MINAL ACHARYA

+1 412-304-5084 ⋄ Pittsburgh, US

acharyaminal04@gmail.com ⋄ LinkedIn : minalmeg ⋄ Github : minalmeg ⋄ Medium : minalacharya

EDUCATION

MS, Mobile and IoT Engineering, Carnegie Mellon University - INI

BE, Computer Engineering, University of Mumbai 2016 - 2020

GPA : 3.4/4

SKILLS

Languages	Python, C
ML Frameworks	Keras, Tensorflow, PyTorch, Openvino, pandas, numpy, scikit, OpenCV
Other Frameworks	GDB, Git, pwntools, MySql, Flask
Platforms	Linux

EXPERIENCE

Research Assistant July 2023 - Current

Carnegie Mellon University - Catalyst Group *Pittsburgh, US*

- Currently, researching efficient methods for fine-tuning LLMs to reduce computational load while training under Prof. Zhihao Jia.

Research Engineer Nov 2021 - Dec 2022

REConnect Energy Solutions Limited *Bengaluru, India*

- Deployed a scalable pipeline for load forecasting model on production with less than 7mins of downtime annually.
- Designed feature engineering and deep learning architectures for load forecasting, with an avg MAPE of 3.8.

Jr. AI and IoT Engineer Nov 2020 - Oct 2021

Cynapto Technologies *Mumbai, India*

- Developed, optimised and deployed more than 8 computer vision applications for edge devices.
- Optimized the image input received from RTSP Streams to efficiently run on low computational edge devices using frameworks like GStreamer.
- Integrated multiple images processing techniques to implement bad feed removal from input images which reduced the computational load by 20%.

Intern May 2020 - Oct 2020

Cynapto Technologies *Mumbai, India*

- Ported windows based training-pipelines to linux servers which halved the computational resources.
- Integrated MySQL database with existing pipelines on linux.

PROJECTS

Scan.It - An OCR focused on improving accuracy of Tesseract library using image processing techniques, mainly for regional languages of India. The research done is published in IEEE Xplore Library

CargoCal - Built a tyre loading optimization application for Bridgestone Tyres to maximise profit by reducing no. of travel trips for delivery of tyres.

Malloc - Implemented a segregated free list allocator to optimise throughput of malloc()

Tiny Shell - Implemented a tiny shell that handled signals, errors and system calls using multiprocessing.

Proxy - Implemented a proxy between client and server to handle multiple requests using multi-threading.