

Raj Pulapakura

Machine Learning Engineer

rajedwinpulapakura@gmail.com | 0469 766 823 | Melbourne, Victoria

github.com/raj-pulapakura | linkedin.com/in/rajpulapakura

www.rajpulapakura.com | medium.com/@raj.pulapakura | youtube.com/@rajpulapakura9119

Technical Skills

Languages	Python, Bash, JavaScript/TypeScript, HTML/CSS, Dart, C++ (familiar)
Machine Learning and Data Science	Python, TensorFlow, Keras, PyTorch, Scikit-learn, OpenCV, Hugging Face, NumPy, Pandas, Matplotlib, Plotly, Seaborn, ChromaDB, Langchain, OpenAI API
Database	SQL, MySQL, PostgreSQL, MongoDB (NoSQL), Redis, Firebase Firestore (NoSQL), Firebase Realtime Database
Cloud and Infrastructure	AWS (ECR, EKS, EC2, S3, API Gateway, IAM, DynamoDB, Lambda, Athena, Glue, ElasticSearch), GCP (Vertex AI), BentoML, Terraform, Firebase, Git, GitHub, Docker, Kubernetes, Shell, Modal
Frontend + Backend Development	JavaScript/TypeScript, React.js, Next.js, Vercel, HTML, CSS, Tailwind, Material UI, Redux, Zustand, Node.js, REST APIs, FastAPI, Flask, Express.js, GraphQL, ApolloGraphQL, TypeORM

Work Experience

A.I.GORITHM, Melbourne

Jan 2024 – Feb 2024

Software Engineer, Internship

- **Overview:** Designed and implemented an end-to-end, full-stack Property Report Generation Tool that enables farmers to get detailed agricultural reports on their geographical AOI (area of interest).
- **Backend:** Developed a containerized backend using a microservices architecture with Docker and Docker Compose, building 6 Python-based FastAPI services to process geospatial and agricultural data with Pandas and GeoPandas, integrating with third-party services including Stripe, Firebase and SendGrid.
- **Frontend:** Developed a containerized website using Next.js (React.js meta-framework), TypeScript and Docker, utilizing Server-Side Rendering to minimize load times.
- **Data:** Employed AWS S3 and Glue to manage several agricultural datasets, stored in Apache Parquet and CSV format. Utilised SQL and AWS Athena to query these datasets and extract mesh-block-specific information.
- **Deployment:** Deployed the end-to-end system to the AWS public cloud, utilizing services such as AWS Elastic Container Service (ECS), AWS Elastic Container Repository (ECR) and AWS Load Balancer for public internet access.
- **DevOps:** Conducted DevOps tasks by utilizing Terraform to automate the provision of AWS resources, and GitHub Actions to develop the CI/CD pipeline.

Certifications

- | | |
|---|----------|
| • Database and SQL for Data Science with Python , IBM | Dec 2023 |
| • Deep Learning Specialization , DeepLearning.AI | Dec 2023 |
| • Advanced Machine Learning on Google Cloud , Google Cloud | Nov 2023 |
| • IBM Professional Machine Learning Certificate , IBM | Sep 2023 |
| • TensorFlow Developer Certificate , TensorFlow | Aug 2023 |

Project Work

ClarityScan: Deep Learning for Tumour Diagnosis, [code](#), [video](#)

Project Overview: ClarityScan is a medical imaging and analysis tool that enables radiologists to enhance brain MRI scans and automatically identify glioma tumours in the scans.

- Developed 2 Deep Learning models using TensorFlow for Noise Removal and Tumour segmentation, reaching a Dice Coefficient of up to 91%.
- Deployed models to RESTful APIs with GPUs using FastAPI and Modal serverless functions, ensuring inference time during production was in the range of 5 to 30 seconds.
- Built a web application so that radiologists can upload their MRI scans, denoise them, and automatically identify glioma tumours.
- *Technologies:* Next.JS, TypeScript, Tailwind CSS Python, TensorFlow, NumPy, FastAPI, Modal, AWS ECS & LB, Git, GitHub, AWS ECS, AWS ECR, AWS EC2

Computer Vision-Powered Search Engine, [code](#), [web](#), [video](#)

Project Overview: An app that enables users to find similar visuals based on their images.

- Used a pre-trained MobileNetV3 neural network in PyTorch to compute image embeddings.
- Deployed the model through a REST API backend built with BentoML and hosted using automated Terraform infrastructure through AWS API Gateway and AWS Lambda.
- Developed a web interface with drag-and-drop functionality enabling users to find similar images.
- *Technologies:* Python, PyTorch, REST API, BentoML, TerraForm, AWS API Gateway, AWS Lambda, Next.JS, TypeScript, Vercel, Tailwind CSS, Git, GitHub

Natural Language Processing (NLP) Text Toxicity Checker, [code](#), [web](#), [model](#)

Project Overview: An app that enables users to check their text for toxicity across 6 toxicity categories (toxic, severe toxic, obscene, threat, insult, identity hate).

- Employed TensorFlow to train a text classification model using LSTMs, on 60 MB of textual data.
- Quantized the model using TensorFlow Lite and TensorFlow JS for efficient edge-device inference and stored the model shards in a GitHub repo.
- Developed a web interface using Next.js (deployed with Vercel), TypeScript and Tailwind CSS, which enables users to enter their text and analyse the toxicity of their text.
- *Technologies:* Python, TensorFlow, TensorFlow Lite, TensorFlow JS, Kaggle, Next.js, TypeScript, Vercel, Tailwind CSS, Git, GitHub

Other

- Have made community contributions to multiple open-source packages including TensorFlow and Scikit-learn, mostly documentation-related. Actively collaborated on GitHub issues with fellow developers, focusing on writing clean and maintainable code.
- Write a technical blog on [Medium](#), sharing content on machine learning, deep learning, computer vision, full-stack development, and software development.
- Post videos on my [YouTube](#) channel, demonstrating my projects and sharing knowledge on machine learning and data science, web/mobile development, and software engineering.