Reshav Abraham

Full Stack ML Engineer

reshavabraham@gmail.com
908-731-1426
https://github.com/reshav-abraham
www.linkedin.com/in/reshav-abraham
160 Vroom Street, Apt 46,
Jersey City, NJ 07306

About me

Passionate software engineer with experience in Full Stack Development and Machine Learning. Experience building backend API's, Frontends. Experience with training and serving Machine Learning models.

Work Experience

Nlmatics

NLP Engineer

New York, NY — July 2019 - April 2021

Early stage startup specializing in Search and Extraction from vast volumes of unstructured text data. The product enables users to perform semantic search queries on their digitally uploaded data. The product empowers users to automate data entry tasks from elaborate and heavily detailed text (i.e. Loan Agreements, Offerendum Memorandums, Public Offerings) that would otherwise consume the time and effort of a skilled professional. The products backend API's were built with **Python**, **Swagger** (OpenApi 3.0.0) and served with **Flask** and **Gunicorn**. The frontend was implemented with **React** and served with **Nginx**. It was containerized with **Docker** and deployed with **Kubernetes**. The product was hosted on **GCP** and **Azure**.

∗ **Developed** a service to parse and index PDF documents with high fidelity. Implemented and tested document layout analysis algorithms to improve the classification of header, paragraph, and table text. This significantly improved the search quality of the product.

∗ **Lead** On-Prem installations for clients and customized deployments for restrictive environments. Prepared installations by simulating client environments. Designed deployment scripts for regular updates and roll-outs.

∗ **Designed** and **Implemented** a Search Flagging and Approval System to track regressions and improvements in search quality. This guided the development when testing new changes to the Search Engine pipeline.

∗ **Designed** and **Developed** back-end APIs with Python and Swagger.

∗ **Developed** front-end features with React and ANTD.

∗ Maintained and debugged CI/CD pipelines with Github Actions, Docker, Kubernetes, and GCP/Azure.

Dell EMC

Software Intern

Charlotte, NC — May 2017 — August 2017

∗ Extrapolated memory usage for enterprise data pipelining software by modeling a regression on real-time memory consumption data using Apache Spark.

Projects

Human Voice Detection

∗ Developed a neural network architecture using **CNN** and linear for processing audio signals to identify human voices.

∗ Developed a script for scraping audio from YouTube playlists.

∗ Utilized MFCC and Signal Processing techniques to prepare data.

Technical Skills

Languages Python, JavaScript, Bash, C++, Java

Frameworks PyTorch, Tensorflow, Swagger, React

DevOps Docker, Kubernetes, Nginx

Cloud GCP, Azure

Databases MongoDB, Postgres, Redis

Markup LaTeX, HTML, CSS, Markdown

Project Management Clubhouse, Jira

Misc Git, Linux, Web Scraping, Jupyter

Education

B.S Computer Engineering
Purdue University

West Lafayette, Indiana 2014 — 2018

Multi-core Processor System Verilog

∗ Implemented a synthesizable multi-core processor for processing MIPS assembly language in SystemVerilog.

Automated Nerf-Gun Turret

∗ Engineered a turret gun with 3D printed parts, STM32F407VGT7, and Raspberry pi. The turret gun was controllable with a Wii nunchuck and could detect and shoot human targets with a nerf dart.

∗ Implemented human-target detection and tracking with MobileNetSSD and OpenCV.

Certificates

Natural Language Processing with Deep Learning
Stanford University, CS224N
October 2020 — December 2020

∗ Developed a Neural Machine Language Translation model in **PyTorch**.

∗ Implemented **Encoder** and **Decoder** networks using **LSTM** and **CNN** layers for processing out-of-vocabulary words.

Coursera Certificates

∗ Natural Language Processing with Attention Models

∗ Natural Language Processing with Probabilistic Models

∗ Natural Language Processing with Classification and Vector Spaces

Planner

∗ Designed and Implemented A planner for productivity management. A user can create multiple plan workspaces and keep track of goals and progress in an organized manner. Frontend components were implemented with **React** and **Material Design**. Backend was implemented with **Python**, **FastApi**. **MongoDB** was used as the database.