ROBERT PAULS

rpauls2002@gmail.com | 07476630872 | linkedin.com/in/rpauls02 | github.com/rpauls02 | rpauls.dev

PROFILE

Computer science graduate specialising in software engineering and AI/ML, with hands-on experience shipping full-stack apps (React/TypeScript, Flask, GCP/Firebase) and computer vision models (PyTorch, OpenCV, YOLO) from concept to cloud deployment. Independently designed and deployed cloud infrastructure, debugged cross-stack issues end-to-end, and built and evaluated ML pipelines across computer vision and NLP. Pairs this with three-plus years of hands-on hardware diagnostics, targeting graduate software engineering or AI/ML roles.

EXPERIENCE

Computer Repairs, Hornchurch - Founder Aug 2022 – Present

- Diagnose and repair desktop systems, cutting average turnaround time by 30% through systematic hardware/software fault-finding.
- Advise customers on tailored hardware builds for gaming, creative, and business use cases, translating technical trade-offs into plain-language recommendations.

Rok Co., Romford - Assistant Feb 2015 – Mar 2021

- Supported a small business by completing 100+ handyman tasks, maintaining repeat custom and resolving client issues on first visit in most cases.

EDUCATION

City St George's, University of London - BSc Computer Science (2:1) Sept 2021 – June 2026

All Saints Catholic School, Dagenham Sept 2014 – May 2021

- A-Levels: Design and Technology (A), Computer Science (B), CTEC Business Studies (Pass).
- GCSE: 10 GCSEs (9-4) including Computer Science.

TECHNICAL SKILLS

Programming/Scripting: CSS, HTML, JavaScript, Python, SQL, TypeScript.

Frameworks & Libraries: Flask, Node.js, OpenCV, Pandas, PyTorch, React, TensorFlow, Vite, YOLO.

Databases & Cloud: Cloud Run, Firebase, Firestore, GCS, GitHub Actions, MySQL, MariaDB.

Tools & Platforms: GitHub, Linux, Roboflow, Visual Paradigm, VirtualBox, Windows.

Hardware: Assembly, Maintenance, Optimisation, Troubleshooting.

PROJECTS

- F1StatFinder: live F1 statistics platform covering every championship season since 1950, built solo across a React/TypeScript frontend, Flask backend and GCP infrastructure.
  - Designed a Firestore caching layer with decade-bucketed documents in front of the FastF1 and Jolpica-F1 APIs, fitting 75+ years of results within Firestore's 1 MB document limit.
  - Automated weekly data sync with Cloud Scheduler, serving a React/TypeScript frontend on Firebase Hosting with race-weekend countdowns and Open-Meteo forecasts.
  - Shipped AI-generated race recaps via the Claude API, gated behind an explicit user-consent prompt.
  - Fixed production deployment failures across build configuration, IAM permissions and environment variables, using Claude as a coding assistant to speed up debugging and architecture decisions.
- F1Detect: real-time object detection pipeline for identifying F1 cars in footage.
  - Built custom OpenCV scripts to extract and sample frames from raw footage, then trained YOLOv11 with CUDA acceleration, resolving training-stability issues through iterative debugging.
  - Ran evolutionary hyperparameter searches (10 to 100 iterations) on a Roboflow dataset blending video-game and real-world footage, with a benchmarking harness for per-frame timing.
  - Reached 0.945 mAP@50 (0.906 mAP@50-95) on validation, with ~60 FPS real-time inference on GPU across 1,100+ test frames.
- Face Cover Detection: multiclass mask detection system comparing computer vision pipelines.
  - Labelled a balanced real-world test set for a 2,800+ image coursework dataset skewed 85% to one class.
  - Built and compared HOG+SVM, SIFT bag-of-visual-words+MLP and custom CNN pipelines, grid-searching the classical models and stress-testing SIFT features against noise, rotation and affine warping.
  - Custom CNN scored best at 91% test accuracy against an 85% majority-class baseline; real-world testing (46%) exposed weak generalisation on minority classes.
- Phishing Email Detection: NLP-based classifier distinguishing phishing emails from legitimate ones.
  - Cleaned and tokenised email text, removing nulls, duplicates, and empty bodies.
  - Built a custom CBOW Word2Vec model from scratch in PyTorch, evaluating embeddings against bag-of-words baselines (DT, SVC, NB).
  - Achieved 96% accuracy (phishing recall 94%, F1 0.95) with an MLP on CBOW embeddings, beating Random Forest's 93%.

HOBBIES AND INTERESTS

- Formula 1 and wider motorsport; following desktop and server hardware developments.