

Amlesh Sivanantham

CONTACT INFORMATION

WEBSITE: <http://zamlz.org>
EMAIL: zamlz@pm.me
PHONE: (408) 219-6474

SKILLS

LANGUAGES: Python, C++, C, Java, JavaScript, Scheme, L^AT_EX, Bash, Markdown, Verilog
LIBRARIES: TensorFlow, OpenAI Gym, PyTorch, NumPy, SciPy, Matplotlib, OpenCV
TECHNOLOGIES: Git, Vim, Linux

EDUCATION

08.2017 - 05.2019 UNIVERSITY OF SOUTHERN CALIFORNIA
M.S. in Computer Science (GPA: 3.78)
Concentration: Intelligent Robotics

09.2013 - 06.2017 UNIVERSITY OF CALIFORNIA, SANTA CRUZ
B.S. in Computer Engineering (with honors)
B.S. in Computer Science
Thesis: *Detecting Anomalies in Time-Series Data using Long Short-Term Memory Networks* - Advisor: Dr. Patrick Mantey

WORK EXPERIENCE

09.2017 - PRESENT **Graduate Research Assistant**
University of Southern California - Robotic Embedded Systems Laboratory
Perform graduate research in Deep Reinforcement Learning and its application to Robotics. Working for a PhD student on research problems related to perception and navigation with Deep Reinforcement Learning.

09.2016 - 06.2017 **Undergraduate Research Assistant**
University of California, Santa Cruz - Jack Baskin School of Engineering
Performed undergraduate research in Machine Learning and Deep Learning for the Smart Energy Analytic Disaggregation System project for Dr. Ali Adabi to explore methods to analyze and identify time-series data.

PROJECTS

QUADCOPTER REINFORCEMENT LEARNING AGENT *(USC R.E.S.L.)*

Working on building a Reinforcement Learning agent that is capable of navigating quickly through a cluttered environment. Train the agent in simulation, and transfer the learnt policy to a real-world quadcopter. *In Progress*

IMAGINATION AUGMENTED AGENTS FOR RUBIK'S CUBES *(Jeju DL Camp 2018, S. Korea)*

Participated in the Jeju Deep Learning Camp where I worked to implement the paper *Imagination Augmented Agents for Deep Reinforcement Learning* on a self-made Rubik's Cube environment

DEEP Q-LEARNING WITH STRUCTURE2VEC FOR SOLVING THE VEHICLE ROUTING PROBLEM *(USC - Topics in Discrete Optimization and Learning)*

Worked to adapt previous work done by *Hanjun et al.* that utilized Q-learning and the Structure2Vec graph embedding to solve discrete optimization problems. We adapted the approach to work with the domain of the Vehicle Routing Problem.

ANOMALY DETECTION IN TIME SERIES DATA *(Undergraduate Senior Thesis)*

Researched Deep Learning and implemented a long short-term memory network that identifies if a given subsequence of a particular time-series system is anomalous or not. The dataset that I worked with was provided by my faculty advisor which corresponds to the energy usage of an electric meter on the circuit that provides power to the water pump. After training, the network was able to identify anomalies with an accuracy of 90%.

WHO'S LAZY? NOT EYE *(Hack UCSC 2017)*

A vision therapy program for people with lazy eye using a standard webcam. The app uses the webcam to constantly analyze the user's eyes and notifies them when their eyes drift away. Particularly, the client can pause any media application that the user has playing in the background and will only let them resume their application once they have focused their eyes. I worked on the algorithm that located the position of the pupils using machine learning and identified whether the pupils correlated with lazy eyes or not based on that position.

OTHELLO (REVERSI) FOR THE PSoC 5LP *(Project for Microcontroller System Design Class)*

Implemented Othello using C and the Cypress API for the Othello PSoC 5LP. It was built using all the different concepts learned in class. Refer to the report in the documentation directory in the repository for more details.

UCSC PLAZA *(Project for Software Engineering Class)*

UCSC Plaza is an event manager designed for members of UCSC. This is simply a prototype. Worked with JavaScript and JQuery to make the interface which connects the front-end to the web-server. Project was developed using Scrum practices.

HUMMUSLITE *(Project on Logic Design)*

A simple project I worked on over Summer of 2016. Goal was to build a simple CPU in Minecraft. The CPU supports 16 instructions and has a program data size of 256 bytes. Each instruction is a byte. Programming for the CPU requires using a punch board system that was also built from scratch in Minecraft.

RESEARCH INTERESTS AND HOBBIES

Enjoy researching about Reinforcement Learning and its applications in Robotics and Competitive Games. Also a Linux enthusiast currently learning/using Gentoo Linux. Amateur Competitive Super Smash Bros. Melee Player. Also enjoy playing the violin and speedcubing.