Phone: (+1)949-546-5905
Email: davechirav@gmail.com

CHIRAV DAVE
Kirkland, WA

Linkedin Github
Website

EDUCATION

Masters, Computer Science, Arizona State University | GPA 3.92 *August 2017 - December 2019*

Bachelor of Technology, Computer Science & Engineering, India | GPA 8.67/10 *July 2012 - July 2016*

TECHNICAL SKILLS & COURSEWORK

Languages:	Python, Java, C, C++, PDDL, SQL, Html, JavaScript
Databases:	MySQL, SQLite
ML Technologies:	Numpy, OpenCV, Pandas, Scikit-learn, Matplotlib, Tensorfow
Coursework:	Foundations of Algorithm, Artificial Intelligence, Perception in Robotics, Intelligent & Assistive Robotics, Statistical Machine Learning

WORK EXPERIENCE

Software Developer | Amazon | Seattle, Washington *Feb 2020*

- Helping good sellers to get paid quickly and efficiently, and preventing bad actors from profiting over the Amazon platform.
- Analysing the risk inherent in a seller's transactions to figure out how much money should be reserved to mitigate any risk.

Technology Stack: Java, Apollo, Brazil

Graduate Researcher | Arizona State University | Tempe, Arizona *Jan 2018 - October 2019*

- Developed a visual programming interface with dynamically populated puzzle shaped blocks encoding the robot's possible actions to allow users to perform navigation and manipulation by simply connecting them in some logical order and integrated a feedback mechanism to explain failures based on the user's capability of understanding (**Thesis**).
- Designed and developed programming assignments on AI search problems in ROS and the Ping Pong game for Game Trees in RVIZ.

Technology Stack: Python, OpenRAVE, PDDL, ROS, Gazebo, RVIZ, Java, Html5, JQuery, AJAX

AI Engineer Intern | Invitae | Boston, Massachusetts *May 2019 - August 2019*

- Engineered an ML application to perform PHI scrubbing on unstructured medical reports using a Spacy and dictionary based model.
- Designed and build an end-to-end object detection pipeline for scanned images using OpenCV, OCR and ML models.
- Performed various data augmentation methods for generating a large custom dataset.

Technology Stack: Python, Numpy, Pandas, NLTK, Spacy, OpenCV, Sklearn, Flask, JS

Software Engineering Intern | Rockwell Collins | Irvine, California *May 2018 - August 2018*

- Architected an Android application for IFE which auto-detected cast enabled monitors powered by Raspberry Pi 3 and played DRM protected media on it without any internet connection.

Technology Stack: Python, Java, Electron, Raspberry Pi, Android Studio

ACADEMIC PROJECTS

Anomaly Detection | Statistical Machine Learning *Fall 2018*

- Adapted a hybrid model using five different machine learning models with weighted polling to detect component failures in the Air Pressure System of heavy Scania trucks.
- Applied various feature engineering methods to deal with inconsistent data.

Technology Stack: Python, Pandas, Scikit-learn, Matplotlib

Multi view 3D Object Reconstruction using Deep Neural Networks | Robotics *Spring 2018*

- Integrated ROS enabled 3D Recurrent Reconstruction Neural Network (3DR2N2) to generate the 3D shape of an object from 2D images and detected grasping poses on it.

Technology Stack: Python, Numpy, CNN, GRU

Smart Video Surveillance System using Deep Neural Network & POMDPs | Robotics *Spring 2018*

- Utilized a deep object detection network (YOLO) to capture an object's movements in the current camera frame which then served as evidence to a Partially Observable Markov Decision model for visual servoing.

Technology Stack: Python, Java, OpenCV, CNN

PERSONAL PROJECTS | GITHUB

- Implemented Policy Gradient & Double DQN to play the famous Atari game Pong (Tensorflow, OpenAI Gym, Numpy, OpenCV).
- Implemented vanilla GAN and VAE on MNIST dataset (Tensorflow, Tensorboard, Numpy).
- Designed and developed a high-performance algorithm for matrix multiplication and transpose using threads and without using any linear algebra library (C++).
- Devised algorithms that can detect cars parked at multiple parking spots, compare if two cars are same or not, predict the color of a car and output each car that was detected and how long it was parked for (approximately) within a given time interval. (Python, OpenCV, Numpy, CNN).