: An Incremental Improvement / J. Redmon, A. Farhadi // University of Washington. – 2016. – С. 1–6.
9. Rother C. Computer Vision I - Image Processing / Carsten Rother // Computer Vision Lab. – 2016. – С. 41–46.
10. Redmon J. Real-Time Grasp Detection Using Convolutional Neural Networks / J. Redmon, A. Angelova. // University of Washington. – С. 1–7.
11. Smith R. An Overview of the Tesseract OCR Engine / Ray Smith // IEEE. – 2007. – С. 629–633.

**References**

1. Statystyka vykradennia avtomobiliv [Car Jacking Statistics]. (n.d.). [www.unn.com.ua](http://www.unn.com.ua). Retrieved from <https://www.unn.com.ua/uk/news/1676376-v-ukraini-za-pivroku> [in Ukrainian].
2. Chan, J. W. (2018). *Learn 5 Single Board Computer: Raspberry Pi, Asus Tinkerer Board, Banana PI M2, Pine A 64, Chip, Rock 64*. Amazon Digital Services LLC [in English].
3. El-Rabbany, A. (2006). *Introduction to GPS: The Global Positioning System*. Boston: Artech House [in English].
4. Kaplan, E., Hegarty, C. J. (2006). *Understanding GPS principles and applications*. Boston: Artech House [in English].
5. Kleppmann, M. (2010). *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems*. Sebastopol, CA: O'Reilly Media [in English].
6. Patel, S. K. (2013). *Instant GSON*. Packt Publishing [in English].
7. Welty, E. (2001). *The Jackson cookbook*. Jackson, MS: The League [in English].
8. Protocol Buffers | Google Developers. (n.d.). [developers.google.com](https://developers.google.com/protocol-buffers/). Retrieved from <https://developers.google.com/protocol-buffers/> [in English].
9. YOLOv3: An Incremental Improvement. (n.d.). [pjreddie.com](http://pjreddie.com). Retrieved from <https://pjreddie.com/media/files/papers/YOLOv3.pdf> [in English].
10. Rother, C. (2016). *Computer Vision I - Image Processing*. Computer Vision Lab [in English].
11. Redmon, J., Angelova, A. (2014, December 01). Real-Time Grasp Detection Using Convolutional Neural Networks. [adsabs.harvard.edu](http://adsabs.harvard.edu). Retrieved from <http://adsabs.harvard.edu/abs/2014arXiv1412.3128R>. [in English].
12. Smith, R. (2007). *An Overview of the Tesseract OCR Engine*. IEEE [in English].

UDC 004.382:004.042

*Inna Stetsenko, Maryna Sukhaniuk, Vladyslav Shyshkin***SMART DVR INFORMATION SYSTEM**

**Relevance of research topic.** Nowadays, smart devices are very popular, namely modern devices with built-in information technologies. This article presents the development of a smart DVR that supports the ability to recognize Ukrainian automobile numbers and their further analysis. Such a tool can speed up the search of stolen cars, which is a topical issue.

**Target setting.** During the development of the hardware-software complex, issues related to its architecture and data processing are solved, namely, which methods and algorithms are used for the transformation, transmission, receiving, consolidation of data and storing in the database. It should be kept in mind that during construction of device, the physical features of the modules and their processing and transmission capabilities, such as processor speed, memory size have to be considered.

## TECHNICAL SCIENCES AND TECHNOLOGIES

**Analysis of recent research and publications.** The latest technologies in the field of data processing (serialization and deserialization libraries), automobile number recognition algorithms and the database with the possibility of accelerated text search are analyzed.

**Uninvestigated parts of general matters defining.** The architecture and software modules of smart DVR is proposed for the first time, the problem of data transmission to the cluster under the conditions of high load and intermittent Internet communication is solved; full-text and partial search of automobile numbers in the database is proposed; algorithm of automobile numbers recognition in motion is developed.

**The research objective.** Identify the required combination of methods and algorithms for the implementation of the smart DVR information system, which ensures the reliable and efficient execution of its functional capabilities.

**The statement of basic materials.** The article presents the description of modules, which represent the smart DVR, and the purpose of each module is taken into consideration. The scheme of the system as a whole is presented and the algorithm of real-time search of automobile numbers based on the YOLO neural network is developed. The basic principles of communication between servers and devices are considered.

**Conclusions.** The architecture of a new smart device is proposed and the main algorithms that implement its functionality are developed. Identified problems that may arise in further development and identified ways to resolve them.

**Keywords:** DVR; smart device; ASUS Tinker Board; GPS; NMEA; GPSD; serialization; YOLO.

*Fig.: 8. References: 11.*

**Стеценко Інна Вячеславівна** – доктор технічних наук, професор кафедри автоматизованих систем обробки інформації та управління, НТУУ «КПІ ім. Ігоря Сікорського» (вул. Борщагівська, 126, Київ, 03056, Україна).

**Stetsenko Inna** – Doctor of Science, Professor of Computer-Aided Management And Data Processing Systems Department, NTUU «Igor Sikorsky Kyiv Polytechnic Institute» (126 Borschagivska Str., 03056 Kyiv, Ukraine).

**E-mail:** stiv.inna@gmail.com

**Суханюк Марина Валентинівна** – студентка, НТУУ «КПІ ім. Ігоря Сікорського» (вул. Борщагівська, 126, Київ, 03056, Україна).

**Sukhaniuk Maryna** – student, NTUU “Igor Sikorsky Kyiv Polytechnic Institute” (126 Borschagivska Str., 03056 Kyiv, Ukraine).

**E-mail:** sukhaniuk.marina@gmail.com

**Шишкін Владислав Ігорович** – студент, НТУУ «КПІ ім. Ігоря Сікорського» (вул. Борщагівська, 126, Київ, 03056, Україна).

**Shyshkin Vladyslav** – student, NTUU “Igor Sikorsky Kyiv Polytechnic Institute” (126 Borschagivska Str., 03056 Kyiv, Ukraine).

**E-mail:** shyshkin.vladyslav@gmail.com