Zahin M. Zaman

✉ zm2zaman@uwaterloo.ca

☎ (519) 721-2837

linkedin.com/in/zahin-zaman

zahin-zaman.vercel.app

github.com/alvii147

devpost.com/alvii147

PROJECTS

Hachiko's Journal

HackRU – **1st Place Health Track**

- AI-based digital therapeutic journal writing application for mental health patients with interactive virtual assistant
- Performed **sentiment analysis** using **Google Cloud Language** to provide real-time feedback and compliments
- Implemented multi-threading in Python to accommodate **NLP** computations while running frontend

cram.ai

WinterHacklympics – **Best Use of Google Cloud**

- Web platform that uses natural language processing to analyze lecture videos and generate learning flashcards
- Incorporated **NLTK** and **Google Cloud Language** to summarize lecture videos
- Generated flashcards with questions and answers in a user-friendly frontend served by a **Flask** backend hosted on **Heroku**

Image Noise Interpolation

- Applied methods described in research paper to detect salt-and-pepper noise in colored images and retrieve original pixels
- Outlined functions for manipulation of **NumPy** arrays and used **Matplotlib** color maps for visualization
- Performed **normalized mean-squared error** as performance metric to measure effectiveness of implemented method

pupil

HackDuke – **Wolfram Award**

- Computer vision application that detects pupil movement and develops eye-tracking communication for Cerebral Palsy patients
- Applied **Haar Cascade classifiers, blob detection,** & **morphological transforms** in **OpenCV** to process images in real-time
- Utilized **multi-state sigmoid activation** function to calibrate pupil coordinates

TECHNICAL SKILLS

Languages: Python, C++, Go, JavaScript, HTML, CSS, Bash

Frameworks: Django, NumPy, pandas, Matplotlib, TensorFlow, Keras, scikit-learn, OpenCV, MySQL, PostgreSQL, React.js

Tools/Services: Docker, Kubernetes, AWS, Google Cloud, Jupyter

EDUCATION

University of Waterloo

B.A.Sc. in Electrical Engineering, 4th Year | Sept 2018 – May 2023

- **Term Dean's Honour List,** for outstanding academic performance
- **Artificial Intelligence Degree Specialization**

EXPERIENCE

Open-Source Software Developer

codePrentice | Sept 2020 – Present

- Expanded Python multiparty-computation library, **MP-SPDZ,** to support **CNNs** including SqueezeNet, ResNet, and DenseNet
- Composed comprehensive tutorial based on Matrix Profile research paper for Python time series analysis library, **STUMPY**
- Optimized cache utilization in STUMPY's multi-threaded Matrix Profile computation by implementing tiling scheme algorithm using **NumPy** arrays and **Numba** just-in-time compilation

Full-Stack Developer

Prodigy Education | Jan 2022 – April 2022

- Maintained **OAuth 2.0** & **OpenID Connect** identity service built on **Ruby on Rails**
- Leveraged multi-staged **Docker** builds to optimize production server container hosted on **Amazon ECS**, reducing image size by **58.4%**
- Designed & documented **Apache Kafka** calls to stream backend server events for data tracking and verified data payloads with unit tests utilizing **RSpec** & **Minitest** on Rails

Full-Stack Developer

Nokia | May 2021 – August 2021

- Developed & managed authentication security, community articles page, and voucher redemption system for Nokia Network Developer Portal on **Django** server with an **Azure MySQL** database
- Secured backend using **Django REST framework** API permissions, cross site scripting protection, and honeypot setups
- Composed unit tests for Django forms, models, and API endpoints, and configured **Docker** image for **Gitlab CI** automation testing

Software Developer

Wind River Systems | Sept 2020 – Dec 2020

- Devised thread-safe functions in **C** and inline **Assembly** to fix multi-threading and memory-based defects for VxWorks RTOS
- Developed interactive application in **PyQt5** that assists in writing git commit messages and verifies status of Jira issues and code reviews

Display Verification Engineer

Qualcomm | Jan 2020 – May 2020

- Attained **20%** increase in functional coverages by engineering **SystemVerilog** assertions to verify processor design
- Automated formal verification using **Perl** scripting to extract design hierarchy and formulate assertions