Robust Signal Classification Using Siamese Networks

Zachary Langford
Virginia Tech
Blacksburg, VA
zlangford@vt.edu

Logan Eisenbeiser
Virginia Tech
Blacksburg, VA
loganryn@vt.edu

Matthew Vondal
Virginia Tech
Arlington, VA
mvondal@vt.edu

ABSTRACT

We propose a noise-robust signal classification approach using siamese convolutional neural networks (CNNs), which employ a linked parallel structure to rank similarity between inputs. Siamese networks have powerful capabilities that include effective learning with few samples and noisy inputs. This paper focuses on the advantages that siamese CNNs exhibit for classification of quite similar wireless signal emitters across signal-to-noise ratio (SNR) and dataset size. Without any *a priori* information, candidate siamese and baseline CNNs were trained on compressed spectrogram images to distinguish modulated signal pulses with randomized symbols and identical signal parameters, save for slight frequency offsets commonly exhibited in commercial RF emitter reference oscillator uncertainty distributions. Compared with baseline CNN approaches the proposed methods demonstrate improved classification performance under poor SNR. Moreover, this advantage holds the potential for superior, low-SNR, semi-supervised classification using embeddings from within the networks.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; • **Security and privacy →** *Mobile and wireless security.*

KEYWORDS

Siamese networks, convolutional neural networks, signal classification, specific emitter identification

ACM Reference Format:
Zachary Langford, Logan Eisenbeiser, and Matthew Vondal. 2019. Robust Signal Classification Using Siamese Networks. In *ACM Workshop on Wireless Security and Machine Learning (WiseML2019), May 15–17, 2019, Miami, FL, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3324921.3328781

1 INTRODUCTION

1.1 Signal Classification Machine Learning

Learning to distinguish similar signal emitters in a monitored radio frequency environment with limited *a priori* signal information is often a challenging classification problem. In many practical cases a surveillance receiver may be tasked with learning to classify received signals based upon a small number of noisy labeled samples of each emitter. Classical single receiver approaches often focus on discerning differences in signal patterns or internal data structures. For example, Tang et al. [16] applied an artificial neural network in conjunction with cyclostationarity-based spectrum sensing to detect signals at low SNR values. However, these and similar methods tend to work by carefully hand-crafting specialized feature extractors (i.e., "expert features") for specific signal types, which are often fragile outside particular conditions or rely on computationally inefficient preprocessing steps [13].

Signal classification and emitter identification are crucially important in a number of commercial and military applications such as authentication, threat detection, and situational awareness [2]. General signal classification can be a difficult task due to a non-cooperative environment where various factors like interference, low SNR, fading, phase and frequency offsets cause distortion to the received signal [9]. In recent work, signal classification has been accomplished by various supervised and unsupervised learning techniques [2]. These machine learning (ML) approaches can range from support vector machines (SVMs) [14] to unsupervised Bayesian non-parametric learning [7]. Deep neural networks, which consist of many hidden layers [5], have shown promise for feature learning directly on raw, high-dimensional input data [13].

For the considered scenarios, we employ RF machine learning for specific emitter identification of devices transmitting with nominally identical signal parameters, except for slight frequency offsets commonly exhibited by commercial RF oscillators. The modulated signal pulses all employ randomized symbols. Thus, no internal data structures or features provide any emitter-discerning information. Moreover, the learning classifier operates directly on compressed spectrogram images without any synchronization or preprocessing feature formation that might favor learning the frequency offset as the relevant distinguishing factor.

1.2 Siamese CNN RF Signal Classification

Convolutional neural networks (CNNs) are a type of neural network that is useful for processing data that has a degree of spatial correlation between local data points, sometimes referred to as a grid-like topology [5]. CNNs have shown to be effective for signal classification with data presented as spectrograms [12]. However, CNNs have several potential issues, such as requiring large amounts of data for training, tending to overfit to particular classes [3], and vulnerability to adversarial examples [6]. As a result, systems that incorporate standard CNN architectures fail to offer robust solutions for noisy datasets [6].

Siamese CNNs hold some promise for overcoming some of the conventional CNN drawbacks in signal classification. They accept a pair of inputs to measure their similarity, and were originally

WiseML 2019, May 15–17, 2019, Miami, FL, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6769-1/19/05…$15.00
https://doi.org/10.1145/3324921.3328781

used in image verification applications. Moreover, the networks we employ can be trained with standard optimization methods on input pairs sampled from source datasets. Chopra et al. [4] presented a Siamese architecture based on convolutional networks that learned a discriminative feature embedding and a similarity measurement. Siamese CNNs have recently been used on imagery datasets such as scenery hyperspectral remote sensing to mitigate the impact of noisy datasets and few samples per training class [11].

1.3 Siamese Approach Advantages

The proposed siamese CNN approach provides key benefits:

(1) Several structural advantages over conventional CNN approaches, such as limited labeled training data and reduced susceptibility to overfitting certain classes
(2) A learned feature space for classification that is more robust to input samples with low SNR
(3) The potential for superior, low-SNR, semi-supervised classification using final network-layer embedding

2 DATASETS AND LEARNING METHODS

2.1 Dataset Generation

Signal Generation and Formatting. The signals are modeled upon push-to-talk and satellite communications handset transmissions. Each signal pulse is composed of 100 QPSK symbols (RRC filter, 0.4 excess bandwidth) that are randomized across all pulse instances to exhibit no consistent internal pulse structure. The generated complex baseband signals are sampled at 4 samples per symbol. Besides the symbol randomization, the signals from the emitters have identical parameters, save for frequency offsets that are fixed over each pulse duration. These frequency offsets are uniformly spaced to model L-band emitters with reference oscillators spanning a 0.5 parts-per-million (ppm) total range. AWGN and uniform random time offsets reflect channel noise and the absence of any time synchronization preprocessing. Unlike some other imagery applications, the spectrogram dataset generation does not apply skew or other transform distortions that do not have physical parallels in this scenario. Finally, the signal is transformed into a log10 intensity scale spectrogram plot that is resized to a compressed 100x100 pixel image for classifier input. Figure 1 shows an example of a 4.0 dB SNR pulse spectrogram. For these scenario performance metrics we constrain the incoming images to contain single complete pulses without partials or multiple pulse collisions. No synchronization or correction is applied to the incoming pulses prior to classification.

Training and Testing Data Segmentation. Table 1 categorizes the datasets used for training and testing, a total of 4000 spectrogram images. The dataset split was targeted to provide joint samples of classes and SNR values. We experimented with two sets of data, splitting either 75% or 25% for training. This was done to compare how sensitive the algorithms were to sample size. For the Siamese CNN, two images from the same class are called a positive pair, and those from different classes are called a negative pair. Each positive or negative image pair forms an input. The model weights were saved during the training process based on the best validation accuracy measures from the test dataset.

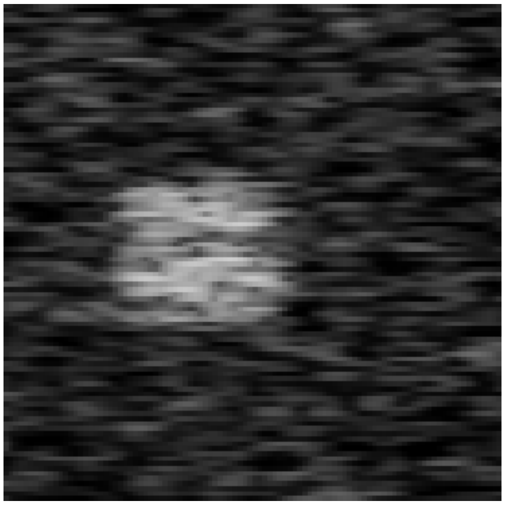


Figure 1: Example signal pulse spectrogram as a compressed grayscale image

Table 1: Dataset segmentation for two scenarios, where Tr is training and Te is testing

	Tr 75%	Te 25%	Tr 25%	Te 75%
SNR Type				
SNR=-6.0 dB	750	250	250	750
SNR=-4.0 dB	750	250	250	750
SNR= 4.0 dB	750	250	250	750
SNR= 6.0 dB	750	250	250	750
Signal Class				
Class 0=-0.17 ppm	760	240	228	772
Class 1= 0.00 ppm	752	248	248	752
Class 2= 0.17 ppm	732	268	268	732
Class 3= 0.33 ppm	756	244	256	744

2.2 Siamese CNN Structure and Metrics

A siamese network is composed of twin convolutional neural networks that share parameters and weights. Figure 2 illustrates the proposed siamese CNN. The parameters of both linked CNNs are jointly updated through backpropagation by a loss function, which computes a particular metric between the feature representations of each CNN. Sharing weights causes similar input images to be mapped to similar positions in feature space [11].

Contrastive Loss Function. A common siamese CNN form learns the embedding by minimizing the contrastive loss, which is defined over the distance between the outputs from the two identical inner neural networks [4]. Let x_1 and x_2 be a pair of spectrogram inputs to the CNN and y be a binary label of the pair, where $y = 0$ if the spectrograms belong to the same pairs and $y = 1$ if belonging to different pairs. Let W be the shared parameter vector that is subject to learning and let $G_W(x_1)$ and $G_W(x_2)$ be the two points in the low-dimensional space that are generated by mapping x_1 and x_2. Then the siamese network can be regarded as a scalar metric

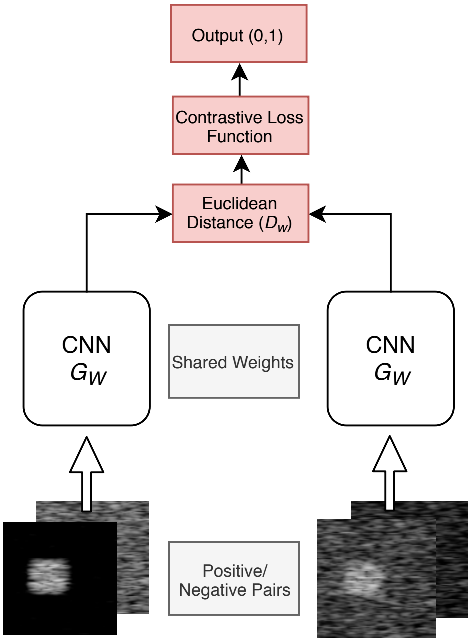


Figure 2: Siamese CNN architecture

function $D_W(x_1,x_2)$ to measure the compatibility between x_1 and x_2, and the Euclidean distance, D_W. Designed such that minimizing L with respect to W would result in low values of D_W for similar pairs and high values for dissimilar pairs [4]:

$$D_W(x_1,x_2) = \|G_W(x_1) - G_W(x_2)\|_2 \tag{1}$$

To simplify notation $D_W(x_1,x_2)$ will be represented as D_W. Then the contrastive loss is implemented by:

$$\mathcal{L}\sum_{i=1}^{p} L(W,(x_1,x_2)^i) \tag{2}$$

$$L(W,(y,x_1,x_2)^i) = (1-y) + L_s(D_W^i) + yL_D(D_W^i) \tag{3}$$

where (y,x_1,x_2^i) is the i-th labeled sample pair, L_S is the partial loss function for a pair of similar points, L_D is the partial loss function for a pair of dissimilar points, and P is the number of training pairs. L_S and L_D must be designed such that minimizing L with respect to W would result in low values of D_W for similar pairs and high values for dissimilar pairs [4].

3 OPTIMIZATION

We performed hyperparameter optimization on the exercised model. Table 2 lists the final layers and parameters for the CNN. The baseline CNN model includes an additional layer for classification, which uses the cross-entropy loss function for the softmax function [1]. The CNN network applies five convolutional (ReLU activation) layers to the output feature maps followed by max pooling with a filter size and stride of 2. This CNN architecture was inspired by [10] that uses a siamese CNN for one-shot learning handwritten characters.

The metric we set to measure the optimization of a given model was the Validation Accuracy. Our hyperparameter search boundaries are similar to the search boundaries given in [10], and were capped at 20 trials. The algorithm employed has a model type of GP_MCMC (Gaussian process Markov Chain Monte Carlo) and an acquisition function of EI_MCMC (Expected Improvement Markov Chain Monte Carlo).

SHERPA Framework. We decided to select the SHERPA optimization framework because it is highly configurable and allowed us to explore a large range of hyperparameters [8]. We used the Bayesian Optimization framework in SHERPA, which is based upon GPyOpt, a Gaussian process optimization library. We focused primarily on the filter size as kernel size for our hyperparameter optimization, but we did also explore different numbers of convolutional layers and slightly modified network structures. The ten layer structure outlined in Table 2 is the structure we found to exhibit the best performance during tuning. We also tested a small selection of learning rates, held constant throughout the layers, chosen from the following set: $\{10^{-3}, 10^{-4}, 10^{-5}\}$. We found 10^{-3}, or 0.001, to be the optimal learning rate for our configurations, even with the varying of other hyperparameters.

Filter Size. We varied the filter size in the range of [16, 256] with step sizes of 16, using the Ordinal designation in SHERPA to select only the given values. The filter size for each of the five layers was allowed to vary independently. For the examined CNN model the highest accuracies were observed with filter sizes of (32, 64, 128, 64, 32) for the five convolutional layers.

Kernel Size. In general, we varied the kernel size from 3x3 to 11x11, but with constraints. To prevent dimensionality errors in the neural network, we manually narrowed the optimization window for each layer. The ranges are as follows.

- Kernel 1: 3x3 - 11x11
- Kernel 2: 3x3 - 10x10
- Kernel 3: 3x3 - 7x7
- Kernel 4: 3x3 - 5x5
- Kernel 5: 3x3 - 5x5

There was an additional constraint put on all of the kernel sizes except for the first one, where every layer must be less than or equal in size to the kernel size of the layer before it. Our optimal kernel sizes for the baseline CNN were 3x3 for all five layers. This is likely due to the first layer being optimized at a kernel size of 3x3 and the remaining layers being constrained to the same size.

4 RESULTS

Table 3 and Table 4 compare classification accuracy for the siamese CNN and baseline CNN approaches, sorted alternately by SNR and signal class, with the former listing the 25% training dataset split and the latter listing the 75% dataset split. The baseline CNN struggles to provide any classification value in the SNR states below 0 dB. In both cases, the siamese CNN learns a better classifier for low SNR conditions and three of the four signal classes overall. It also exhibits improvement in accuracy when provided more training data. This consistency for the siamese CNN and concentrated over-performance in one category for the baseline CNN suggest that the baseline model overfits to these particular signal and SNR types,

Table 2: Structure of the Baseline CNN model (G_W)

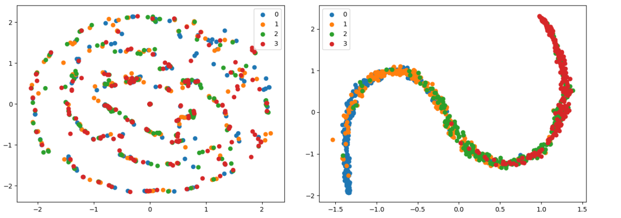
Layer	Name	Size
1	Convolutional-ReLU	32 filters size of 3x3
2	Convolutional-ReLU	64 filters size of 3x3
3	Max-Pooling	2x2
4	Convolutional-ReLU	128 filters size of 3x3
5	Max-Pooling	2x2
6	Convolutional-ReLU	64 filters size of 3x3
7	Max-Pooling	2x2
8	Convolutional-ReLU	32 filters size of 3x3
9	Max-Pooling	2x2
10	Full-Connected	512

Table 3: Classification accuracies using 25% for training

	CNN	Siamese CNN
SNR Type		
SNR=-6.0 dB	24.40%	64.08%
SNR=-4.0 dB	25.33%	68.34%
SNR= 4.0 dB	97.87%	96.43%
SNR= 6.0 dB	99.47%	99.31%
Signal Type		
Class 0=-0.17 ppm	50.00 %	82.12%
Class 1= 0.00 ppm	49.47 %	80.05%
Class 2= 0.17 ppm	98.09 %	82.10%
Class 3= 0.33 ppm	50.67 %	82.39%

Table 4: Classification accuracies using 75% for training

	CNN	Siamese CNN
SNR Type		
SNR=-6.0 dB	24.00 %	81.14%
SNR=-4.0 dB	24.00 %	80.51%
SNR= 4.0 dB	99.20 %	98.73%
SNR= 6.0 dB	100.00 %	99.79%
Signal Type		
Class 0=-0.17 ppm	99.58 %	88.75%
Class 1= 0.00 ppm	50.00 %	90.32%
Class 2= 0.17 ppm	49.63 %	87.69%
Class 3= 0.33 ppm	50.00 %	90.57%



(a) CNN

(b) Siamese CNN

Figure 3: Embedding plot using SNR=-4.0 dB for CNN baseline (a) and siamese CNN (b)

emphasizing that robustness is an issue when classifying noisy and unseen signals.

Due to the difference between the CNN and the Siamese loss functions, we will also compare and contrast the internal layer representations for some insight into their behavior in noisy conditions. Figure 3 (a) and (b) show the high-dimensional representations of the dense layer (layer 9 from Table 2) for the baseline CNN and siamese CNN, respectively, using t-SNE for visualization [17] with input -4.0 dB SNR for each signal category. Though we should not read too much into the t-SNE method's output (which can be sensitive to its tunable parameters), a superficial impression of the plots suggest that siamese CNNs can develop a better decision boundary for classification compared to the baseline CNN. Future work may investigate using similarity metrics in the embedded space to generalize beyond the trained dataset and differentiate previously unseen classes.

5 CONCLUSION

This study applied two approaches to signal classification - a baseline convolutional neural network and a siamese neural network. Framed as a scenario for specific emitter identification of Radio Frequency devices, each signal category represented a small frequency offset such as those exhibited in commercial handset oscillators. In all other ways the signal class parameters were the same, save for randomized modulated symbol content. Compressed signal spectrogram images formed the input datasets, beyond which no other preprocessing or feature formation was employed. Both classification approaches trained and tested on a mix of relatively few noisy samples, reflecting realistic scenarios with a limited number of available data samples per class.

The results showed consistency for the siamese convolutional neural network across SNR range and signal class, while the baseline CNN performed poorly under low SNR conditions. Though some initial optimization was performed, a more comprehensive optimization approach, such as variations on Bayesian optimization, could lead to better performance [10].

Recent progress on siamese CNNs indicate other distance metrics and loss functions could yield better results [10, 11]. For example, Koch et al. [10] used a weighted L_1 distance between the twin feature vectors combined with a cross-entropy loss. Additionally, other few-shot learning algorithms may be suited for signal classification, such as prototypical networks [15], matching networks [18], or triplet networks [19].

Trainable similarity metrics have numerous applications beyond the examples described in this paper. Siamese networks, along with other limited data learning approaches, could help better resolve unseen classes during training and testing by learning features that are suited to the variations commonly seen in the target signal classes. This may enable scenarios where a system or user can accept or reject a class assignment based on similarity or outlier metrics. The success of the Siamese approach in this case suggests that these neural networks might be effective substitutes for conventional CNNs in other machine learning-based signal processing.

REFERENCES

[1] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, Berlin, Heidelberg.

[2] M. Bkassiny, Y. Li, and S. K. Jayaweera. 2013. A Survey on Machine-Learning Techniques in Cognitive Radios. *IEEE Communications Surveys Tutorials* 15, 3 (Third 2013), 1136–1159. https://doi.org/10.1109/SURV.2012.100412.00017

[3] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* 106 (2018), 249 – 259. https://doi.org/10.1016/j.neunet.2018.07.011

[4] S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 539–546 vol. 1. https://doi.org/10.1109/CVPR.2005.202

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[6] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*. http://arxiv.org/abs/1412.6572

[7] Z. Han, R. Zheng, and H. V. Poor. 2011. Repeated Auctions with Bayesian Nonparametric Learning for Spectrum Access in Cognitive Radio Networks. *IEEE Transactions on Wireless Communications* 10, 3 (March 2011), 890–900. https://doi.org/10.1109/TWC.2011.010411.100838

[8] L. Hertel, J Collado, P Sadowski, and P Baldi. 2018. Sherpa: Hyperparameter Optimization for Machine Learning Models. (Oct. 2018). https://openreview.net/pdf?id=S1lX0KaE3m

[9] J. Jagannath, H. M. Saarinen, and A. L. Drozd. 2015. Framework for automatic signal classification techniques (FACT) for software defined radios. In *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*. 1–7. https://doi.org/10.1109/CISDA.2015.7208628

[10] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese Neural Networks for One-shot Image Recognition. In *International Conference on Machine Learning*.

[11] X. Liu, Y. Zhou, J. Zhao, R. Yao, B. Liu, and Y. Zheng. 2019. Siamese Convolutional Neural Networks for Remote Sensing Scene Classification. *IEEE Geoscience and Remote Sensing Letters* (2019), 1–5. https://doi.org/10.1109/LGRS.2019.2894399

[12] T. O'Shea, T. Roy, and T. C. Clancy. 2017. Learning robust general radio signal detection using computer vision methods. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*. 829–832. https://doi.org/10.1109/ACSSC.2017.8335463

[13] T. J. O'Shea, T. Roy, and T. C. Clancy. 2018. Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE Journal of Selected Topics in Signal Processing* 12, 1 (Feb 2018), 168–179. https://doi.org/10.1109/JSTSP.2018.2797022

[14] I. Santamaria, C. Pantaleon, L. Vielva, and J. Ibanez. 2004. Blind equalization of constant modulus signals using support vector machines. *IEEE Transactions on Signal Processing* 52, 6 (June 2004), 1773–1782. https://doi.org/10.1109/TSP.2004.827176

[15] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4077–4087.

[16] Y. Tang, Q. Zhang, and W. Lin. 2010. Artificial Neural Network Based Spectrum Sensing Method for Cognitive Radio. In *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*. 1–4. https://doi.org/10.1109/WICOM.2010.5601105

[17] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

[18] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 3630–3638.

[19] Meng Ye and Yuhong Guo. 2018. Deep Triplet Ranking Networks for One-Shot Recognition. *CoRR* abs/1804.07275 (2018). arXiv:1804.07275 http://arxiv.org/abs/1804.07275