

# Uroš Vojičić

BSc in Electrical and Computer Engineering

[LinkedIn](#) | +381-63-8069068 | [vojicicuros@gmail.com](mailto:vojicicuros@gmail.com) | [GitHub](#)



## Education

### School of Electrical Engineering, University of Belgrade

- Bsc of Electrical and Computer Engineering
- Department of Signals and Systems

## Skills

- Python | C++ | C | Matlab/Octave
- Object-Oriented Programming (OOP) | Algorithms and Data Structures
- Linux
- Probability and Statistics | Signals and Systems
- Digital image processing | Digital audio processing

## Experience

### Rivian & Volkswagen Group Technologies - Software Engineer Intern (6 months - fulltime)

- Created a complete automation testing tool from scratch as part of an individual internship project, used to validate vehicle state prior to OTA firmware updates.
- Built a robust pre-check script using **Object-oriented Python**, incorporating **Threading** for concurrent operations and integrating **Linux system administration** tasks
- Worked with **TCP/IP, UDS, and SSH protocols** to interface with embedded systems and perform system-level validations on in-vehicle ECUs.
- Managed the preparation and **configuration of Linux-based environments** on test vehicles, ensuring compatibility and stability prior to firmware updates.
- Successfully integrated the script into **GitLab CI/CD pipelines** as a required preliminary stage, enabling safer and more automated firmware rollouts.
- The solution was adopted into the production firmware testing workflow, contributing to improved reliability and **automation of the OTA (Over-The-Air) update process**.
- Performed hands-on testing of nightly and monthly firmware builds on real vehicles, analyzing logs, validating ECU behavior, and documenting results using TestRail dashboards.

## Projects

### • Real-time Eye Gaze Tracking using Webcam: (Python, OpenCV, MediaPipe, scikit-learn, NumPy, Multithreading, Pygame, JSON)

Developed a **real-time eye gaze tracking** application using **threaded Python**.

It uses **OpenCV** and **MediaPipe** for face detection, image processing techniques for eye landmarks detection, and algorithms/ ml models for estimating gaze direction based on eye movement. Calibration is performed for improving the accuracy of gaze estimation during runtime, where users provide gaze data to calibrate the system. The estimated gaze direction is then visualized on the webcam stream or a separate display window.

### • Public Transport Network - Optimal routes: (C++)

Object-oriented programming principles in **C++** for creating city transport network simulation imported through a txt file; Objective is to identify optimal routes for traveling between stations. These routes are determined by considering both the times of arrival of buses and the frequency of buses at certain stations. The implementation involves three key algorithms: **Breadth-First Search (BFS)**, **Depth-First Search (DFS)**, and **Dijkstra's algorithm**.

### • Liquid Sorting Puzzle Game: (C++, OOP, Tree traversal Algorithms)

Developed a Liquid Sort Puzzle game using Object-oriented programming techniques in **C++**.

The starting state is generated using a Lagged Fibonacci generator. A decision tree is created to map out possible player moves, which the program processes and iterates through. Additionally, the program offers a hint feature for the player's next move.

### • Simpsons Character Classifier using CNN: (Python, Scikit-Learn, Tensorflow)

This project aims to classify characters from "The Simpsons" using **Deep Learning**. It involves importing and analyzing a dataset of character images, setting up directories for training and validation data, and utilizing **Convolutional Neural Networks (CNNs)** for classification.

- **Digital Image Processing projects:** (Python, MATLAB, OpenCV, scikit-image, NumPy, SciPy, PIL)

- Implemented global and local histogram equalization from scratch, using custom and built-in functions.
- Applied spatial filters (mean, median, Wiener) and frequency-domain filtering (e.g. for motion blur and periodic noise removal).
- Performed edge detection (Sobel, Canny), segmentation (thresholding, clustering), and PSNR/MSE evaluation.
- Simulated image degradation (salt & pepper, blur), and applied restoration techniques.
- Developed and visualized LDR→HDR conversion using inverse tone mapping and evaluated dynamic range.
- Created artificial grayscale images and applied neighborhood-based median filtering.
- Implemented background blurring with real-time webcam feed and OpenCV.
- Trained YOLO models for object detection
- Built deep learning models (e.g. CNNs) for image classification; implemented training and evaluation pipeline.
- Etc.

- **Audio Signal Processing projects:**

- Acoustic phonetics and the physiology of the speech and auditory system
- Speech generation theory and speech perception (including masked speech perception and perception models)
- Speech intelligibility and quality assessment
- Digital speech signal processing techniques
- Speech synthesis and automatic speech recognition
- Speaker recognition and voice biometrics
- Evaluation of speech communication quality and methods for quality improvement
- Application of modern speech technologies in real-world systems

## **Engagements**

---

- **Student Mentorship Programme:** As an older student I've had the opportunity to guide and support a group of students from my high school in their journey towards enrolling in our faculty. I have assisted them in gathering necessary information and materials, as well as providing guidance and support to help them navigate through the freshmen year.

## **Interests**

---

- Brazilian Jiu-Jitsu; Fishing; DIY embedded and Python projects;