Tanmay Bishnoi

416-818-9925 | tbishnoi@torontomu.ca | linkedin/tbishnoi | github/tanmayyb

Education

Toronto Metropolitan University (TMU) Toronto, ON
Bachelor of Engineering in Electrical Engineering *Aug 2019 – Exp Aug 2025*

- Dean's List (2019)
- Relevant Courses:
  Linear Algebra, Discrete Maths, Calculus I & II, Probability & Statistics I, Digital Systems, Signals & Systems I & II, Control Systems I, Computer Networks, Intelligent Systems, Data Structures & Algorithms, Electromagnetics

Neuromatch Academy (NMA) Remote
Deep Learning and Computational Neuroscience Summer School *Jul 2021 – Aug 2022*

- Deep Learning Student - Summer of 2022
- Computational Neuroscience Student - Summer of 2021

Professional Experience

Software Solutions Engineering Intern, R&D May 2023 – Present
Advanced MicroDevices (AMD) *Markham, ON*

- Assisted AMD in optimizing software power and performance for next-gen Display and Video IPs
- Studied high-res power signals to model SW Architecture Power Profile and developed Signal Processing Pipelines
- Developed Simulator in Python to predict power and performance of a Video IP using real-world OS events (ETW)
- Boosted org-wide productivity by developing data collection, processing, and simulation tools in C++ and Python

Machine Learning Engineer, Audio Tagging Jan 2023 – Sep 2023
UofT Machine Intelligence Student Team (UTMIST) *Toronto, ON*

- Created ML solution for detecting Wind Turbine Noise (WTN) in Ontario farms for Aercoustics Ltd. (AEL)
- Spearheaded literature reviews, exploratory analysis, dataset generation, and model training and optimization
- Implemented a Semi-Supervised Contrastive Learning based model (COLA, Google Research, 2020) in TensorFlow
- Delivered trained model with >96% validation accuracy to AEL for automation of manual-labelling pipeline

Software Engineering Lead, Rover Autonomy Feb 2022 – Jul 2023
Toronto MetRobotics (TMR) *Toronto, ON*

- Developed full-stack autonomy software in Python and ROS 2 for a 6-wheeled 50 lbs Mars Rover
- Researched 2D RGB + 3D (pointcloud) based navigation algorithms using OpenCV and Zed Stereo Cameras
- Achieved <2m accuracy for point-to-point robust autonomous traversal on wide range of terrains
- Secured position in top 5% at the prestigious University Rover Competition (URC) held at MDRS, Utah, USA
- Contributed 8K+ lines of code via Git and mentored team members on leadership and problem solving

Undergraduate Research Volunteer, SLAM Aug 2022 – Dec 2022
Autonomous Vehicles Lab (AVL) *Toronto, ON*

- Implemented Real-Time Appearance Based Mapping (RTABmap) SLAM algorithm using Intel Realsense and ROS 2

Selected Projects

RISC-V Assembly FW Simulator | *Python, Plotly Dash, Websockets* Aug 2023 – Dec 2023

- Developed a RISC-V assembly firmware simulator for a 32-bit microprocessor on a next-gen AMD SoC
- Enhanced development workflow via register-level visualization of program control flow and algorithm compute

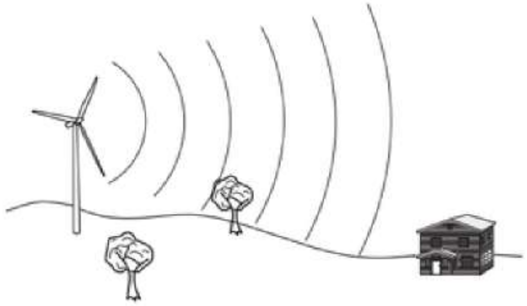
Modelling Information Flow (CompNeuro) | *Python, NumPy, SciPy, Plotly* Jul 2021 – Aug 2021

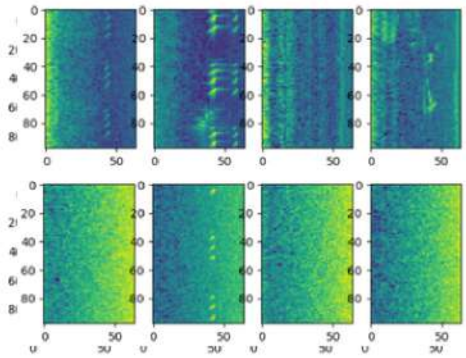
- "Feedforward Functional Hierarchy of Information Processing in the Mouse Brain during a Sensorimotor Task"
- Researched Neuropixel dataset to study how information travels through Visual Cortical layers in the Mouse Brain
- Verified Inter-Spike Interval (ISI) latencies in early Visual layers follow established theoretical Neuroscience model

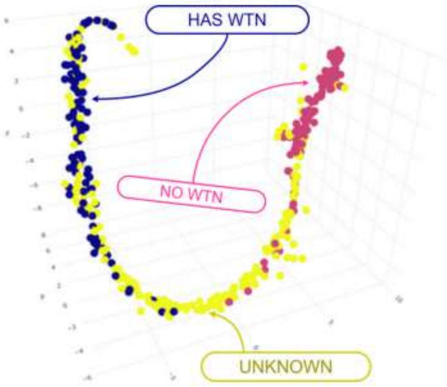
Technical Skills

Languages: Python, C/C++, MATLAB, Java, RISC-V Assembly, Bash
Deep Learning: TensorFlow (TF), PyTorch, NumPy, Pandas, scikit-learn
Robotics: Robot Operating System (ROS) 2, Gazebo, RViz, OpenCV, Nvidia Jetson, RaspberryPi, STM32
Tools: Git, Linux/Unix, CMake, CUDA, Visual Studio, LaTeX, Docker, Virtual Machines (VM)

WIND TURBINE NOISE DETECTION - ML MODEL - UTMIST X AEL







What?

- **Aercoustics Engineering Ltd. (AEL)** sponsored to develop ML solution to detect **Wind Turbine Noise (WTN)** pollution in Ontario Farms.
- Project aimed at saving time and labor by **automating WTN detection** for stakeholders interested in **minimizing health impact** of WTN.

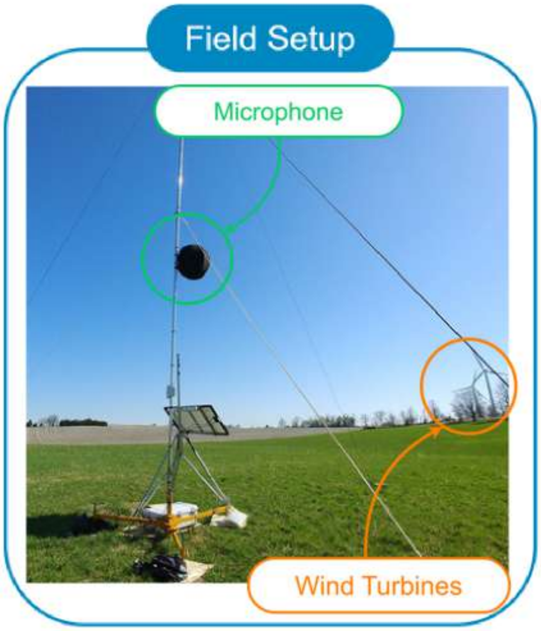
How?

- Conducted Literature Review and selected model architecture.
- Implemented **Contrastive Learning** model (**COLA** by Google Research, 2020) with **Tensorflow** and **CUDA** to solve for WTN detection.
- Spearheaded **full MLOps Lifecycle** (Dataset Gen, to Model Val.).

Results

- Final model able to predict WTN noise on labelled **validation datasets** with **96% accuracy**.
- Interpretability study showed **t-SNE embeddings** of model's encodings **seperated** different audio classes fairly well.

Field Setup

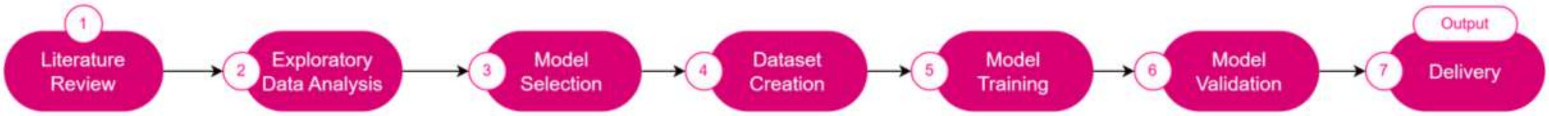


Model	Training Dataset	Validation Dataset	Validation Accuracy	Batch Size (n=)	Epochs
COLA	R03 (n=848)	R01 (n=600)	81.99%	64	50
			95.99%	1024	50
EfficientAT	R01 (n=600)	R03 (n=848)	**91.00%**	32	5

Contrastive Loss

$$\mathcal{L} = -\log \frac{\exp\left(\mathrm{s}(\overset{\text{Anchor}}{x}, \overset{\text{Positive}}{x^{+}})\right)}{\sum_{x^{-} \in \mathcal{X}^{-}(x) \cup \{x^{+}\}} \exp\left(\mathrm{s}(x, \underset{\text{Negative}}{x^{-}})\right)}$$

- The similarity between **"anchor" example** and a **related example** should be greater than between **anchor** and **unrelated examples**.
- **positives** are chosen from the same audio class as the **anchor**, and **negatives** are chosen from other audio classes.

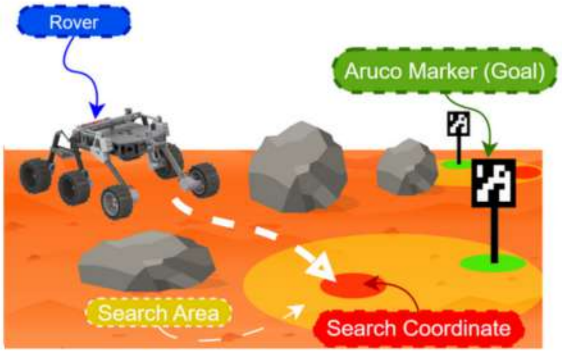


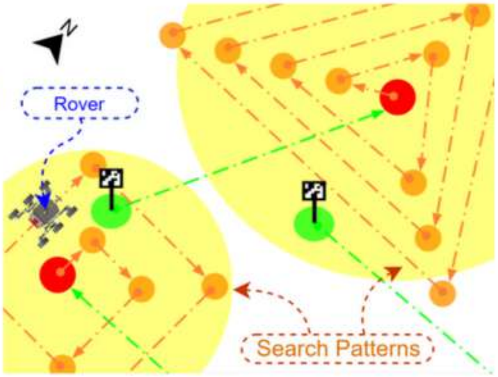
TANMAY BISHNOI

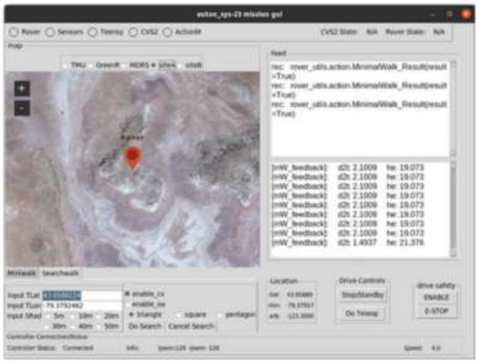
ELECTRICAL ENGINEERING AT TORONTO METROPOLITAN UNIVERSITY

tbishnoi@torontomu.com
linkedin.com/in/tbishnoi
(416) 818 9925

MARS ROVER - AUTONOMY SOFTWARE - R3 ROBOTICS






auton_sys-23 mission gui
○ Rover ○ Sensors ○ Teensy ○ CVS2 ○ ActionM
CVS2 State: N/A Rover State: N/A
map
TMU Greenft MDRS ● Siteh sitelt
+
-
feed
MiniWalk SearchWalk
Input TLat
Input TLon -79.3792462
Input SRad 5m 10m 20m 30m 40m 50m
● enable_cv
enable_ow
● triangle square pentagon
Do Search Cancel Search
Location
Drive Controls
Stop/Standby
Do Teleop
drive safety
ENABLE
E-STOP
Controller ConnectionStatus
Controller Status: Connected
Info:
Speed: 4.0

```
rec: rover_utils.action.MinimalWalk_Result(result=True)
rec: rover_utils.action.MinimalWalk_Result(result=True)
rec: rover_utils.action.MinimalWalk_Result(result=True)
```

```
[mW_feedback]:  d2t: 2.1009  he: 19.073
[mW_feedback]:  d2t: 2.1009  he: 19.073
[mW_feedback]:  d2t: 2.1009  he: 19.073
[mW_feedback]:  d2t: 2.1009  he: 19.073
[mW_feedback]:  d2t: 2.1009  he: 19.073
[mW_feedback]:  d2t: 2.1009  he: 19.073
[mW_feedback]:  d2t: 2.1009  he: 19.073
[mW_feedback]:  d2t: 2.1009  he: 19.073
[mW_feedback]:  d2t: 1.4937  he: 21.376
```

What?

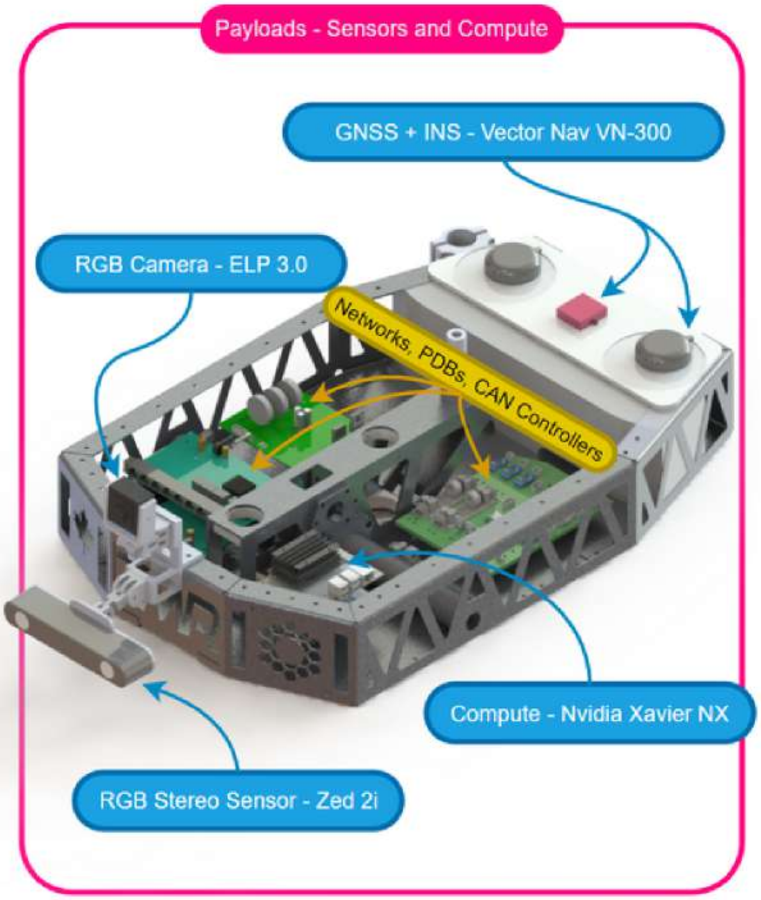
- Wrote **complete autonomy software** for Mars Rover for **University Rover Challenge (URC) 2023**.
- The rover **navigates** to and **searches** for visual markers (AR Tags) at specified search coordinates.
- The Rover navigates **full course autonomously** and **avoids obstacles** like rocks and ditches.

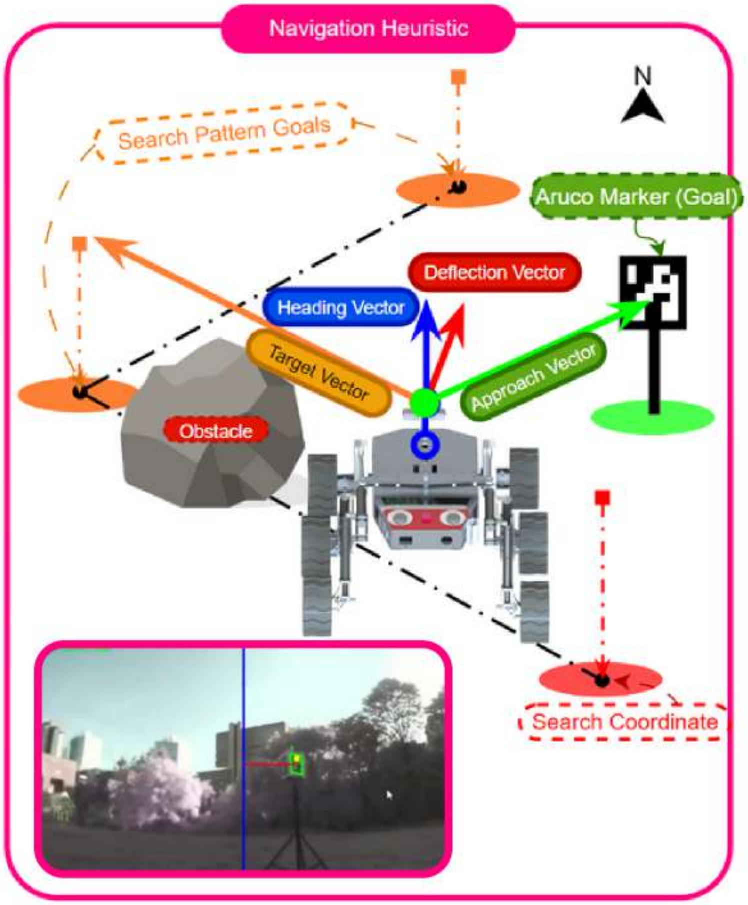
How?

- Designed **point-to-point** navigation algorithm by integrating **GNSS**, **INS**, and **RGB Stereo** sensor data.
- Designed **Search Patterns** for searching at target coordinates.
- Created **Computer Vision** Algorithm for **scanning** and **approach**.
- Implemented software using **Python**, **ROS 2**, **OpenCV** and **GStreamer.**

Results

- Achieved **<2m** accuracy on p2p autonomous navigation task.
- Completed **60%** of obstacle course under record time.
- Achieved **6th** place out of 38 teams for autonomous mission.
- Mentored **3 members** to lead the team for following URC missions


Payloads - Sensors and Compute
GNSS + INS - Vector Nav VN-300
RGB Camera - ELP 3.0
Networks, PDBs, CAN Controllers
Compute - Nvidia Xavier NX
RGB Stereo Sensor - Zed 2i


Navigation Heuristic
N
Search Pattern Goals
Aruco Marker (Goal)
Deflection Vector
Heading Vector
Target Vector
Approach Vector
Obstacle
Search Coordinate