CHONG HU

(917)-388-5186 | ch3467@columbia.edu | Apt. 4A, 169 Manhattan Ave, New York, NY, 10025
https://jacksnowwolf.github.io/ | https://www.linkedin.com/in/chong-hu-5b1a6514b/

EDUCATION

Columbia University (CU), New York, US *Aug 2019 - Dec 2020 (expected)*
The Fu Foundation School of Engineering and Applied Science GPA: 4.0/4.0
M.S. in Electrical Engineering
Courses: Database, Algorithm, Computer Networks, Programming Language & Translator, Stream Processing

Shanghai Jiao Tong University (SJTU), Shanghai, CN *Sep 2015 - Aug 2019*
Joint Institute: University of Michigan-Shanghai Jiao Tong University Joint Institute (UM-SJTU JI) GPA: 3.3/4.0
B.S. in Electrical and Computer Engineering; Minor in Data Science
Courses: Data Structures and Algorithms, Operating System, Methods and Tools for Big Data, AI Techniques

WORK EXPERIENCE

CertiK LLC., New York, US *May 2020 - Aug 2020*
Research & Development Intern, R&D Team

- Designed and built a task management system to connect ethereum and cosmos chain through websockets and handle tasks with multiprocessing in Golang; used DynamoDB as a cached database to provide data for front-end
- Constructed multiple strategies to combine security check logic and established endpoints with Lambda Function
- Provided RESTful APIs and a command line interface for task management; wrote unit tests for the functionality verification, logging mechanism and error handler to improve the system robustness; managed instances with Docker

MokaHR Inc., Beijing, CN *Dec 2018 - Apr 2019*
Software Engineer Intern, AI Team

- Developed a model with CTPN and CRNN to solve OCR problems (Chinese & English) in resume images using TensorFlow; simplified Network Structure and sped up inference time 2s/10s on average, lost only 2% accuracy
- Packaged model into a web service using gunicorn and Flask, provided APIs and deployed with systemd on Alibaba cloud
- Implemented cache mechanism with Redis and multistage recognition with high accuracy (over 90% per label)
- Improved 15% performance and 200% QPS over the original third-party service with parallel processing in Python

Beijing Infervision Inc., Beijing, CN *Jan 2018 - May 2018*
Software Engineer Intern, Modeling Team

- Applied YOLO V2 & V3 under darknet frame and FPN under MXNet for illness detection on DR images
- Utilized Focal Loss to focus on cases with fewer samples with TensorFlow; increased average accuracy by about 3%
- Connected recognition model to back-end inside docker and fixed bugs about medical images in data pipeline

PROJECTS

Rule-based Marketing Platform to Manage Call Detail Record (CDR) *Mar 2020 - May 2020*
Team Member, CU *Course: Large-scale Stream Processing*

- Simulated streaming CDR data in a generator with real-time interface to change modes, speed, distribution, etc
- Built Pub/Sub scheme using Redis as Message Queue and set up a middle-ware to provide stream to Spark streaming
- Provided multiple customizable templates to extract features; modularized and optimized streaming process
- Implemented a GUI application to visualize real-time streaming features and to receive live updates for Django back-end

Web Application for Video Object Segmentation and Visualization *Sep 2019 - Dec 2019*
Team Member, CU *Course: Big Data Analytic*

- Adapted OSVOS model to segment foreground object from short video; applied FFmpeg and OpenCV to extract single frames from video, mask with recognized foreground area, and render to video; calculated position of segmented object
- Provided web API to communicate video and corresponding metadata with front-end by using Flask
- Built Django web application to receive video files, play rendered video, visualize metadata of foreground object

Music Recommendation System Analyzed from Million Song Dataset (MSD) *Jun 2019 - Aug 2019*
Team Member, SJTU *Course: Methods and Tools for Big Data*

- Deployed Hadoop with Spark and Drill and extracted song information from 160GB avro files containing h5
- Built similar artist adjacent matrix using MapReduce in Hadoop and Spark using Java; used Naive Bayes to guide the scaling data; ran hierarchical and k-mean++ clustering methods to split the genres of different music.
- Visualized results in Matplotlib using Python and ggplot2 using R and constructed music recommendation logic

TECHNICAL SKILLS

Programming Language: Python, C++, C, R, Java, Golang, MATLAB, Julia, OCaml, Javascript, HTML, SQL.
Toolkits/Frameworks: Linux, Hadoop, Spark, Git, NumPy, pandas, TensorFlow, Matplotlib, OpenCV, Flask, Django