

УДК 004

DOI <https://doi.org/10.32689/maup.it.2025.1.29>**Андрій СЕРГАЧОВ**

магістр кафедри Спеціалізованих комп'ютерних систем,
Навчально-наукового інституту «Український державний хіміко-технологічний університет»,
Український державний університет науки і технологій, a.serhachov@gmail.com
ORCID: 0009-0001-9344-3390

Денис ДУБОВИК

кандидат технічних наук, IT expert KIVI MEDIA, denic3d@gmail.com
ORCID: 0000-0002-8983-0646

Тетяна ДУБОВИК

старший викладач кафедри спеціалізованих комп'ютерних систем,
Навчально-наукового інституту «Український державний хіміко-технологічний університет»,
Український державний університет науки і технологій, tetyana_dubovik@udhtu.edu.ua
ORCID: 0000-0002-2359-2569

РОЗРОБКА СПЕЦІАЛІЗОВАНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ НА БАЗІ ARDUINO UNO «РОЗУМНА» АВТОПАРКОВКА

Анотація. У даній статті представлено розробку спеціалізованої комп'ютерної системи на базі **Arduino Uno**, яка буде включати кілька важливих компонентів для створення ефективного та зручного інструменту для паркування. Система включатиме веб-сторінку, на якій користувачі зможуть забронювати вільне паркувальне місце. Це дозволить зменшити час на пошук вільного місця, а також знизить ймовірність неправильного паркування.

Метою цього проекту є розробка та впровадження розумного паркомату, який зможе автоматизувати процеси бронювання, оплати та контролю за паркувальними місцями, а також покращити ефективність використання наявних паркувальних ресурсів.

Практичне значення цієї розробки полягає в автоматичному контролі за паркувальними місцями. Для цього використовуються потужні алгоритми машинного навчання з автоматичним розпізнаванням тексту на номерних знаках, що дозволяє швидко та точно ідентифікувати транспортні засоби в реальному часі.

Методологія. На основі технології **TensorFlow** використовується камера, яка здійснюватиме зчитування номерних знаків автомобілів. Завдяки цьому паркомат зможе автоматично фіксувати автомобіль, що зайняв місце, і в реальному часі оновлювати статус доступності місця в системі. Для зберігання та обробки інформації про бронювання, доступність місць та транзакції користувачів буде створена база даних, що забезпечить надійне зберігання даних і дозволить забезпечити зручний доступ до інформації. **Arduino Uno**, завдяки своїй доступності, простоті та широким можливостям для інтеграції з різними датчиками та модулями, є ідеальним вибором для реалізації такого проекту. Інтеграція веб-сторінки для бронювання, розпізнавання номерних знаків за допомогою **TensorFlow** та база даних для обробки інформації дозволить створити ефективну систему, яка значно покращить процес паркування.

Актуальність такої розробки як «розумне» паркування авто, полягає в зростаючому попиті на гнучкі роботизовані систем зі штучним інтелектом, які можна адаптувати під різні сфери застосування.

Ключові слова: комп'ютерний зір, паркувальні місця, **Arduino Uno**, **TensorFlow**, веб-сторінки, розпізнавання авто.

Andrii SERHACHOV, Denys DUBOVYK, Tetiana DUBOVYK. DEVELOPMENT OF A SPECIALIZED COMPUTER SYSTEM BASED ON ARDUINO UNO «SMART» CAR PARKING

Abstract. This article presents the development of a specialized computer system based on **Arduino Uno**, which will include several important components to create an efficient and convenient parking tool. The system will include a web page where users can book a free parking space. This will reduce the time spent searching for a free space, as well as reduce the likelihood of improper parking.

The goal of this project is to develop and implement a smart parking meter that will be able to automate the processes of booking, payment and control of parking spaces, as well as improve the efficiency of the use of available parking resources.

The practical significance of this development lies in the automatic control of parking spaces. To do this, powerful machine learning algorithms with automatic text recognition on license plates are used, allowing for quick and accurate identification of vehicles in real time.

Methodology. Based on **TensorFlow** technology, a camera is used that will read car license plates. Thanks to this, the parking meter will be able to automatically record the car that has taken a seat and update the status of space availability in the system in real time. A database will be created to store and process information about reservations, seat availability and user transactions, which will ensure reliable data storage and provide convenient access to information. **Arduino Uno**, due to its affordability, simplicity and wide possibilities for integration with various sensors and modules, is an ideal choice for the implementation of such a project. The integration of a web page for booking, license plate recognition with **TensorFlow** and a database for processing information will allow you to create an effective system that will significantly improve the parking process.

The relevance of such a development as "smart" car parking lies in the growing demand for flexible robotic systems with artificial intelligence that can be adapted to different applications.

Key words: computer vision, parking spaces, **Arduino Uno**, **TensorFlow**, web pages, car recognition.

Аналіз досліджень та публікацій. В роботі [2] представлені розумні авто парковки. Автор пропонує LoRa система інтелектуальної парковки від українського системного інтегратора iotji своєчасно повідомляє про зайнятість чи вакантність паркувальних місць, спрощує для водіїв навігацію та пошук. В свою чергу, власник розумної парковки отримує ефективну систему моніторингу навантаженості об'єкта, а також – можливість інтегрувати свою систему у мережу розумного міста.

В роботі [5] автор демонструє інноваційну систему паркування без водія Ford показав інноваційну систему паркування без водія, яка полягає в тому, що припаркувати автівку до гаража або на паркувальне місце можна лише за допомогою смартфона. Коли буде потрібно виїхати, автомобіліст дасть команду машині з того ж смартфона, після чого вона під'їде до вказаного місця.

В роботі [11] UI/UX-кейс: автоматизація автопарковки в аеропорту автор пропонує швидке та зручне бронювання паркомісць, доповнює розробку датчиків. Датчик являє собою невеликий пристрій, який кріпиться до підлоги на кожне паркувальне місце. Відповідно, після того, як автомобіль стоїть на місці, система автоматично розуміє, що він зайнятий. Після цього бронювання автоматично переходить в етап «Автомобіль припаркований». Тобто датчик значно спрощує процес прийому та видачі авто.

Виклад основного матеріалу. Розроблений прототип спеціалізованої комп'ютерної системи «розумної» автопарковки оснащений мікроконтролером Arduino UNO [4], сервоприводом, LED – індикатором, датчиком руху, камерою та комп'ютером. Комп'ютер виступає в якості вебсервера, сервера збереження даних та сервера обробки відеопотоку.

Підключення мікроконтролера Arduino UNO з периферійними пристроями відбувається за допомоги з'єднувальних дрітків та макетної плати (рис. 1), подається напруга для компонентів: сервопривід SG90, LED – індикатор TM1637, датчик руху HC-SR501, та підключення сигнального піну 2, 3, 9 відповідно.

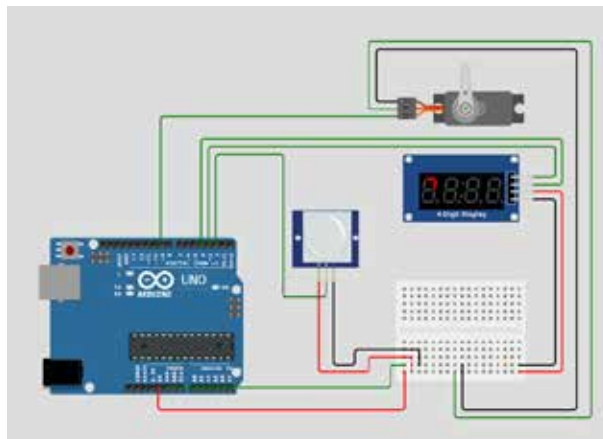


Рис. 1. Загальна схема підключення

Фрагмент коду програми:

Лістинг 1 – Текст файлу «arduino.ino»

```
#include <Servo.h>
#include <TM1637Display.h>
const int pirPin = 2;
const int servoPin = 9;
const int clkPin = 3;
const int dioPin = 4;
Servo myServo;
TM1637Display display(clkPin, dioPin);
void setup() {
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  myServo.attach(servoPin);
  myServo.write(0);
}
```

```

    display.setBrightness(0x0f);
  }
  void loop() {
    int motionDetected = digitalRead(pirPin);
    if (motionDetected == HIGH) {
      Serial.println(1);
      delay(10000);
    }
    if (Serial.available() > 0) {
      int receivedValue = Serial.parseInt();
      if (receivedValue > 0) {
        myServo.write(90);
        delay(10000);
        myServo.write(0);
        int hours = receivedValue / 60;
        int minutes = receivedValue % 60;
        display.showNumberDecEx(hours * 100 + minutes, 0x40, true);
        delay(2000); // Display the time for 2 seconds
      }
    }
  }
}

```

За допомогою Node.js було створено вебсайт [6] (рис. 2), який дозволяє водіям здійснювати бронювання паркомісць. Цей сайт забезпечує зручний інтерфейс, де користувачі можуть швидко знайти вільні паркувальні місця, забронювати місце на необхідний час. Завдяки використанню технології Node.js, система працює ефективно і швидко, забезпечуючи миттєву обробку запитів на бронювання. На сайті водії можуть також перевіряти наявність паркомісць у реальному часі, що дозволяє їм планувати свою поїздку більш зручно і без зайвих непорозумінь. Крім того, сайт дозволяє зберігати історію бронювань, що дає змогу користувачам легко переглядати свої попередні резерви та зручно планувати наступні. Використання Node.js дозволяє масштабувати систему для обробки великої кількості одночасних користувачів, забезпечуючи стабільність і високу швидкість роботи сайту навіть при великих навантаженнях.

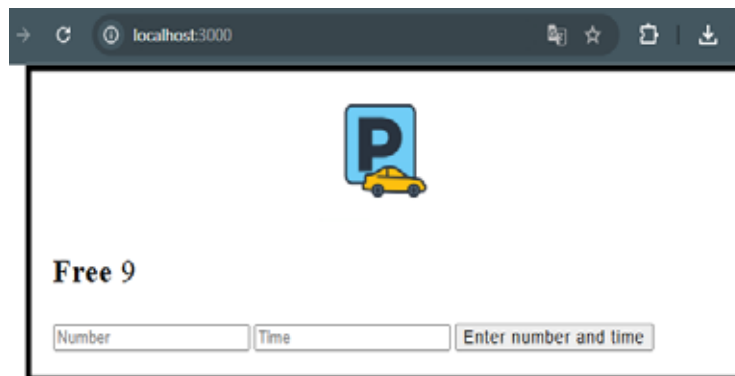


Рис. 2. Вебсайт

Фрагмент коду програми:

Лістинг 2 – Текст файлу «server.js»

```

const express = require('express');
const fs = require('fs');
const path = require('path');
const bodyParser = require('body-parser');
const app = express();
const port = 3000;
app.use(bodyParser.json());

```

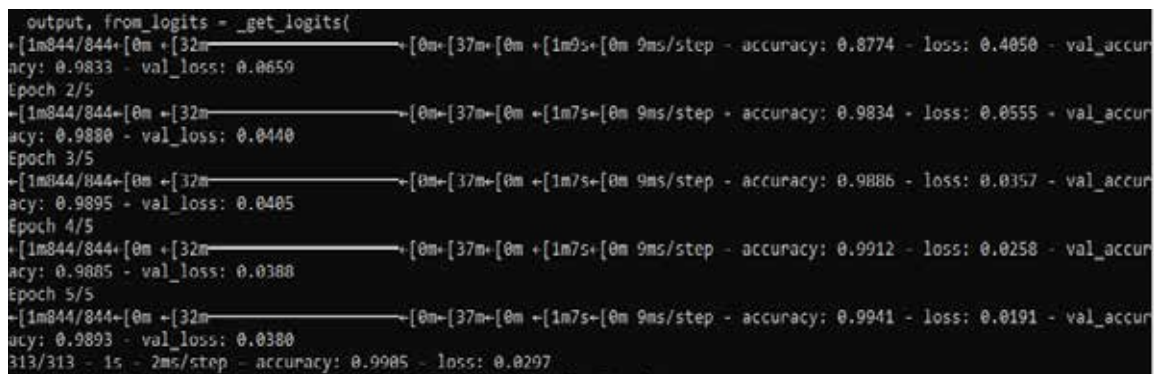
```

app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));
app.use(express.static(path.join(__dirname, 'public')));
app.get('/', (req, res) => {
    fs.readFile(path.join('D:/parking/db/counter.txt'), 'utf8', (err, data) => {
        if (err) {
            res.status(500).send('Error reading file');
            return;
        }
    })
})

```

Було розроблено систему комп'ютерного зору [10] (рис. 3), яка здатна зчитувати номерні знаки автомобілів за допомогою технології TensorFlow. Ця система використовує потужні алгоритми машинного навчання для автоматичного розпізнавання тексту на номерних знаках, що дозволяє швидко та точно ідентифікувати транспортні засоби в реальному часі.

Завдяки використанню TensorFlow, система навчається на великих наборах даних, що дозволяє їй враховувати різні умови освітлення, кути зйомки та варіації в дизайні номерних знаків. Це дозволяє значно підвищити точність і надійність розпізнавання навіть у складних умовах, таких як погана видимість чи високий рівень динаміки.



```

output, from_logits = _get_logits(
+ [1m844/844+ [0m + [32m -----> [0m+ [37m+ [0m + [1m9s+ [0m 9ms/step - accuracy: 0.8774 - loss: 0.4050 - val_accu
ary: 0.9833 - val_loss: 0.0659
Epoch 2/5
+ [1m844/844+ [0m + [32m -----> [0m+ [37m+ [0m + [1m7s+ [0m 9ms/step - accuracy: 0.9834 - loss: 0.0555 - val_accu
ary: 0.9880 - val_loss: 0.0440
Epoch 3/5
+ [1m844/844+ [0m + [32m -----> [0m+ [37m+ [0m + [1m7s+ [0m 9ms/step - accuracy: 0.9886 - loss: 0.0357 - val_accu
ary: 0.9895 - val_loss: 0.0405
Epoch 4/5
+ [1m844/844+ [0m + [32m -----> [0m+ [37m+ [0m + [1m7s+ [0m 9ms/step - accuracy: 0.9912 - loss: 0.0258 - val_accu
ary: 0.9885 - val_loss: 0.0380
Epoch 5/5
+ [1m844/844+ [0m + [32m -----> [0m+ [37m+ [0m + [1m7s+ [0m 9ms/step - accuracy: 0.9941 - loss: 0.0191 - val_accu
ary: 0.9893 - val_loss: 0.0380
313/313 - 1s - 2ms/step - accuracy: 0.9905 - loss: 0.0297

```

Рис. 3. Тренування комп'ютерного зору

Фрагмент коду програми:

Лістинг 3 – Текст файлу «camera.py»

```

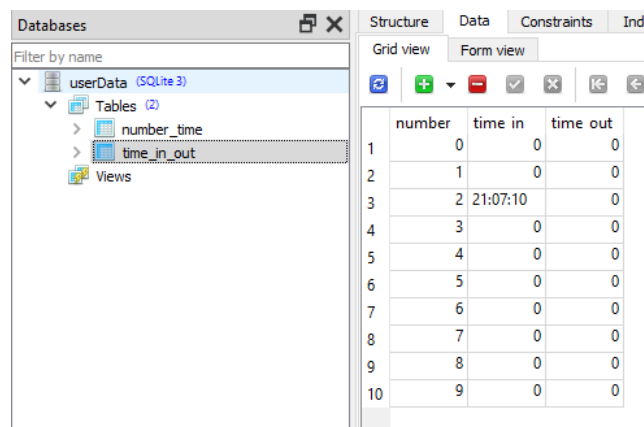
import cv2
import tensorflow as tf
import numpy as np
import os
import matplotlib.pyplot as plt
import re
serial = 'db/serial.txt'
predicted_file = 'db/predicted_number.txt'
signal= 0
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
x_train = x_train.reshape((x_train.shape[0], 28, 28, 1))
x_test = x_test.reshape((x_test.shape[0], 28, 28, 1))
model = tf.keras.models.Sequential([
    tf.keras.layers.Reshape((28, 28, 1), input_shape=(28, 28)), # Reshape for CNN
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),

```

```
tf.keras.layers.Dense(128, activation='relu'),
#tf.keras.layers.Dropout(0.5),
tf.keras.layers.Dense(10, activation='softmax')
])
```

Було розроблено базу даних [9] (рис. 4), в якій зберігається інформація про номер автомобіля, заброньований час, а також історія бронювань. Ця база даних організована таким чином, щоб забезпечити ефективне збереження, оновлення та доступ до даних, що дозволяє швидко обробляти запити від користувачів та системи. Кожен запис у базі містить номер автомобіля, дату і час, коли було здійснено бронювання, а також історичну інформацію про попередні бронювання цього ж автомобіля. Завдяки такій структурі дані про бронювання зберігаються в зручному вигляді, що дозволяє легко відслідковувати всі попередні запити та планувати майбутні резерви.

База даних також дозволяє відстежувати актуальний статус кожного паркомісця в реальному часі, що важливо для управління паркуванням. Інтерфейс системи забезпечує доступ до історії всіх бронювань, дозволяючи адміністраторам або користувачам переглядати та керувати записами, коригувати час бронювання або скасовувати попередні резервації. Завдяки цій базі даних, процес бронювання паркомісць став більш зручним та ефективним, а також забезпечує прозорість і надійність управління системою паркування.



	number	time in	time out
1	0	0	0
2	1	0	0
3	2	21:07:10	0
4	3	0	0
5	4	0	0
6	5	0	0
7	6	0	0
8	7	0	0
9	8	0	0
10	9	0	0

Рис. 4. База даних SQLite

Розроблена основна програма розумної автопарковки, яка сполучає та забезпечує взаємодію між усіма компонентами системи: вебсайтом, комп'ютерним зором, базою даних та Arduino Uno. Ця програма виконує роль центрального елемента, який координує роботу кожного з компонентів, гарантуючи їх ефективну взаємодію. Вебсайт служить інтерфейсом для користувачів, де водії можуть здійснювати бронювання паркомісць. Основна програма обробляє запити, отримані через вебсайт, передає їх до бази даних для збереження інформації про бронювання та забезпечує актуальність статусу паркувальних місць. Комп'ютерний зір, заснований на технології TensorFlow, здійснює розпізнавання номерних знаків автомобілів, що дозволяє ідентифікувати транспортні засоби та перевіряти їх наявність бронювання. Програма обробляє зображення, отримані від камер, і передає результати розпізнавання в базу даних для подальшої обробки та перевірки. База даних зберігає всю інформацію про заброньовані паркомісця, номери автомобілів та історію бронювань, що дає змогу ефективно управляти паркувальним процесом. Водночас, система Arduino Uno відповідає за контроль фізичних паркувальних місць, відкриваючи або блокуючи доступ до них залежно від статусу бронювання. Завдяки такій інтеграції всіх компонентів, програма забезпечує безперебійну роботу розумної парковки, ефективно обробляючи запити користувачів, виконуючи автоматичне розпізнавання транспортних засобів, зберігаючи історію бронювань і забезпечуючи автоматизований доступ до паркувальних місць.

Фрагмент коду програми:

Лістинг 4 – Текст файлу «userData.db»

```
import sqlite3
from datetime import datetime
import re
```

```

import time
import numpy as np
import os
import serial
ser = serial.Serial('COM4', 9600)
conn = sqlite3.connect('db/userData.db')
cursor = conn.cursor()
output = 'db/output.txt'
counter = 'db/counter.txt'
predicted_file = 'db/predicted_number.txt'
serial1 = 'db/serial.txt'
def create_table():
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS number_time (
            number INTEGER,
            time INTEGER
        )
    ''')

```

Розпізнавання номеру здійснюється за допомогою моделі глибокого навчання, тренованої на даних з набору MNIST [3].

Завантаження та підготовка даних включає використання набору даних MNIST, який містить зображення цифр від 0 до 9. Зображення розміром 28x28 пікселів нормалізуються, приводячи значення пікселів до діапазону [0, 1]. Дані перетворюються в 4D формат з формою (num_samples, 28, 28, 1), де num_samples – кількість зразків, 28x28 – розміри зображень, 1 – кількість каналів (чорно-білі зображення).

Моделі нейронної мережі складається з кількох типів шарів: конволюційних шарів (Conv2D), що дозволяють витягувати ознаки з зображень, шарів зниження розміру (MaxPooling2D), що зменшують розміри зображень, шарів для перетворення векторів (Flatten), що перетворюють багатовимірні дані у 1D вектор, і повнозв'язних шарів (Dense), що виконують класифікацію на основі отриманих ознак. Моделі компілюється за допомогою оптимізатора Adam та функції втрат SparseCategoricalCrossentropy, яка підходить для багатокласової класифікації.

Моделі тренується на даних x_train та y_train, де x_train – зображення, а y_train – відповідні етикетки (цифри). Процес навчання триває протягом 5 епох з розміром пакета 64.

Для розпізнавання цифри в реальному часі використовується відеопотік з камери. Кожен кадр обробляється, на ньому зображається прямокутник, що обмежує область для розпізнавання цифри. Кадр обрізається, конвертується у відтінки сірого та змінюється до розміру 28x28 пікселів. Зображення інвертується та передається в модель для прогнозування [1].

Моделі визначає ймовірності для кожної цифри (від 0 до 9) та виводить на екран цифру з найбільшою ймовірністю (рис. 5).



Рис. 5. Тестування розробки

Результати натуральних іспитів. Підключений до мікроконтролера датчик руху виявляє зміну положення об'єктів у своєму полі зору, після чого мікроконтролер Arduino Uno передає сигнал на сервер. Отримавши сигнал основна програма на сервері запускає програму розпізнавання цифр, результат порівнює з базою даних, що містить дані, отримані з вебсайту. У випадку, якщо в базі даних знайдена відповідність між розпізнаною цифрою та спланим часом, комп'ютер передає сигнал до мікроконтролера Arduino Uno, який в свою чергу відображає інформацію про залишковий час на LED-індикаторі та активує сервопривід. Тестування системи здійснювалося за різних умов, результати якості розпізнавання цифри (0-9) представлені в табл. 1.

Таблиця 1

Точність розпізнавання цифри (0-9)

Освітлення	Положення	Якість зображення	Точність (%)
Нормальне	Центр	Висока	98-100
Темне	Центр	Висока	85-90
Яскраве	Центр	Висока	95-98
Нормальне	Верхній правий кут	Висока	85-90
Нормальне	Лівий нижній кут	Висока	85-90
Нормальне	Центр (обертання 45°)	Висока	80-85
Нормальне	Центр	Низька (шум)	70-75
Темне	Центр	Низька	60-70
Нормальне	Центр (обертання 90°)	Висока	70-80

Висновки. Проведено всебічний аналіз теми розробки спеціалізованої комп'ютерної системи на базі Arduino Uno розумна автопарковка: визначено технічні та функціональні вимоги до розумного паркомату та описано можливості використання платформи Arduino Uno для реалізації проекту [7-8].

Проектування апаратної частини: розроблено схему підключення компонентів, включаючи камеру для зчитування номерів автомобілів; спроектовано систему для зчитування номерних знаків за допомогою TensorFlow. Розроблено програмне забезпечення для Arduino Uno (для обробки сигналів від датчиків і камери). Реалізовано функціонал для зчитування даних про наявність паркувальних місць. Розроблено взаємодію між Arduino та іншими модулями системи.

Розробка веб-сторінки для бронювання паркувальних місць:

- Створено веб-інтерфейс для користувачів, на якому вони зможуть бронювати вільні паркувальні місця.

- Реалізовано функціонал для перегляду статусу доступних місць у реальному часі.

- Забезпечено інтеграцію веб-сторінки з базою даних для зберігання інформації про бронювання.

Розпізнавання номерних знаків за допомогою TensorFlow:

- Навчено модель на основі TensorFlow для розпізнавання номерних знаків.

- Інтегровано модель у систему для автоматичного визначення автомобілів, що паркуються, та оновлення статусу місць.

Розробка та інтеграція бази даних: створено базу даних для зберігання інформації про користувачів, паркувальні місця, бронювання та транзакції; реалізовано механізм взаємодії між базою даних і іншими компонентами системи.

Виконання цих завдань дозволило створити повноцінну автоматизовану систему паркування, яка значно спрощує процес бронювання та контролю за паркувальними місцями, підвищує ефективність використання інфраструктури та зручність для користувачів.

Дана розробка є прикладом того, як можна поєднати різні технології у єдиній теоретичній моделі. Наголошено на важливості теоретичного підходу в розробці інноваційних рішень, які можуть бути реалізовані в практичному контексті в майбутньому.

Перспективи. В даний час розроблена система актуальна, та має перспективу на модернізацію та удосконалення. Може бути використана в різних сферах, де затребувана роботизація виробництва – автопаркування, тощо.

Список використаних джерел:

1. Дубовик Т. Комп'ютерна система розпізнавання облич (2025). Agriculture, technical and applied sciences: main problems and solutions Collective monograph. Published by Primedia eLaunch, Boston 2025, С. 305–324
2. Розумні авто парковки: URL: <https://iotji.io/ru/solutions/umnuye-parkovki/> (дата звернення 20.01.2025)

3. Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Zheng X. TensorFlow: Large-scale machine learning on heterogeneous systems. arXiv preprint arXiv:1603.04467. 2016. URL: <https://arxiv.org/abs/1603.04467> (дата звернення 21.01.2025)
4. Arduino, Inc. (2021). Arduino UNO Rev3. URL: <https://www.docs.arduino.cc/hardware/uno-rev3/> (дата звернення 21.01.2025)
5. Ford показав інноваційну систему паркування без водія : URL: <https://tsn.ua/auto/news/ford-pokazav-innovaciynu-sistemu-parkuvannya-bez-vodiya-video-1616704.html> (дата звернення 21.01.2025)
6. Node.js. (n.d.). Node.js®: JavaScript runtime built on Chrome's V8 JavaScript engine: URL: <https://nodejs.org/> (дата звернення 21.01.2025)
7. Sergeyeva O., Lisenko V., Dubovik T., Patalakha M. "Development of a Wi-Fi controlled mobile video device on the Arduino NANO basis", *Eastern-European Journal of Enterprise Technologies*, 2020. Vol. 3/9 (105), 55–60, DOI: <https://doi.org/10.15587/1729-4061.2020.206558> (дата звернення 21.01.2025)
8. Sergeyeva O. V., Robuck E. A., Pavlenko N. Y., Dubovik T. N. Applications of computer systems based on the Arduino microprocessor system in chemical laboratory. The Eurasian Union of Scientists (ESU), 2019. 12 (69 (5)), 34–38.
9. SQLite. (n.d.). SQLite Documentation: URL: <https://www.sqlite.org/docs.html> (дата звернення 21.01.2025)
10. TensorFlow. (n.d.). TensorFlow: An Open Source Machine Learning Framework for Everyone: URL: <https://www.tensorflow.org/> (дата звернення 21.01.2025)
11. UI/UX-кейс: автоматизація парковок в аеропорту: URL: <https://ux.pub/editorial/uiux-kieis-avtomatizatsiia-parkovok-v-aeroportu-5dgh> (дата звернення 21.01.2025)