Vasudev Vikram

https://vasumv.github.io

Email: vasumvikram@gmail.com
Github: vasumv

Education

University of California Berkeley Berkeley, CA
B.S. in Electrical Engineering and Computer Science *Aug. 2016 – May. 2020*

- Graduated with *High Honors*

Carnegie Mellon University Pittsburgh, PA
PhD in Software Engineering *Aug. 2021 – present*

- GPA: 4.0

Publications

Pinning is Sinning: Towards Upgrading Maven Dependencies.
Vasudev Vikram, Yuvraj Agarwal, Rohan Padhye.
Preprint 2023. (link)

Can Large Language Models Write Good Property-Based Tests?.
Vasudev Vikram, Caroline Lemieux, Rohan Padhye.
Preprint 2023. (link)

Guiding Greybox Fuzzing with Mutation Testing.
Vasudev Vikram, Isabella Laybourn, Ao Li, Nicole Nair, Kelton OBrien, Rafaello Sanna, Rohan Padhye.
ISSTA 2023 - *Distinguished Paper Award.* (link)

Growing A Test Corpus with Bonsai Fuzzing.
Vasudev Vikram, Rohan Padhye, Koushik Sen.
ICSE 2021. (link)

Experience

Cruise Automation San Francisco, CA
Software Engineer *August 2020 – September 2021*

- Main engineer leading traffic light detection projects
- Built tools for traffic lights evaluation framework and improving triaging process
- Trained and deployed new flashing emergency vehicle detector for the Scene Understanding team

Cruise Automation San Francisco, CA
Computer Vision Software Engineering Intern *Summer 2019*

- Implemented a system that detects whether objects are occluding traffic light bulb detections using semantic segmentation, visual detections, and LIDAR camera projections as data sources, which was pushed to production on the car. On the road, this was shown to solve the traffic light occlusion use cases while showing no regressions.
- Designed and implemented an ML pipeline for flashing traffic light detection, including scripts for data extraction, HTML / JS labeling and visualization tools, and experiments with various machine learning methods (Bayesian modeling, HMM, RNN). These methods showed significant reductions in flashing light detection latency and an increase in recall compared to the baseline.

Cruise Automation San Francisco, CA
Computer Vision Software Engineering Intern *Summer 2018*

- Architected and built a full ML pipeline for an RNN detector for flashing emergency vehicles, improving over the original model in precision and recall. Utilized CNNs and LSTMs for the model.
- Integrated the model into the Cruise emergency lights package using Tensorflow C++ API

- Generated precision-recall curves and used a Javascript SVG visualization tool for EMV sequences

Cubic Global Defense San Diego, CA
Software Engineering Intern *Summer 2017*

- Implemented and deployed a Node JS/Express website that tracks soldier vests on Google Maps
- Added a Java API (using websockets) to integrate with the existing training app
- Presented to CEO and President of division

University of San Diego, California San Diego, CA
Research Intern *Summer 2016*

- Tested a novel method of using saliency maps of images as an attention model to improve classification on the North American birds dataset
- Sampled high intensity image points from the saliency maps with Python/Numpy for patch extraction, trained and fine-tuned a convolutional neural network (VGG16) in Tensorflow

University of San Diego, California San Diego, CA
Research Intern *Summer 2015*

- Natural language processing project to determine the amount of cultural bias in trivia game questions - used optical character recognition and web scraping in Python to create large database of Quizbowl, Jeopardy, NAQT and other trivia games question
- Calculated word frequencies and ran Latent Dirichlet Allocation on the corpus to find distances between datasets and traditional curricula datasets

San Diego Supercomputer Center San Diego, CA
Software Engineering Intern *Summer 2014*

- Worked in the San Diego Supercomputer Center under Dr. Amit Majumdar, used HTML, Python, and MYSQL to implement an email based form on the NSG Portal

Research

Carnegie Mellon University Pittsburgh, PA
Advisor - Rohan Padhye *August 2021 – present*

Currently working on using large language models to synthesize property-based tests for Python APIs. Also working on dependency analysis techniques to detect breaking changes in libraries and encourage developers to upgrade vulnerable dependencies. Proposed Mu2: a technique to guide greybox fuzzing with mutation testing to automatically generate a set of robust test inputs with high mutation score, winning Distinguished Paper Award at ISSTA 2023. (link)

University of California Berkeley Berkeley, CA
Advisor - Koushik Sen *January 2020 – August 2020*

Worked under Prof. Koushik Sen and Prof. Rohan Padhye. Proposed Bonsai Fuzzing: a novel technique using fuzz-testing to automatically generate comprehensive and readable test inputs for programs such as compilers. Able to significantly improve upon the prior state-of-the-art solution of using fuzzing and test-case reduction tools. Accepted at ICSE 2021 (link)

Teaching

University of California Berkeley
CS 168 - Internet Architectures and Protocols

- *Spring 2020 (Sylvia Ratnasamy and Murphy McCauly)* - Undergraduate Student Instructor; lead and taught discussion sections; contributed to weekly discussion worksheets; helped write a midterm; held weekly office hours; contributed to two student projects

CS 61A - Structure and Interpretation of Computer Programming

- *Fall 2017 (John Denero)* - Lab Assistant; helped tutor students on assignments and labs

Projects

- Member of **HKN** (EECS Honors Society)
- **Won Calhacks 2016 Best Hardware Project and Best 3d Printed Project (11/2016)** with Take a Picasso, a robotic sketch artist that draws a sketch on a piece of paper from a provided photo
- **Won HackJam 2016 (10/2016)** with Apple Mouse, a program that used computer vision to control the movement and clicking of a mouse with an apple
- **2nd Place at Greater San Diego Science and Engineering Fair (03/2015)** with Pokemon AI: a Python bot that uses minimax and machine learning to play Pokemon Showdown online, a two player adversarial game with teams of six Pokemon and movesets hidden to the opponent. A naive Bayesian classifier was used to predict opponent movesets so that minimax could properly be applied.