2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

Terrain Classification with a Reservoir-Based Network of Spiking Neurons

Xinyun Zou*, Tiffany Hwu†‡, Jeffrey Krichmar†*, and Emre Neftci†*

**Department of Computer Science, University of California, Irvine,* Irvine, California, USA, 92697
†*Department of Cognitive Sciences, University of California, Irvine,* Irvine, California, USA, 92697
‡*HRL Laboratories, LLC,* Malibu, California, USA, 90265

2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

Background

Outdoor Robotics Challenges

- Changes in lighting
- Different terrains
- Lack of continuous power

Autonomous Navigation Implementation

- Long-term strategies
  - path planning
  - SLAM
- Reactive strategies
  - terrain classification
  - obstacle avoidance
  - road following

Complete Neuromorphic Solution

- Massively parallel
- Event-driven
- Energy-efficient

Introduction

- <u>Neuromorphic architectures</u> have potential for controlling outdoor robotics under *tight power constraints*.
- <u>Spiking neural networks</u> take advantage of neuromorphic hardware
- Terrain information is critical for planning trajectories.
- We developed a Reservoir-based Spiking Neural Network (r-SNN)
  - A **biologically inspired**, **energy efficient** algorithm for terrain classification
  - Terrain data collected on **grass, dirt, road** with an **Android-Based Robot (ABR)**

Terrain Data Collection with ABR

- Testing environment: a 19-acre botanical garden (Aldrich Park) at UC Irvine
- Data collection with our Android-Based Robotics (ABR) Platform
  - 3D linear accelerometer and gyroscope data collected at 100 Hz
  - Screenshots captured at 20 Hz
  - 42 trials (1-5 min per trial)

Grass (0)

Dirt (1)

Road (2)

Captured by the phone camera

(resolution of 176-by-144 pixels)

Android phone with GPS, accelerometer, gyroscope, camera, and our navigation control app

IOIO-OTG

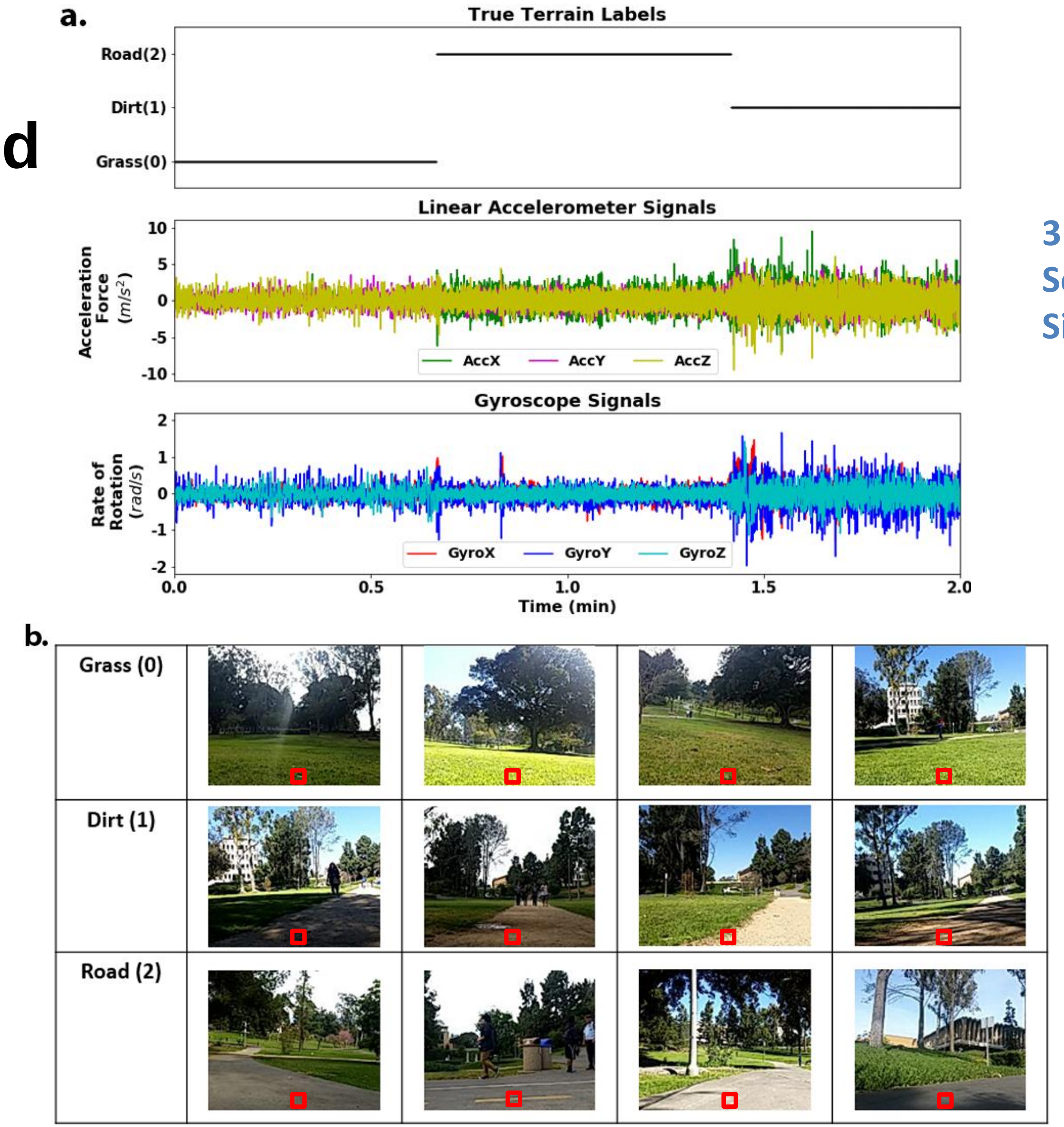
Dagu Wild Thumper 6WD All-Terrain Chassis

http://www.socsci.uci.edu/~jkrichma/ABR/index.html

Terrain Data Collection with ABR

Grass (label = 0)	Dirt (label = 1)	Road (label = 2)
		

Sample Collected Data



3D Sensor Signals

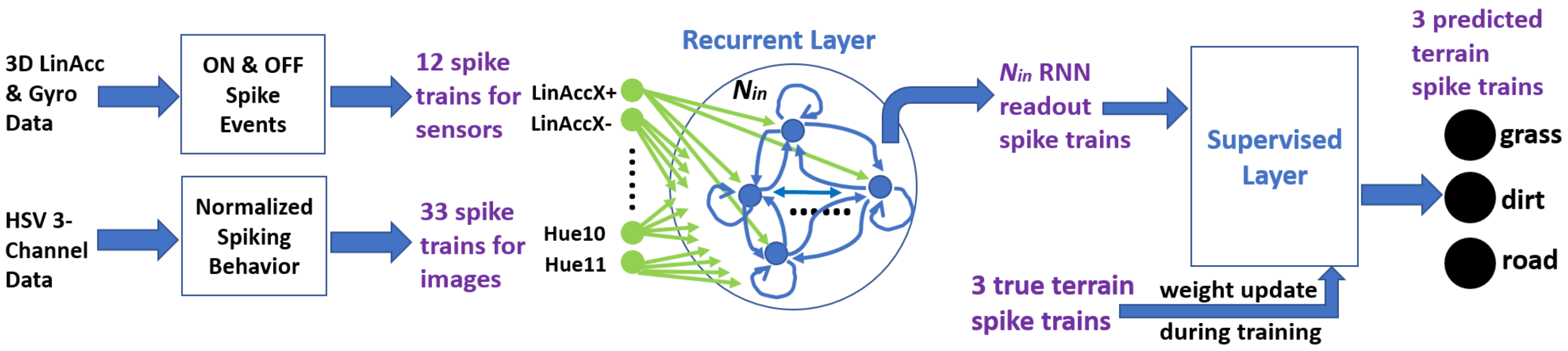
Camera Frames

Note: Each frame was cropped to keep only the bottom-center 5-by-5 pixels as the **terrain visual information** fed into the model.

Reservoir-based Spiking Neural Network (r-SNN)

- A recurrent neural network (RNN) uses its internal state as the memory to process arbitrary sequences of inputs
- This recurrence can be tractably harnessed using a **reservoir-based approach**, such as Liquid State Machines (LSM)
  - **recurrent weights** in the RNN are **randomly generated**,
  - only **the RNN readout is trained**.

- Our r-SNN model



Spiking Neuron Model

- **Leaky integrate-and-fire (LIF)** neurons for recurrent and supervised layers

- **Postsynaptic membrane potential ($\boldsymbol{U_i}$)** update:

$$\frac{dU_i}{dt} = \frac{U^{rest} - U_i}{\tau^{mem}} + I_i^{syn}(t)$$

- **Synaptic input current $\boldsymbol{I_i^{syn}(t)}$** update:

$$\frac{d}{dt} I_i^{syn}(t) = -\frac{I_i^{syn}(t)}{\tau^{syn}} + \sum_{j \in pre} w_{ij} S_j(t)$$

- When $U_i \geq \theta^{mem}$, **a spike was triggered** (i.e., when $S_i(t) = 1$). A refractory period followed for n^{ref} time steps.

Supervised Learning Rule

- The readout from the recurrent layer was trained using **a surrogate gradient approach** [1] that can learn using **precise spike times** in the LSM

- Inspired by SuperSpike [2], the RNN readout weight w_{ij} was updated according to **a nonlinear Hebbian rule**:

$$\Delta w_{ij} = \eta \cdot \left[\epsilon_j \otimes \left(\widehat{S_i} - \sigma(U_i)\right)\right] \cdot \sigma(U_i) \cdot \left(1 - \sigma(U_i)\right)$$

  - Note: ⊗ represents the cross product

- **Presynaptic trace $\boldsymbol{\epsilon_j}$** update:

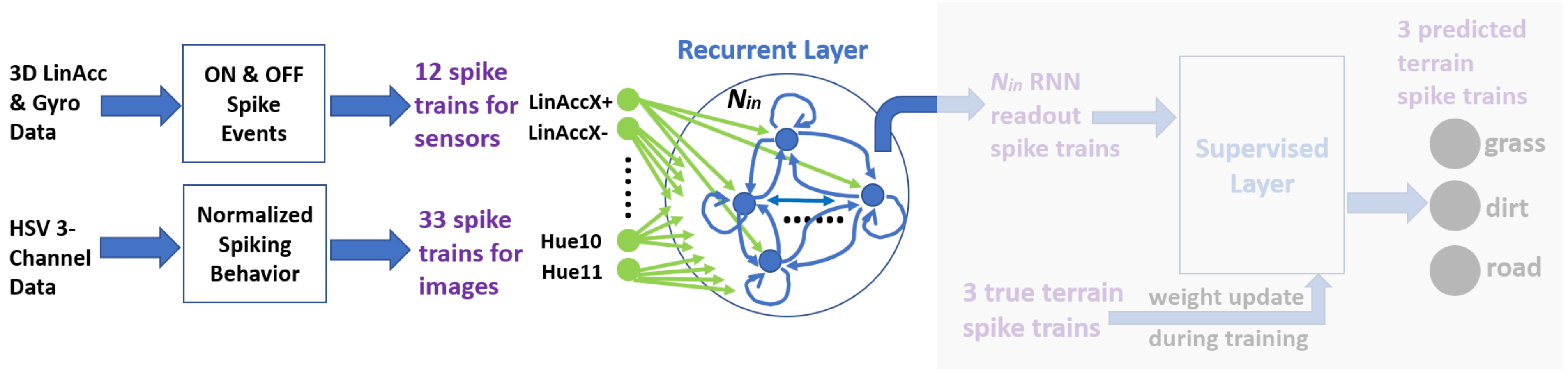
$$\frac{d\epsilon_j}{dt} = -\frac{\epsilon_j}{\tau^{syn}} + S_j(t)$$

References:

[1] EO Neftci, H Mostafa, and F Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51-63, 2019.
[2] F Zenke and S Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

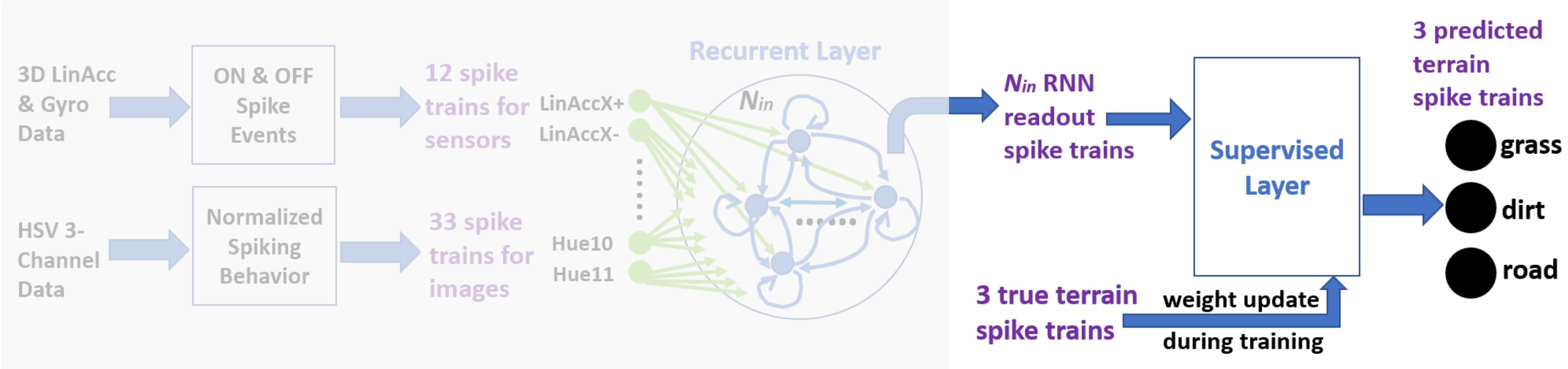
Recurrent Layer for Terrain Feature Extraction

- **12 sensor neurons** with "plus" and "minus" spike trains
    - **Threshold crossing method** applied to get **ON and OFF events**
- **33 image neurons** for hue, saturation, value (HSV) channels
    - Gaussian tuning curves over bottom-center 5-by-5 pixels
- N_{in} = 70 for recurrent neurons
- **Full connections** among input→recurrent and recurrent→recurrent neurons.
- $I_i^{syn}(t)$ <= **summation of input and recurrent weights** upon spike arrival
- **Random input and recurrent weights** from a Gaussian distribution with zero mean



Supervised Layer for Terrain Classification

- 70 RNN readout spike trains were fed into the supervised layer
- Output weights were updated for 100 training epochs and fixed during testing
- During training, $I_i^{syn}(t)$ <= **summation of output weights** upon spike arrival
- The postsynaptic neurons: **3 terrain prediction neurons**
- A terrain class was predicted according to the **highest activity** at that time step



Terrain Prediction Results for the r-SNN

- With both image and sensor (the linear accelerometer and gyroscope) inputs:

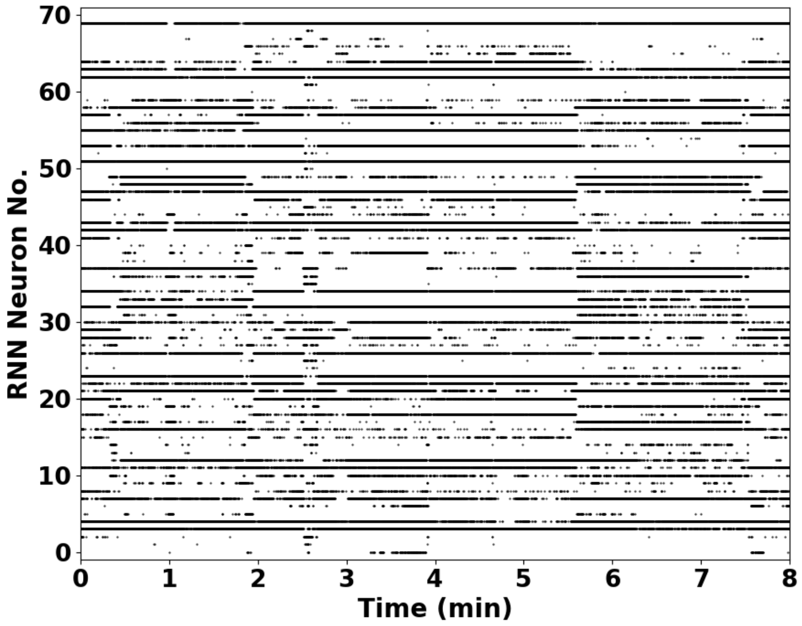


Fig. 4. Readout spikes from all the recurrent neurons.

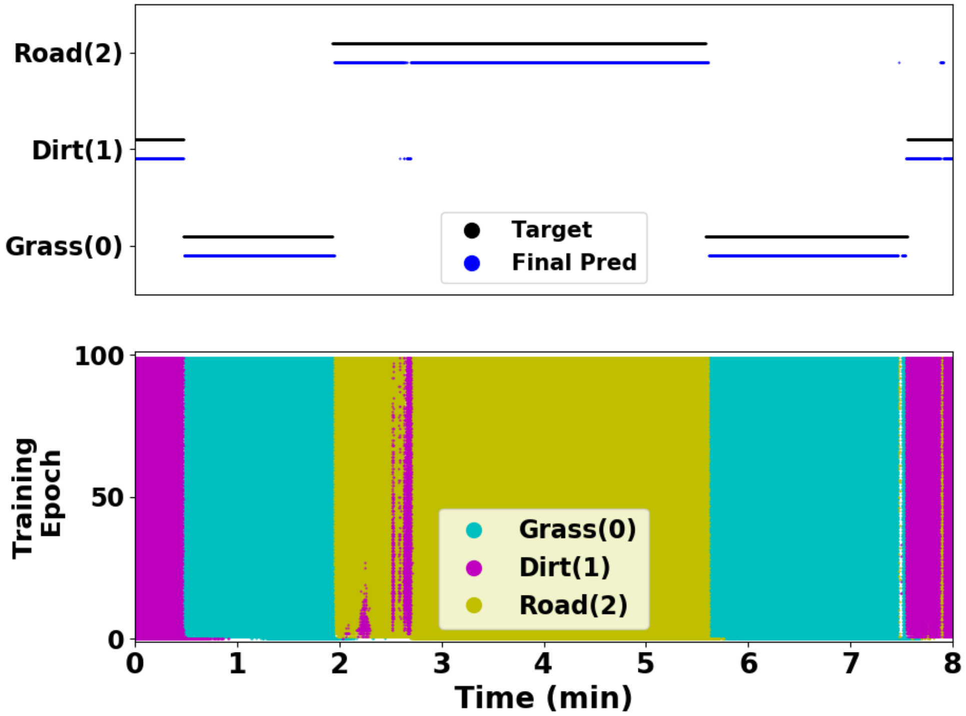


Fig. 5. Test prediction results after 100 training epochs (upper) and after each training epoch (lower)

Comparison Among Three Approaches

TABLE I
TEST ERROR RATES ON THREE MODELS FOR TERRAIN CLASSIFICATION.

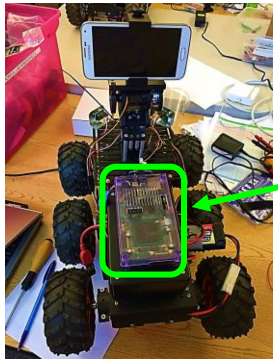
	r-SNN	SVM	3L Logistic Regression	
			mse	xent
Images only	**5.2%**	13.9%	11.5%	16.2%
Sensors only	**8.1%**	14.5%	13.7%	59.6%
Images + Sensors	**3.5%**	8.8%	10.2%	34.3%

- For both ML models, original signals were processed in 500-msec data chunks
  - For optimal performance, 9 features for SVM and 5 features for *3L logistic regression* [details in paper]
- The standard 80/20 rule applied for training and testing
- **Lowest test prediction error with images+sensors and r-SNN**
- **The r-SNN was the most efficient method:**
  - Only 70 RNN internal neurons
  - **Adaptation of only the RNN-to-output weights**
  - **No need of** data splitting into **time chunks**
  - With the **lowest computational cost** among the three
    - roughly 10^9 SynOps for r-SNN v.s. roughly $10^9 \sim 10^{10}$ MACs for SVM and 3L logistic regression
    - A SynOp consumes many fold less energy than a MAC.

Conclusion

- Our r-SNN approach is compatible with **event-driven and highly parallel** neuromorphic hardware.
- The r-SNN outperformed SVM and 3L logistic regression.
- **Having both image and sensor information** improved the learning performance.
- The r-SNN can be used to **augment a SLAM or GPS map** with metadata pertaining to the **cost of traversal**.
- We are developing a **complete neuromorphic robot navigation system** capable of operating **over long durations with minimal power consumption.**

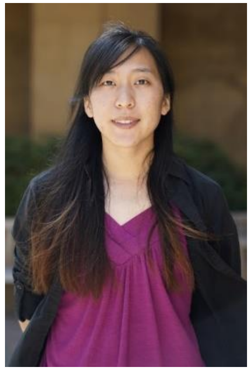
Android-Based Robot (ABR)



NS1e with IBM TrueNorth

Neuromorphic Implementation

(Hwu, Isbell, Oros, and Krichmar, *IEEE IJCNN*, 2017.)



Xinyun Zou
xinyunz5@uci.edu

Tiffany Hwu
tjhwu@hrl.com

Jeffrey Krichmar
jkrichma@uci.edu

Emre Neftci
eneftci@uci.edu

Thank you!