

OBJECTIVE

Seeking a Software Engineer intern position where I can leverage my technical expertise to tackle demanding challenges.

EDUCATION

- Stony Brook University, New York**

Master of Science (Computer Science)

Aug’17-Dec’18 (Expected)
GPA: 3.75/4.0

Member of Human Interaction (HI) Lab: Under the guidance of **Professor Roy Shilkrot**.
Course Highlights: Computer Vision, Artificial Intelligence, Analysis of Algorithms, Probability and Statistics for Data Science, Machine Learning, Database Systems.

- Pune Institute of Computer Technology (PICT), University of Pune, India**

Bachelor of Engineering (Information Technology)

May’13-May’17
GPA: 3.72/4.0

Course Highlights: Machine Learning, Operating Systems, Data Structures, Design & Analysis of Algorithms, Business Intelligence.

SKILL SET

- Programming Languages** : C++, Python, C, R.
- Tools & Technologies** : MATLAB, OpenCV, Tensorflow, Keras, SQL, MongoDB, Cassandra, Linux, HTML, CSS, PHP.

WORK EXPERIENCE

AlgoAnalytics, India
(Software Developer Intern)

Jul’16-May’17

- Modeled a software for **Logo Detection and Recognition in Real-World Images** in Convolutional Neural Networks using Tensorflow Deep Learning Library to recognize the presence/absence of a logo in a live camera feed or in static images.
- Retrained the final layer of **Google’s Inception-v3 model** to Flickr-logo32 dataset to transfer knowledge of significant features by using **Transfer Learning** concepts.
- Improved the overall training accuracy from **83%** to **87.6%** by data augmentation and hyperparameter tuning.
- Expanded this project by extracting logo images from Instagram public account (on hashtag basis) and utilized the above model to classify them into top 3 logo categories.
- Performed smart data analysis of the results obtained and stored in Cassandra.

Technologies used: Python, Tensorflow, Tensorboard, Convolutional Neural Networks, Machine Learning, Computer Vision.

PERSONAL PROJECTS

Drowsy Driver Detection

Nov’17-Dec’17

- Developed a software model for drowsiness detection of a driver by real-time **Eye-Tracking** in videos by implementing **Haar Cascade** and **CamShift** algorithm.
- Generated a sequence of feature vectors for consecutive frames by stitching significant features for each video frame extracted by CNN from the final **pooling layer**.
- Measured the drowsiness of the driver by providing the sequence (2048-D) as an input to **Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN)**, and sounded an alarm in case of detection of drowsiness.
- Enhanced and optimized network weights by **Adam Optimization Algorithm**.
- Accomplished an accuracy of **87.5%**, surpassing similar drowsiness detection models.
- [Project Demo Link \(YouTube\)](#)

Technologies used: Python 2.7, OpenCV 3.3.0, Tensorflow, Keras, CNN, RNN, LSTM, Computer Vision.

ACADEMIC PROJECTS

Face Detection/Tracking in Videos

Sept’17-Nov’17

- Programmed a **Face-Detector** software to detect a face in video frames using pre-trained **Viola-Jones** detector, and then to track the face throughout the video frames using **CAMShift, MeanShift, Particle Filter, Kalman Filter, Optical Flow** trackers.
- Analyzed the performance and accuracies of these trackers, Optical Flow tracker outperformed the rest.

Technologies used: Python 2.7, OpenCV 3.3.0.

3-D Scene Construction

Sept’17-Nov’17

- Constructed a **3-Dimensional** scene from multiple **Structured-light** scanning of the scene by calibrating the projector and obtaining 2D-3D pixel correspondences.
- Performed stereo calibration and stereo triangulation to obtain depth maps, which is utilized to construct the 3-D scene.

Technologies used: Python 2.7, OpenCV 3.3.0.

Semi-Automatic Binary Segmentation

Sept’17-Nov’17

- Built a user interactive binary image segmentation model to separate the object in focus from its background.
- Computed **Simple Linear Iterative Clustering (SLIC) super-pixels**, and executed a graph-cut algorithm to obtain a final binary segmentation.

Technologies used: Python 2.7, OpenCV 3.3.0.

Spam Filter for Emails

Nov’17-Dec’17

- Coded a Spam Filter to classify emails as spam/ham (2005 TREC Public Spam Corpus Dataset) by implementing **Multinomial Naive Bayes Algorithm**.
- Increased the training accuracy from **88.3%** to **91.8%** by applying **Laplace Smoothing** technique to smoothen the data.

Technologies used: Python 2.7, Artificial Intelligence

The Pac-Man Project

Sept’17-Nov’17

- Designed a **Pac-man game** utilizing different artificial intelligence techniques like **Reinforcement Learning, Probabilistic Inference** and generic graph search algorithms such as Depth First Search, Breadth First Search and **A-star with heuristics**.
- Implemented multi-agent Pac-man adversarial search by using **Alpha-beta Pruning, Minimax and Expectimax** algorithms.
- Programmed the Pac-man to navigate optimal routes to food locations in presence of hidden ghosts by implementing **Markov model** (Probabilistic Inference).

Technologies used: Python 2.7, Artificial Intelligence