Spiking Neural Network For SAR Image Classification Through Reinforcement Learning and Spike-Time Dependent Plasticity

Robert Max Slick
Department Of Computer Science
Old Dominion University
Norfolk, USA
rslic001@odu.edu

***Abstract*—Synthetic aperture radar (SAR) imagery is a technique often applied to the task of identifying highly reflective objects due to to its operational ability to perform in inclement weather conditions and in the absence of visible light. One such application of SAR imagery is in the identification of vehicles. Traditionally, convolutional neural networks (CNNs) have been applied to the task of SAR image classification. However, CNNs are energy inefficient in many online learning applications making them often unsuitable for applications running on embedded systems or that are energy sensitive. Relative to CNNs, Spiking Neural Networks (SNNs), when coupled with emerging neuromorphic hardware, provide energy efficient computational ability with high biological plausibility. This makes SNNs suitable for intelligent computations on data in low-power and embedded applications. In this paper, we present a Spiking Neural Network implementation that utilizes reinforcement learning to enable low-power classification on the MSTAR vehicle dataset for embedded systems. All codes and data are made available at: https://github.com/rmslick/SNN-SAR.**

***Index Terms*—Spiking Neural Network, STDP, Reinforcement Learning, SAR, Despeckling**

I. Introduction

Synthetic Aperture Radar (SAR) originated in the 1950s. SAR is a high-resolution imaging radar that can operate in all-weather and day-and-night conditions [1]. It has become an important part of the ground and sea observation system, and the research focus of remote sensing technology. It is not only widely used in the various fields of national economic construction and ecological environment protection, but also plays an increasingly important role in the national security and military fields. Posessing the capability of collecting energy maps from radar backscatter, SAR provides a 2D representation of energy distributions on which common computer vision techniques can be applied to inform decision making. The domain applications of SAR data is increasing. Examples such as the satellite launch of the Gaofen-3, have enabled an increase in high-resolution SAR image data acquisition. This, in turn, has further promoted the development and application of the SAR image interpretation technology.

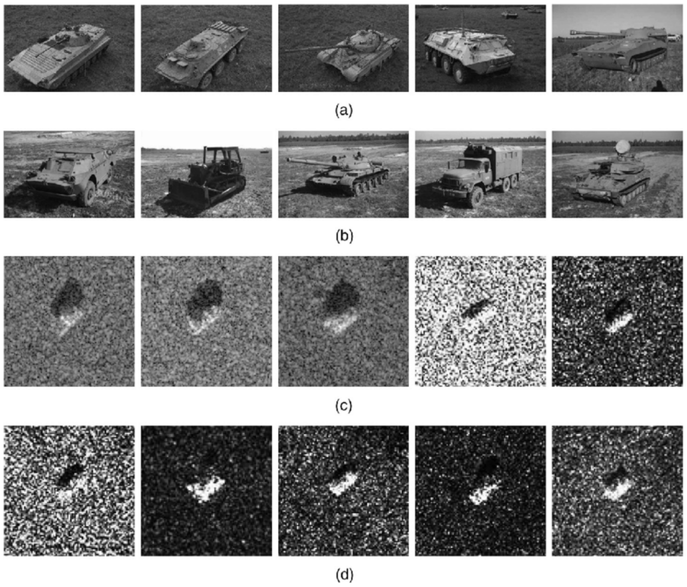

Fig. 1. MSTAR database. (a) and (b) Visible light images for BMP2, BTR70, T72, BTR60, 2S1, BRDM2, D7, T62, ZIL131, and ZSU23/4. (c) and (d) Corresponding SAR images for 10 targets measured at azimuth angle of 45 deg.

Synthetic Aperture Radar (SAR) is a technique that is used to create two-dimensional images from radar data. To generate SAR Images, the radar antenna is moved over a target region while emitting pulses in succession that are transmitted to illuminate a target scene. As the SAR device on board the aircraft or spacecraft moves, the antenna location relative to the target changes with time. Signal processing of the successive recorded radar echoes allows the combining of the recordings from these multiple antenna positions. This process forms the synthetic antenna aperture and allows the creation of higher-resolution images than would otherwise be possible with a given physical antenna. (Taken from [1])

The realization of Artificial Neural Networks (ANNs) in recent years has been applied with much success in the field of computer vision. With the increase in SAR technology the

Identify applicable funding agency here. If none, delete this.

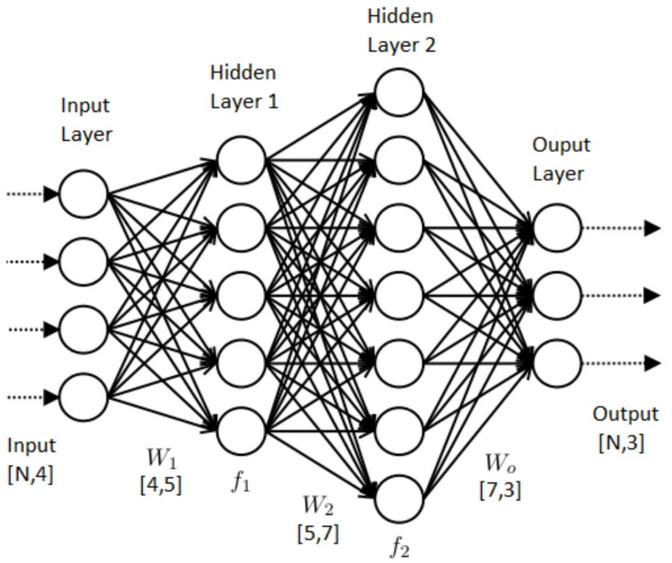

Fig. 2. Example Artificial Neural Network with two hidden layers.

ability to form high-resolution radar images has encouraged many to apply ANNs to leverage computer vision techniques for intelligent sensing of radar data. The preeminent technology enabling ANNs to accomplish a wide range of computational tasks intelligently is deep-learning. An Artificial Neural Network is a collection of artificial neurons interconnected by synapses with signals affected by trainable weights. An artificial neuron receives a signal, processes it and can signal neurons connected to it. Each neuron has an associated weight which increases or decreases during the learning process to strengthen or decrease the output signal. Through extending the amount of layers of neurons an Artificial Neural Network can perform automated feature extraction and learn the distribution of the data. The deep learning method represented by the neural network can learn image features independently without relying on the manual design [2]. It has achieved breakthroughs in many tasks such as classification, objection detection, and segmentation. In recent years there has been a push to develop networks that are more biologically plausible and those that are more energy efficient. Investigations into more biologically plausible networks has placed an emphasis on reconsidering the fundamental computing unit of ANNs - the neuron. In traditional ANNs the neuron is modeled by a linear, weighted sum. Signals connected to the input of the neuron are translated into a corresponding output function thereby creating a continuous function of information propoagation throughout the network. While this model has neabled the successes of backpropogation and other gradient based learning mechanisms, there is a concern that they are not adequately reflective of biological neurons. A neuron in a spiking neural network attempts to incorporate the notion of electrical charges to define their neuron. Each neuron maintains an electrical potential as a state. This value of the state is a decaying function of time. A neuron in a SNN produces output if and only if incoming potentials raise the state potential above a given threshold. A biologically plausible neuron models is a key ingredient of SNNs. A model proposed by Gerstner and Kistler (See Fig. 3) in 2002 represents a popular and reputed model for its capacity to capture informational dynamics observed among real biological neurons, and to represent and integrate several information dimensions (e.g. time, space, frequency, phase, and to deal with large volumes of data). [7] A SNN is composed of biologically plausible neuron models and synaptic weights. Similar to ANNs, the learning process exists to inform the update of the synaptic weights. The learning process itself, however, is different in consequence to the different information propogation approaches in each. The theory behind SNNs is currently mostly accepted to describe realistic brain-like information processing, which in addition eases their implementation on super-fast and reliable hardware platforms. Considered nowadays as the third generation of ANNs (Maass, 1997), the advent of SNN was propelled by the need for a better understanding of the information processing skills of the mammalian brain, for which the community committed itself to the development of more complex biologically connectionist systems. [7]

ANNs have progressivley been adapted into the deep-learning networks known today, which are composed of millions of neurons and synapses. Consequent to the increase in model complexity, the power required to train and use these models has increased significantly. This rise in power consumption has been a concern and has prompted various inquiries attempting to quantify how much power neural networks are consuming. According to the latest research [6], the process of training a common deep learning model can emit more than 626,155 pounds of carbon dioxide, which is equivalent to 5 times the emissions of a car's life cycle, 57 times the carbon emissions of a person in his lifetime. This introduces a stark problem into the field of machine learning: training machine learning models is computationally expensive and perhaps untenable as the application domain continues to grow that use increasingly complex models. Additional to the training process, online learning for embedded applications require offloading data to remote computation centers thereby posing an increase to global network traffic.

Consequent to the problem of energy inefficiency in modern ANNs, alternative intelligent solutions have been explored both at the algorithmic and platform level. One such platform that has emerged as a potential solution toward lower power computing is neuromorphic hardware. Neuromorphic hardware is the use of very-large-scale integration (VLSI) systems containing electronic analog circuits to mimic neuro-biological architectures present in the nervous system. Neuromorphic hardware designed specifically for Spiking Neural Networks has introduced a new low-power computing paradigm that offers significant reduction in energy consumption for intelligent computing. Developing Spiking Neural Networks to run on neuromophric computing platforms presents an entirely different solution than many power-reduction techniques

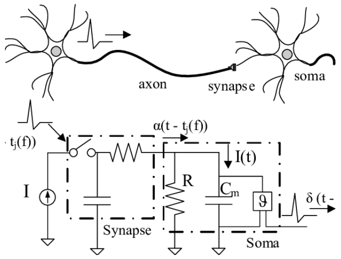

Fig. 3. Schematic diagram of the Leaky Integrate and Fire model (From [6]) (Gerstner W. Kistler W. M., 2002) Fig. 3 shows the schematic diagram of the basic Leaky Integrate and Fire model. Inside the dashed circle on the right side of Fig. 3 is the soma circuit. On the left side of Fig. 3, the synapse which contains the low pass filter circuit acts as the connection between the input spike (t-t j (f)) and the soma. The input spike is filtered out and a current (t-t j (f)) is the result of the process and gets transferred to the soma. The current I(t) charges the RC circuit thus changing the membrane potential of the neuron. The membrane potential of the neuron,
[8]

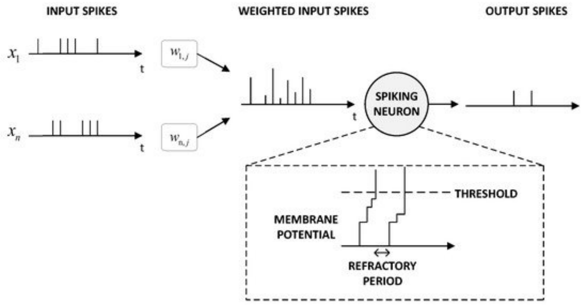

Fig. 4. Schematic representation of a spiking neuron. The neuron receives many input spikes from the previous layers' neurons. Each input spike is modulated with a weight, producing weighted synaptic inputs. Those inputs are accumulated as the neuron membrane potential. If the membrane potential crosses a certain threshold, the neuron will emit a spike to the downstream layer and the membrane potential will be reset. A refractory period, where the neuron remains inactive (i.e., does not further integrate any incoming spikes) after firing, may be implemented in some models.

that attempt to increase ANN eficiency. In SNNs, like in biological neural networks, neurons communicate with each other through isolated, discrete electrical signals (spikes), as opposed to continuous signals, and work in continuous instead of discrete time. Neuromorphic hardware (Indiveri et al., 2011; Esser et al., 2016; Furber, 2016; Thakur et al., 2018) is specifically designed to run such networks with very low power overhead, with electronic circuits that faithfully reproduce the dynamics of the model in real time, rather than simulating it on traditional von Neumann computers. Some of these architectures (including Intel's Loihi, IBM's TrueNorth, and SynSense's DynapCNN) support convolution operations, which are necessary for modern computer vision techniques, by an appropriate weight sharing mechanism. (Taken from [14])

Reinforcement learning is a machine learning training method based on rewarding desired behaviors and/or punishing undesired ones. In general, a reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error. (Taken from [20]) In reinforcement learning, developers devise a method of rewarding desired behaviors and punishing negative behaviors. This method assigns positive values to the desired actions to encourage the agent and negative values to undesired behaviors. This programs the agent to seek long-term and maximum overall reward to achieve an optimal solution. Reinforcement learning has found success in robotic applications such as can be found in [9] and [10]. Therefore the problem of finding a low-power reinforcement based solution for intelligent computations is crucial to success in embedded applications.

In this paper we address the problem of developing a low-power, semi-supervised learning algorithm for enabling low-cost SAR imaging applications. To this end we devise a Spiking Neural Network with Spike-timing Dependent Plasticity learning that uses lateral inhibition to achieve the task of classification in SAR images. The dataset we used for training is the popular MSTAR military vehicle dataset. The classification accuracy we achieved on our testing data was 92 percent.

II. Related Works

Owing to the demand for low-power intelligent computing in SAR applications there have been other investigations into the feasibility of a SNN solution to target classification. Here we list two other works that both attempt to use Spiking Neural Networks to perform classification on the MSTAR dataset. The authors of [21] present an unsupervised learning method for SAR image classification that uses STDP based learning to achieve an 84.2 percent accuracy on the MSTAR dataset. In 2021, the authors of [3] present 'SAR Image Classification Based on Spiking Neural Network through Spike-Time Dependent Plasticity and Gradient Descent' in this work the authors similarly use the MSTAR dataset and use a Gradient Descent method that uses the output spike sequences of an unsupervised learning network as teaching signals. Their method achieves a 90.05 percent accuracy on their test dataset. The authors in this method do not account for despeckling explicitly in the SAR data. With our despeckling pre-processing step and reinforcement learning we are able to achieve a higher classification accuracy. To our knowledge we are the first to provide a fully implemented Spiking Neural Network in PyTorch that uses reinforcement learning to perform classification on the MSTAR dataset.

III. Methods

For our implementation we utilize SpykeTorch [2], a library for SNN architecture implentation that wraps around PyTorch. The structure of this network is similar those proposed by Masquelier and Thorpe (PLoS Comput Biol, 2007) and Mozafari et al (IEEE TNNL, 2018) [19] with some amendments made to the network parameters and additional filtering for speckle denoising. The architecture for our Spiking Neural Network is composed of a despeckling method, an conversion layer, a pooling method and a trainable convolutional layer that employs STDP learning rules.

A. Neuron Model

The main distinction between SNNs and ANNs is their representation of the fundamental computing unit, the neuron. There have been many attempts to model more accurately the behavior of the biological neuron. Hodgkin and Huxley [15] were among the first to propose a model with extensive biological detail; however, it comes with a high computational cost. Izhikevich [16] proposed a neuron model that can capture diverse dynamic behaviors of a biological neuron while keeping a moderate computational efficiency. The Leaky Integrate and Fire (LIF) and the simplified version Integrate and Fire (IF) are among the most popular models, as they capture the intuitive property of accumulating charge on the neuron's membrane, with a constant leaky charge and a clear threshold.The LIF is easily modeled and implemented in hardware and has been applied in a wide range of applications. In Fig. 3 we list the LIF neuron circuit model, In Fig. 4 we list an intuitive interpretation of the spiking process along with accompanying descriptions of each.

B. Synthetic Aperture Radar Image Despeckling

Owing to the reflective property of radar imagery, SAR imagery is prone to energy disturbances, or speckles, which add noise to the image and pose a challenge to inference methods. A speckle is a granular disturbance, usually modeled as a multiplicative noise, that affects synthetic aperture radar (SAR) images, as well as all coherent images. [11] Speckle may severely diminish the performances of automated scene analysis and information extraction techniques, as well as it may be harmful in applications requiring multiple SAR observations, like automatic multi-temporal change detection. For these reasons, a preliminary processing of real-valued detected SAR images aimed at speckle reduction, or despeckling, is of crucial importance for a number of applications. In our method we employ Bilateral Filtering as a preprocessing step to accomplish denoising of the incoming SAR image. The basic idea underlying bilateral filtering is to do in the range of an image what traditional filters do in its domain. Two pixels can be close to one another, that is, occupy nearby spatial location, or they can be similar to one another, that is, have nearby values, possibly in a perceptually meaningful fashion. The bilateral filter replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. [12]

C. Spike Encoding

The neuron model underpinning the SNN dictates that all information fed into the network be encoded into a sequence of potentials or spikes. Understanding the mechanism by which the brain transmits and represents information is still an area of active research. Neural coding is the attempt to encode information in a manner similar to the brain to enable similar transmissions. Neural coding plays an essential role in enabling the spiking neural networks (SNNs) to perform different tasks. Among many different encoding techniques there are four common neural coding schemes, namely: rate coding, time-to-first spike (TTFS) coding, phase coding, and burst coding. Here we employ a rate coding scheme to convert our input images into spike tensors. Rate coding is the most widely used coding scheme in neural network models. It should be noted that the choice of encoding scheme may impact the results of the network significantly. The rate coding scheme considers each input pixel as a firing rate and converts the pixel into a Poisson spike train with the firing rate. [15] Each of our images is first converted into a spike wave tensor using the rate encoding scheme before being processed. Following the encoding step we apply a Gabor filter over the spike time tensor. It was shown by several researchers that the profile of simple-cell receptive fields in the mammalian cortex can by described by oriented two-dimensional Gabor functions (Taken from [13]). We use a set of four Gabor filters as a preprocessing step for estimation of stereo disparity and preliminary extraction of oriented image features.

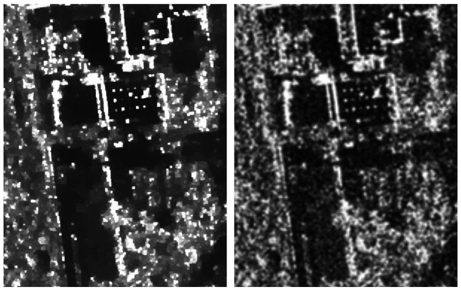

Fig. 5. Example of the effects of noise in Synthetic Aperture radar imagery and the effects of applying a despeckling method.[18]

D. Pooling

Pooling is an important operation in deep convolutional networks for the process of feature extraction. SpykeTorch implements a two-dimensional max-pooling operation that is similar to the commonly used max pooling methods. Using the pooling objects requires providing a pooling window size. The stride is equal to the window size by default, but it is adjustable. Here we select a window size of 2.

The pooling objects provided by SpykeTorch are applicable to both spike-wave and potentials tensors. According to the structure of these tensors, if the input is a spike-wave tensor, then the output will contain the earliest spike within each pooling window, while if the input is a potentials tensor, the maximum potential within each pooling window will be extracted.[2]

E. Input Layer

The goal of this layer is to extract oriented edges from the gray scaled input image and turn them into spike latency's. To this end, the input image is convolved with Gabor filters of four different orientations. Thus, this layer includes four feature maps, each representing the saliency of edges in a particular preferred orientation. [19]

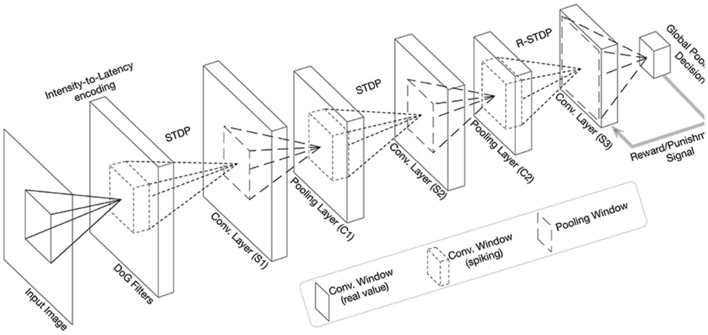

Fig. 6. Sample deep SNN that shows the interrelation between various layers found in this work.

F. Convolutional Layer

Our first complex layer is a local pooling layer over the spikes coming from layer S1. Here, there are four 2D neuronal grids corresponding to each of the orientations. Each C1 neuron performs a local pooling operation over a window of size 2x2 and stride 2 on S1 neurons in a particular grid, after which, it emits a spike immediately after receiving its earliest in- put spike. This pooling operation decreases the redundancy of layer S1, and shrinks the number of required neurons, which consequently increases the computational efficiency. It also adds a local invariance to the position of oriented edges. [19]

G. Reward Based STDP

We utilize the proposed reinforcement learning mechanism to update the pre-synaptic weights of S2 neurons as in [19]. In this work, the magnitude of weight change is modulated by a reward/punishment signal, which is received according to the correctness/incorrectness of the network's decision. As in [19], we also applied a one-winner- takes-all learning competition among the S2 neurons, by which the one with the earliest spike is the winner and the only one which updates its synaptic weights. This neuron is the one determining the network's decision.

IV. Experiment

A. Dataset

For our dataset we used the MSTAR Public Target Synthetic Aperture Radar data. The data set was obtained in September of 1995 by the Sandia National Laboratory SAR sensor platform. The collection was jointly sponsored by DARPA and Air Force Research Laboratory as part of the Moving and Stationary Target Acquisition and Recognition (MSTAR) program. SNL used an X-band SAR sensor in one foot resolution spotlight mode. The MSTAR dataset is composed of SAR imagery of military vehicles. The vehicles names along with their capture information is listed in figur8. .

To test our classification network we selected the BMP-2 vehicle images and the T-72 vehicle images to compose our training and test datasets.

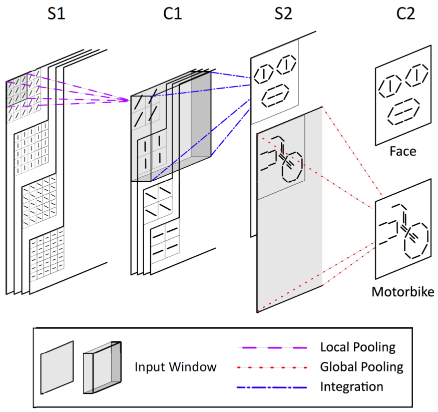

Fig. 7. Overall structure of the network adopted from [19] with four retinotopically organized layers. The first layer (S1) extracts oriented edges from the input image by applying Gabor filters. A local max-pooling operation is applied by the cells in the subsequent layer (C1) to gain some degrees of position invariance. From here, spikes are propagated by the latencies that are inversely proportional to the maximum val- ues. These spikes are the inputs for the IF neurons in the layer S2 that are equipped with the R-STDP learning rule. These neurons are encouraged/punished to learn/unlearn complex features. The activity of S2 neurons are used by C2 neurons for decision-making. These neu- rons are associated with class labels and the decision is made based on the neuron with the earliest spike.[19]

Target	Train		Test	
	Depression	No. Images	Depression	No. Images
BMP-2	17°	699	15°	587
BTR-70	17°	233	15°	196
T-72	17°	699	15°	588
BTR-60	17°	256	15°	196
2S1	17°	299	15°	274
BRDM-2	17°	299	15°	274
D7	17°	299	15°	274
T-62	17°	299	15°	274
ZIL-131	17°	299	15°	274
ZSU-234	17°	299	15°	274

Fig. 8. Overview of the set of all vehicles, number of training vehicles and capture angles for the MSTAR dataset.

B. Baseline

To make a proper comparison between our Spiking Neural Network, we identified an existing Spiking Neural Network designed for the same task of MSTAR data classification that utilized the same vehicles for their training and testing datasets. From the paper titled 'SAR Image Classification Based on Spiking Neural Network through Spike-Time Dependent Plasticity and Gradient Descent'[4], these authors completed a SAR image classifier based on unsupervised and supervised learning of SNN by using spike sequences with complex spatio-temporal information. They firstly expound the spiking neuron model, the receptive field of SNN, and the construction of spike sequence. Then we put forward an unsupervised learning algorithm based on STDP and a supervised learning algorithm based on gradient descent. We were able to duplicate their results by downloading their

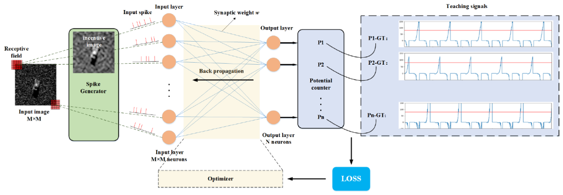

Fig. 9. Depiction of a sample SAR image passing through a SNN as described in [16]. This exemplifies the depiction of an image as a series of spikes.

source code linked within their paper. This paper, listed in the Related Works section above, similarly seeks to perform classification on the bmp2 and the t-70 vehicles taken from the MSTAR dataset.

Our training and testing set, like that used by Chen, et al. in [4] consists of the training set including the t-70 and the bmp2 military vehicles. For our results, we report the classification accuracy's on the aggregate test set provided by the MSTAR dataset for these vehicles. From the results below it can be seen that our network outperforms the published gradient descent method results from [4] for target classification on the provided testing sets from MSTAR. These figures represent the reported percent correctly classified from the testing dataset provided by MSTAR for the above-mentioned vehicles.

Reinforcement SNN	Gradient SNN
0.92	0.90

Summary and Future Work

With this network we hope to explore a hardware, software optimization to study the cost of power for our Spiking Neural Network. In addition, we wish to explore the development of a Generative Adversarial Network implemented with Spiking Neural Network to substitute our despeckling method. With the current network we hope to work towards achieving more computer vision techniques such as object detection and semantic segmentation on SAR data.

References

[1] "Introduction to Airborne RADAR", G. W. Stimson, Chapter 1 (13 pp).

[2] Mozafari, M., Ganjtabesh, M., Nowzari-Dalini, A. and Masquelier, T., 2021. SpykeTorch: Efficient Simulation of Convolutional Spiking Neural Networks With at Most One Spike per Neuron.

[3] J. Chen, X. Qiu, C. Ding, and Y. Wu, "SAR Image Classification Based on Spiking Neural Network through Spike-Time Dependent Plasticity and Gradient Descent," 2021. Accessed: Dec. 16, 2021. [Online]. Available: https://arxiv.org/pdf/2106.08005.pdf.

[4] Chen, et al., Supervised SNN Based on Gradient Descent in TensorFlow, GitHub repository, Mar. 12, 2021, [Online] Available: https://github.com/Jiankun-chen/Supervised-SNN-with-GD.

[5] [1]D.-A. Nguyen, X.-T. Tran, and F. Iacopi, "A Review of Algorithms and Hardware Implementations for Spiking Neural Networks," Journal of Low Power Electronics and Applications, vol. 11, no. 2, p. 23, May 2021, doi: 10.3390/jlpea11020023.

[6] R. Toews, "Deep Learning's carbon emissions problem," Forbes, 10-Dec-2021. [Online]. Available: https://www.forbes.com/sites/robtoews/2020/06/17/deep-learnings-climate-change-problem/?sh=2f3880f56b43. [Accessed: 17-Dec-2021].

[7] J. L. Lobo, J. Del Ser, A. Bifet, and N. Kasabov, "Spiking Neural Networks and online learning: An overview and perspectives," Neural Networks, vol. 121, pp. 88–100, Jan. 2020, doi: 10.1016/j.neunet.2019.09.004.

[8] N. D. Bui Phong, M. Daneshtalab, S. Dytckov, J. Plosila and H. Tenhunen, "Silicon synapse designs for VLSI neuromorphic platform," 2014 NORCHIP, 2014, pp. 1-6, doi: 10.1109/NORCHIP.2014.7004745.

[9] [1]S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End Training of Deep Visuomotor Policies," Journal of Machine Learning Research, vol. 17, pp. 1–40, 2016, [Online]. Available: https://arxiv.org/pdf/1504.00702.pdf.

[10] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-End Training of Deep Visuomotor Policies," Journal of Machine Learning Research, vol. 17, pp. 1–40, 2016, [Online]. Available: https://arxiv.org/pdf/1504.00702.pdf.

[11] F. Argenti, A. Lapini, T. Bianchi, and L. Alparone, "A Tutorial on Speckle Reduction in Synthetic Aperture Radar Images," IEEE Geoscience and Remote Sensing Magazine, vol. 1, no. 3, pp. 6–35, Sep. 2013, doi: 10.1109/mgrs.2013.2277512.

[12] "Bilateral Filtering," homepages.inf.ed.ac.uk. https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html (accessed Dec. 16, 2021).

[13] "Active Stereo Vision," homepages.inf.ed.ac.uk. https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TRAPP1/filter.html (accessed Dec. 16, 2021).

[14] M. Sorbaro, Q. Liu, M. Bortone, and S. Sheik, "Optimizing the Energy Consumption of Spiking Neural Networks for Neuromorphic Applications," Frontiers in Neuroscience, vol. 14, Jun. 2020, doi: 10.3389/fnins.2020.00662.

[15] W. Guo, M. E. Fouda, A. M. Eltawil, and K. N. Salama, "Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems," Frontiers in Neuroscience, vol. 15, Mar. 2021, doi: 10.3389/fnins.2021.638474.

[16] "(PDF) An Efficient Feature Selection for SAR Target Classification," ResearchGate. https://www.researchgate.net/publication/325056647_An_Efficient_Feature_Selection_for_SAR_Target_Classification (accessed Dec. 16, 2021).

[17] J. Li, H. Xie, G. Wang and Z. Chen, "Target Classification Using Spiking Neural Networks for SAR Images," 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), 2021, pp. 886-890, doi: 10.1109/AEMCSE51986.2021.00180.

[18] Ozcan, Caner, Sen, Baha Nar, Fatih. (2014). GPU efficient SAR image despeckling using mixed norms. Proceedings of SPIE The International Society for Optical Engineering. 9247. 10.1117/12.2067074.

[19] Mozafari, M., Kheradpisheh, S. R., Masquelier, T., Nowzari-Dalini, A., Ganjtabesh, M. (2018). First-Spike-Based Visual Categorization Using Reward-Modulated STDP. IEEE Transactions on Neural Networks and Learning Systems (https://doi.org/10.1109/TNNLS.2018.2826721).

[20] J. M. Carew, "What is reinforcement learning? A comprehensive overview," SearchEnterpriseAI, 29-Mar-2021. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning.

[21] J. Chen, X. Qiu, C. Han and Y. Wu, "Unsupervised Learning Method for SAR Image Classification Based on Spiking Neural Network," EUSAR 2021; 13th European Conference on Synthetic Aperture Radar, 2021, pp. 1-4.
[Accessed: 17-Dec-2021].