Mustafa Yalçın

İstanbul, Turkey · mustafayalcin.my@outlook.com
linkedin.com/in/mustafayalcinmy · github.com/mustafayalcinmy

Experience

PHITECH Bioinformatics — Kocaeli - Marmara Teknokent
Software Engineer — Aug 2024 - Mar 2025

- I have experience in developing tools using Python and R, focusing on transforming them into microservices and integrating them with backend services. Throughout my work, I have designed and implemented user interfaces, ensuring a seamless integration process within projects.

  Additionally, I have containerized applications using Docker, making deployment and scalability more efficient. I have also worked on converting tools into libraries, enhancing modularity and reusability. My proficiency in GitHub allows me to effectively manage repositories, version control, and collaborate efficiently in software development projects.(Python, R, Docker, MySQL, MongoDB, GitHub)

SOFTERIA Software — Kocaeli - Marmara Teknokent
Intern — July 2024 - Aug 2024

- Experienced in Data Analysis and Manipulation, Developing Tools, and Building Microservices (Python, FastAPI, TypeScript, Angular, Docker)

PHITECH Bioinformatics — Kocaeli - Marmara Teknokent
Software Engineer — Aug 2023 - July 2024

- Experience in Web Frontend development using Angular and TypeScript (TypeScript, Angular, Docker, MySQL, MongoDB)

Intern — July 2023 - Aug 2023

- During my extended internship as a Full-Stack developer, I utilized Angular for the frontend and JavaSpring for the backend(Java, TypeScript, Angular, SpringBoot, Docker, MySQL, MongoDB)

Software Engineer — Sep 2022 - July 2023

- Part-time work with experience of Data Processing (Python, Pandas, RDKit, Tkinter)

Volunteer Intern — Jun 2022 - Aug 2022

- Molecular Data Analyzing (Python, OpenPyXL, RDKit, Tkinter)

Education

Kocaeli University — Kocaeli
Undergraduate Software Engineering *GPA: 3.04 (4th Grade)* — 2021 - Present

Technical Skills

Programming Languages

- Python, Angular, Java, TypeScript, R, GO, C C#

Web Development

- **Backend:** FastAPI, Spring Boot, Django, Gin, .NET, RESTful API design
- **Frontend:** Angular, JavaScript, HTML/CSS
- **Authentication:** JWT, OAuth, OpenID

Data & Infrastructure

- **Databases:** PostgreSQL, MongoDB, MySQL, SQLite
- **Message Queuing:** RabbitMQ
- **Containerization:** Docker

Data Science & AI

- **Machine Learning:** TensorFlow, PyTorch
- **Data Analysis:** Pandas, NumPy, Matplotlib, OpenPyXL, RDKit, Tkinter
- **Computer Vision:** OpenCV

Development Tools

- **Version Control:** Git
- **Documentation:** Swagger

Projects

DiOne Docs *Angular, Go*

DiOne Docs is a web-based document editor inspired by Google Docs, designed to support both real-time collaboration and offline document editing. The project enables multiple users to edit documents simultaneously while also allowing offline modifications with automatic synchronization upon reconnection. It supports standard DOC format as well as a custom KD format and provides export options in both KD and PDF formats.

- Developed the frontend using Angular and the backend using Go (Golang), ensuring a high-performance and scalable architecture.
- Implemented PostgreSQL as the database system for secure and efficient data storage.
- Integrated Gorilla WebSocket to enable real-time document updates and user collaboration.
- Designed and implemented offline functionality using IndexedDB and Service Workers, ensuring a seamless editing experience.
- Containerized the application with Docker for easy deployment and scalability.

MIDI Music Generation using AI *PyTorch, Tensorflow, LSTM, RNN, Transformer*

This project focuses on AI-driven music composition by leveraging deep learning techniques, particularly Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models. The system is trained on the Maestro dataset, which consists of over 200 hours of professional piano performances in MIDI format.

- Developed a music generation model using LSTM and Transformer architectures to create stylistically diverse MIDI compositions.
- Implemented polyphonic music generation, capturing complex note relationships and expressive dynamics.
- Explored various MIDI data representations to enhance musical structure learning.
- Designed evaluation methods for both objective metrics (harmonic consistency, note patterns) and subjective assessments (musicality, emotional expressiveness).
- Investigated controllable music generation, allowing users to influence the style and mood of generated compositions.

Real-Time Motion Detection System *Python, OpenCV*

This project implements a real-time motion detection system using Python and OpenCV to detect moving objects captured by a camera and mark them with rectangular frames. The system utilizes image processing techniques for motion detection and applies merging algorithms to optimize the detected motion areas. This technology is similar to memory-based motion detection used in platforms like Apple FaceTime and Google Meet, where real-time video streams are analyzed to detect movement and enhance user experience.

- Developed a motion detection system using Python and OpenCV for real-time object tracking.

- Utilized frame differencing to detect moving regions and applied thresholding to generate binary images for analysis.
- Implemented contour detection to identify object boundaries and filter out irrelevant movements based on size.
- Designed a custom box merging algorithm to combine nearby or overlapping bounding boxes for more accurate motion detection.
- Achieved real-time performance with an average frame processing time of 30 ms under varying lighting conditions.

Amazing DQN Game *Deep Q-Network (DQN), Pygame, Gym*
This project implements a Deep Q-Network (DQN) to train an agent for a 2D obstacle avoidance game. The environment is built using Pygame and Gym, and the agent learns to navigate through gaps to maximize its score.

- Developed a custom game environment with Pygame and Gym, defining states, actions, and rewards.
- Implemented a reinforcement learning model using Deep Q-Network (DQN) to train an autonomous agent.
- Designed an efficient reward structure to encourage learning while preventing undesirable behaviors.
- Enabled real-time rendering for visualizing training progress and debugging.
- Developed training and testing scripts, allowing users to train the model and replay learned policies.

Blockly Based Image Processing *Python*
Our approved project, funded by TÜBİTAK 2209A, is designed for individuals and children with an interest in software development but lacking experience and knowledge. The project aims to facilitate users' understanding of image processing and robotics concepts through a Blockly (Drag and Drop) structure. Users can easily comprehend the purpose of each parameter by modifying the parameters of image processing functions. Additionally, they can set the robot in motion using robot blocks, capture real-time images from the robot's camera, and process these images. I am leading this project collaboratively with my team members.

Robot Detection with Image Processing *Python*
Utilizing the Python language and the OpenCV library, two distinct applications were developed for corner and edge detection within a specified dataset. To enhance the success rate in the datasets, significant improvements were made during the preprocessing stage, resulting in the development of an application specifically designed for robot detection.

Restaurant Automation Application *C#*
A restaurant automation application implemented in C# allows the manager to seat a customer at an empty table. When the waiter takes the customer's order, the table number and order details are sent to the kitchen. Once the order is prepared, the kitchen notifies the waiter. After receiving payment at the cash register, the customer leaves, and the table returns to an empty state.

OBS Project *C*
Student information system. It is a struct-based structure that can keep data without copying by using a pointer. It saves data to local files and reads it back.

Turkish Grammar Detection *C*
It is checked whether the entered text complies with the spelling rules. Words or phrases in accordance with Turkish grammar rules are listed.