John Subba

AI FIRST SOFTWARE ENGINEER
johnsubba404@gmail.com | +9779861627795 | Kathmandu
LINKEDIN | GITHUB

EDUCATION

PURBANCHAL UNIVERSITY Feb/2021 - June/2025
INFORMATION TECHNOLOGY BACHELORS Nepal

EXPERIENCE

HIMALOGIC | GPU-ACCELERATED COMPUTER VISION ENGINEER (AR/VR SYSTEMS) Kathmandu, Nepal |
October 2025 – Present

Role Focus:

Low-level optimization of real-time AR/VR computer vision pipelines, emphasizing GPU acceleration, memory efficiency, and cross-platform execution under strict latency and power constraints.

Technical Contributions & Experiments:

* OpenCV SIFT & FLANN pipelines port on GPU:

- SIFT: Implemented Gaussian pyramid, DoG, and scale-space extrema on GPU; orientation assignment and descriptor computation partially attempted due to platform complexity. CUDA used for rapid descriptor testing.
- FLANN matcher: KD-tree traversal attempt failed on GPU; brute-force + voxel-grid approach successfully optimized for latency and memory.

* Developed Metal backend (iOS, Objective-C++) and CUDA backend for testing.

* Occasionally implemented native Android Camera2 to C++ bridges driver for real-time tracking as per needs with goals to minimize memory copies at near-zero-copy buffers for reduced latency. Geometric correction were limited by Camera2/OEM constraints; tested on Android devices.

* Tackled heterogeneous compute challenges: Metal, CUDA, Vulkan, OpenGL ES 3.1/3.2, WebAssembly. Evaluated precision, threadgroup sizing, and memory placement to maximize GPU occupancy under mobile power constraints.

* WebAssembly experimentation with JS runtime limitations constraints for full tracker porting.

Core Themes:

GPU memory hierarchy, SIMD/vectorization, zero-copy pipelines, real-time constraints, cross-platform portability, practical problem-solving under architectural limits.

Tech Stack:

C++, OpenCV, Metal Shading Language, CUDA, OpenGL ES 3.1/3.2, WebAssembly, Objective-C++, Old School AR/VR Vision Pipelines

INDEPENDENT CONTRACTOR | SENSOR FUSION ENGINEER Remote | August 2025 – September 2025

* Designed and implemented real-time multi-sensor fusion systems in C, C++, and CUDA, emphasizing deterministic execution, numerical stability, and memory efficiency.

* Developed state estimation pipelines grounded in rigid-body motion modeling, Lie group formulations (SO(3), SE(3)), and stochastic dynamic and measurement models.

* Implemented probabilistic estimation algorithms including KF, EKF, UKF, and nonlinear estimation, modeling sensor uncertainty, bias, and noise propagation.

* Integrated IMU, positioning, LiDAR, RADAR, and camera measurements into unified state-space models, supporting navigation, tracking, and localization for wheeled, aerial, and pedestrian platforms.

* Designed and prototyped SLAM and graph-based optimization pipelines, loop-closure detection, and trajectory estimation frameworks.

* Applied learning-augmented fusion, combining physics-based models with neural network estimators for sensor correction, perception enhancement, and depth/optical flow refinement.

* Leveraged agentic AI for rapid prototyping, completing multi-month fusion and SLAM development in 2 months, focusing engineer effort on system architecture, integration, and real-time performance.

* Implemented systems on ROS2 and custom low-latency pub/sub frameworks in Linux, ensuring real-time determinism and high-throughput for embedded/desktop platforms.

* Conducted algorithm verification via simulation and controlled experiments, including observability analysis, numerical conditioning, and computational performance tuning.

Technical Stack & Concepts:

Sensor Fusion, Probabilistic State Estimation, SO(3)/SE(3), SLAM / Graph Optimization, Learning-Based Fusion, IMU / GNSS / LiDAR / Vision, CUDA / C / C++, ROS2 / Custom Low-Latency Pub-Sub, AI-Accelerated Development, Carla Simulator, Agentic AI and IDEs

GUARDWARE AUSTRALIA | Windows Kernel Programmer & Security Engineer | Occasional Python Developer

Remote | February 2025 – July 2025

Write, debug, and optimize kernel-mode drivers in C/C++ for enhanced security and performance.

Kernel-Level Security Enhancements.

System Debugging & Troubleshooting.

Cross-Functional Collaboration.

Python-Based Automation.

Code Quality & Best Practices.

GUARDWARE AUSTRALIA | Python & C++ Intern

Remote | Jan/2025 – Jan/2025

Spearheaded the complete analysis and transformation of a legacy system originally developed in C#, by conducting an in-depth study of its architecture and performance bottlenecks.

Successfully redesigned and reimplemented the system from scratch using Python, focusing on scalability, maintainability, and performance improvements integrating critical enhancements that optimized resource usage and reduced latency.

Utilized strong proficiency in Python and C++ to streamline the codebase, refactor legacy modules, and eliminate inefficiencies, resulting in a more robust and agile platform.

Collaborated closely with cross-functional teams to ensure a seamless transition, improved system reliability, and enhanced overall user experience.

Demonstrated a strong commitment to continuous improvement and technical excellence, effectively elevating the systems performance to meet modern industry standards.

GOLDMAN SACHS | Software Engineering Virtual Experience Program on Forage

Feb/2025 – Feb/2025

* Completed a job simulation as a Goldman Sachs governance analyst responsible for assessing IT security and suggesting improvements.

* Identified that the company was using an outdated password hashing algorithm

by cracking passwords using Hashcat.

* Wrote a memo for my supervisor summarizing a range of proposed uplifts to increase the companys level of password protection including extending minimum password length and using a dedicated hashing algorithm.

ELECTRONIC ARTS | Software Engineering virtual experience program on Forage

February 2025 – February 2025

* Proposed a new feature for the EA Sports College Football and wrote a Feature Proposal describing it to other stakeholders.

* Built a class diagram and created a header file in C++ with class definitions

for each object.

* Patched a bugfix and optimized the EA Sports College Football codebase by

implementing an improved data structure.

LAUNCHPAD AI | Research Fellow

October 2024 – December 2025

* Pioneered the development of the Advanced 3rd Party Data Integration System, a solution designed for privacy-preserving data matching and integration. The system employs a sophisticated two-phase approach, utilizing advanced algorithms and privacy-centric methods to securely integrate datasets without exposing Personally Identifiable Information (PII).

* Built with advanced cryptographic and privacy-enhancing frameworks to comply with GDPR, CCPA, and similar regulations along with automated pipelines for data evaluation, candidate matching, and validation, optimized for large-scale applications designed to serve diverse industries, including healthcare, finance, and e-commerce, where ethical data sharing is paramount.

SKILLS

Programming Languages	C++ (Intermediate and above), C (Intermediate and above), Python (Intermediate and above), SQL (Intermediate and above), Proficient at analysing code generated via LLM., English (Fluent)
Libraries/Frameworks	Tensorflow, Numpy, Pandas, Matplotlib, Pytorch, scikit-learn, scipy, Cmake, C++20, LLM Prompting (Intermediate and above)
Tools / Platforms	Git, GitHub, Linux, Cuda
Databases	MySQL

PROJECTS / OPEN-SOURCE

ESP-32 FACE COUNTER SYSTEM | Link

Flask, Mediapipe, Python, C/C++

Developed an ESP-32-based face counter system using Flask, Mediapipe, Python, and C/C++ that achieved above 85% accuracy rate in detecting and counting faces in real-time.

OBJECT DETECTION FOR SELF-DRIVING CAR | Link

CUDA, PYTHON, MobileNetV2, Tensorflow, Numpy, Sklearn, OpenCV,

This was just a simple assessment task that could be further progressed for Self-Driving Cars.

Althougth purely for basic Research purpose but the results using KDDI datasets were worthy of the effort.

The F1-Score I got for each labled Class is shown below:

Cars: 0.90

Trucks: 0.85

Pedestrians: 0.72

Cyclists: 0.75

Vans: 0.82

FINE-TUNING-USING-RESNET50 | Link

Torch, Torchvision, Scipy, PIL, Numpy

I was able to get around 88% accuracy without the risk of overfitting. It's not above 95% but I think it's still the most optimum fit.

FACE-MOOD MUSIC PLAYER | Link

Python (Level: Intermediate and above), Media-pipe

- This prototype system belongs to entertainment category, and can analyse emotions to play songs matching the vibe. What's different is that it doesn't uses pretrained model, but optimized using the face-detection techniques to recognize the emotions.
- Applicable for edge systems and the ML techniques is used based on face-landmarks points calculations.

VISION GUARDIAN COMMANDER | Link

Java, JavaFx, OpenCV, SQL

- This prototype system uses simple Haar classifier for face detection and can LBPH-classifier for face recognition.
- Although not very accurate but works under favourable lightning conditions with above 70% accuracy.
- Used to recognize the person and send the email along with a frame.

ESP-32 LORA GEO FENCING SYSTEM | Link C (Level: Intermediate and Above)

- This prototype system is basically using Lora module to transmit the GPS coordinates in real-time. Although I achieved the intended task but the receiver system has lot of potential for enhancements.

C++ KNOWLEDGE-INDEPTH:

Data Structures from STL : Lists, Stacks, Queues, Trees, Heaps, Graphs, Hash-Tables and Bloom Filters.

Algorithms : Divide and Conquer, Greedy, Graph and Dynamic Programming.

OOP Concepts :- Basic to advanced including the SOLID principles.

Template Meta Programming.

Memory Management and Smart Pointers.

Functional programming including First-Class, Higher-Order and Pure Functions.

Concurrency and Multithreading : Managing threads and sharing data and designing concurrent code using coroutines. Lock-based concurrent data structures like thread-safe singleton pattern, synchronized counters and concurrent stacks. Lock-free concurrent data structures using atomic types like lock-free queue, hash-table and set.

Knowledge of World-Ready Design patterns: Creational, Structural and Behavioural.

Networking and Security: Sound Knowledge of network programming in C.

Debugging and Testing: Root cause analysis, static and dynamic analysis, TDD, BDD, etc

Knowledge of Large-Scale Applications Design: Cross-platform, horizontal and vertical Scaling, designing data-intensive applications.

CERTIFICATIONS

- Comprehensive Introduction to Tensorflow - CodeSignal
- Introduction to Object-Oriented Programming in C++ - Coursera (University of London)
- Data Science Fellow - Fellowship.ai