Convolutional Neural Networks for Radar Emitter Classification

Lindsay Cain*, Jeffrey Clark*, Eric Pauls†, Ben Ausdenmoore*, Richard Clouse, Jr.†, Ted Josue*
*Riverside Research
{lcain, jclark, bausdenmoore, tjosue}@riversideresearch.org
†Harris Corporation
{epauls, rclouse}@harris.com

***Abstract*—In this paper, an application of convolutional neural networks (CNN) for rapid and accurate classification of electronic warfare emitters is investigated; a large data set with 58 separate emitter sources is used for training and testing. Data preprocessing creates 3-dimensional images with a feature space composed of pulse width (PW), radio frequency (RF), and pulse repetition interval (PRI), referenced with respect to time of arrival (TOA). The image representation has proven to be the most effective, consistently producing classification accuracies approaching 98.7%. This study, which evaluates emitter-by-emitter classification, appears to be a novel approach, based on a survey of current literature; previous work citing the use of CNNs in this domain has been limited to radar waveform recognition vice pulse-based specific emitter identification.**

***Index Terms*—Convolutional Neural Network, classification, radar, emitter**

I. Introduction

Within a complex and dynamic military environment, the ability to control and dominate the electromagnetic spectrum is key to achieving control in combat operations. This task is challenging in the most benign situation and is compounded when adversaries possess the capability to deploy adaptive electronic warfare (EW) capabilities. Electronic warfare involves the manipulation and control of the electromagnetic spectrum. This includes using the electromagnetic spectrum to attack or prevent attacks [1]. With the onset of software defined threats, the timeline associated with combating EW threats has been reduced significantly. The current procedures involves collecting the signals, sending the data back for analysis, and determining the best countermeasure for the threat. This process is no longer sustainable in the current operational environment and therefore more intelligent and responsive techniques must be employed. The ability to appropriately detect, identify, and respond to signals is required to maintain superiority in conflicts.

A fast and highly adaptable architecture is required to respond to threats in real-time. In this paper, deep learning techniques are explored to adaptively identify the signals present in a complex EW environment. Specifically, a convolutional neural network (CNN) is employed to automatically classify complex and ambiguous emitter signals from 58 separate emitters. This technique involves preprocessing of the emitter data to create images of the data for input into the CNN. The network is then trained and tested. This process has yielded promising results with regard to accurate signal classification. The implications of such technology would allow for the ability to identify unknown signals and automatically adapt to an ever-changing EW environment in near real-time.

A. Related Work

Convolutional neural networks (CNNs) are exceptionally prolific in the computer vision community. These networks, specifically designed for images, are the driver behind the significant progress in the field of computer vision for object recognition tasks. CNNs have also been applied to less traditional tasks such as speech and hand writing recognition [2] [3]. Recently, CNNs have been employed for spectrum monitoring and radar waveform recognition [4] [5] [6]. Specifically in [4], the authors sought to detect the presence of radar signals even with the simultaneous presence of Long-Term Evolution (LTE) and Wireless Local Area Network (WLAN) signals in the environment. The signals were represented with amplitude and phase spectrograms that resulted in 99.6% classification accuracy. Furthermore in [5], the authors used a CNN in combination with an Elman neural network (ENN) for waveform recognition. The authors used 22 different features to include the Choi-Williams time-frequency distribution (CWD). With the combination of the CNN and ENN, the technique was able to identify 12 kinds of signals, including binary phase shift keying (BPSK), linear frequency modulations (LFM), Costas codes, Frank code, P1-P4 codesand T1-T4 codes with a low signal-to-noise ratio (SNR). The results indicated 94% accuracy with a SNR $\geq -2dB$. Finally in [6], the authors use time-frequency images, specifically the Wigner-Vile Distribution for input into a CNN with high accuracy at SNR $\geq -2dB$ for 7 types of radar waveforms.

B. Contributions

While the related studies have yielded promising results these techniques only identify the waveform type, which is only one piece of the emitter classification problem. This study seeks to identify on an emitter-by-emitter basis to determine which signal belongs to each individual emitter (in this case 58 separate emitters) with 98.7% accuracy. The contributions of this work includes the following four items:

978-1-5386-4649-6/18/$31.00

1) A novel and simple way to create images for each emitter for input into the classifier.
2) A study of the optimal feature set for ambiguous and complex emitter population.
3) A relatively small and efficient feature set for rapid processing of the signals.
4) A CNN framework that can perform classification of 58 individual ambiguous emitters with high accuracy.

II. Method

The classification system consists of three main components: preprocessing, feature generation, and classification. The entire process is automatic and can be deployed to autonomously classify the signals within an environment. In the preprocessing step, the data is received as a pulse descriptor word (PDW). The PDW contains the native features acquired by the receiver, and determined by the de-interleaver hardware and software. This includes the time of arrival (TOA), amplitude, and frequency. Each PDW represents one pulse of the emitter and is arranged together by TOA to create an entire signal. An image is created by plotting the feature versus TOA for a given time sample and windowed across the entire signal. Feature generation is accomplished by transforming the native and derived features to logarithmic space, creating a more discriminative representation of the data for input into the CNN. These images are normalized by the global minimum and maximum and are the inputs of a CNN for training. The following sections include a detailed description of the preprocessing, feature generation, and network structure that describe the overall classification framework.

A. Data Preprocessing

The data is received in the form of a PDW where each pulse has a variety of native features. The PDWs are parsed and read into a matrix for ease of data manipulation. After the data is properly parsed and stored in a matrix, the global minimum and maximum of each feature is saved for normalization. The next step is to create a multidimensional image; this image is a plot of the TOA and the feature magnitude along the x and y axes, respectively. Multiple images of each emitter are created by dividing up TOA based on a sample size and step size. Here, sample size dictates the range of the x-axis of the image and step size determines how far the sample window is shifted to create the next image. As an example, Fig. 1 is a depiction of the definition of sample size and step size as the images are created for each emitter.

An image is created for each feature of each sample and are concatenated to create a multidimensional image. The y-axis of each image is normalized by the global minimum and maximum of each feature. The images are then resized to 100 x 100 images for input into the CNN. Fig. 2 is an example of one image set that includes PW, RF, and PRI as a feature set with a sample size of $40ms$ and a step size of $20ms$.

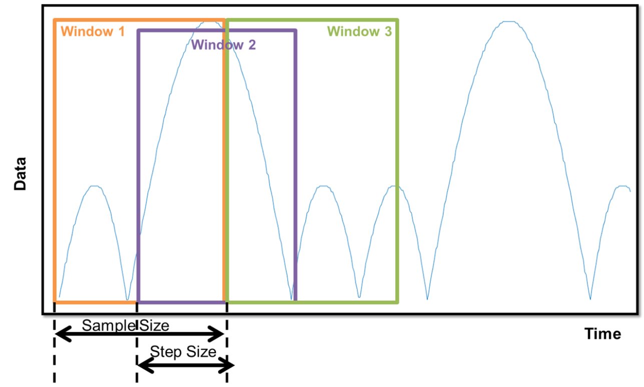


Fig. 1: Depiction of sample size and step size across the data to create images for input into the CNN.

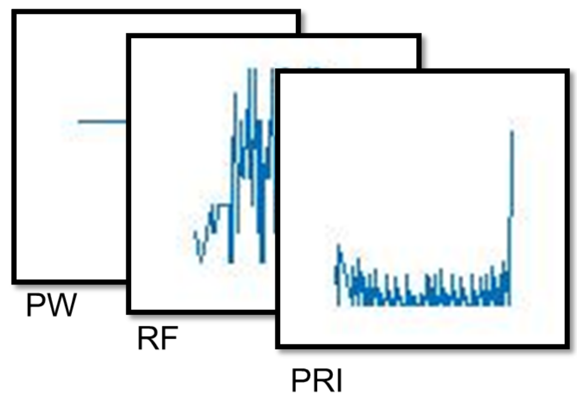


Fig. 2: Sample image set containing the features PW, RF, and PRI, with a sample size of 40ms and a step size of 20ms, image size 100 x 100 x 3.

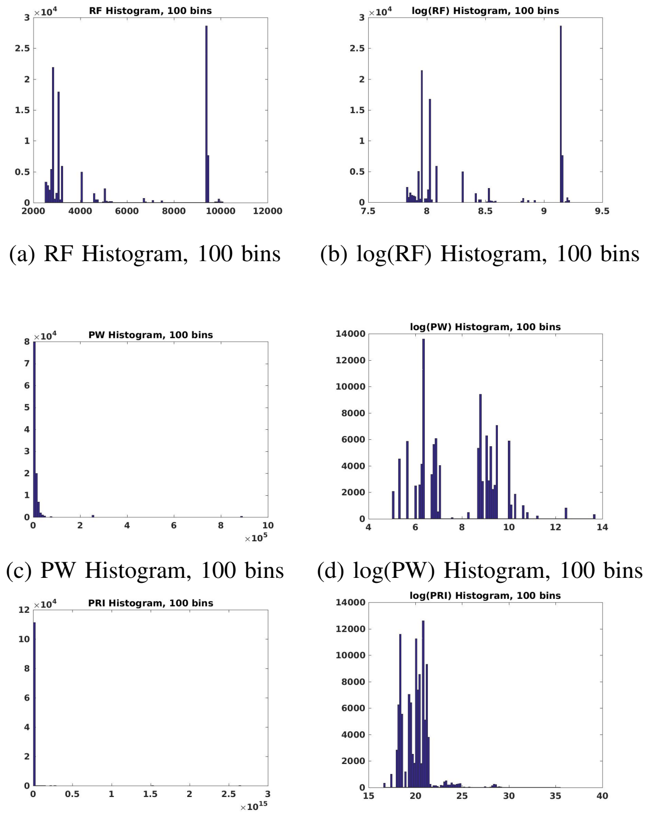
The images are created and randomly assigned to the training and validation set. The images are normalized between $[0, 1]$ by using the following equation:

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)}, \tag{1}$$

where z_i is a normalized sample, x_i is the sample before normalization, and x represents the entire data set. Finally, the mean is subtracted from each image to zero mean the input images.

B. Feature Generation

An extensive investigation was completed to determine the optimal feature set. The native features investigated include radio frequency (RF), pulse width (PW), and pulse repetition interval (PRI). The PRI is calculated from the derivative of the TOA. Due to the distribution of the data and image resolution, the majority of data was initally resolved to the first row of pixels. Therefore, the logarithm transform was applied to each feature to ensure that the data was distributed throughout the image more effectively.



(a) RF Histogram, 100 bins (b) log(RF) Histogram, 100 bins

(c) PW Histogram, 100 bins (d) log(PW) Histogram, 100 bins

(e) PRI Histogram, 100 bins (f) log(PRI) Histogram, 100 bins

Fig. 3: Comparison of the histograms between the native features and the logarithm of the native features. Each histogram is represented with 100 bins, therefore each bin represents a row of pixels in the 100 x 100 images.

Fig. 3 compares the histogram of PW, RF, and PRI to the logarithm of each feature respectively. Each bin of the histogram represents one row of pixels in the image, therefore there are 100 bins for a 100 x 100 image. Fig. 3a, 3c, 3e is the histogram of each native feature and Fig. 3b, 3d, 3f is the result after the log is applied to the data. Notice that the data is shifted out of the first row of pixels for PW and PRI, whereas RF remains relatively unchanged. Applying the log transform to the data ensures that more of the data is represented across the entire image which will aid in discrimination and classification of the emitters.

In addition to the three feature logarithm set, a larger feature set (13 features) was investigated. This feature set was a combination of statistical moments derived from the native features. This set yielded promising results; however, due to shorter computational time, the three feature set was preferable to the 13 feature set. Therefore, the final feature set for this investigation is log(RF), log(PW), and log(PRI).

C. Network Structure

The data is represented as images that are 100 x 100 x 3, where 3 is the number of features. However, the CNN network could be adapted to include more features by changing the depth of the input images and first set of filters. Fig. 4 depicts the proposed CNN architecture, inspired from the LeNet-5 architecture [7]. This relatively simple architecture was chosen as a starting point and has proven to be robust for the emitter dataset. The images are sent through five convolutional and pooling layers followed by one fully connected layer. The softmax loss is used for training and a softmax classifier is used to compute the probability $P(y = k|x; \theta)$ for $k \in \{0, 1\}$, where x denotes the input and θ denotes the model parameters [4]. The training is performed using stochastic gradient descent, where the weights of the network are updated after each iteration with the goal of minimizing classification error.

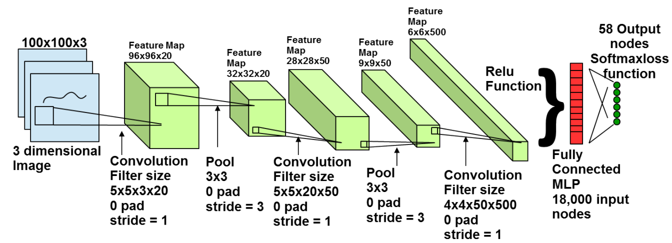


Fig. 4: Example CNN structure, the input image depth will change based upon number of features, changing the size of the feature maps.

III. Experimental Results

A. Data

The PDW data for this experiment is generated using a radar scenario simulator, which creates a stream of pulse data mimicking the PDW buffer of an observing radar receiver. There are 58 independent emitters in the scenario which were selected to provide a diverse and sufficiently challenging data set for classification. The emitters vary in PW, PRI, and RF. Emitters in the scenario have either fixed or agile parameters. With fixed parameters, each pulse from the radar system will occur with the same width, frequency, and repetition rate. In the aforementioned image space, a train of pulses from a fixed emitter would occupy a single row of pixels. An agile emitter allows one or more of its pulse parameters to change over time. In this scenario, about 80% of the emitters are agile in RF, PRI, or both. Of the agile set of emitters, some exhibit pulse-to-pulse agility, meaning the signal parameters can change with each pulse. Other emitters exhibit group-to-group agility, where a set number of pulses are transmitted with the same parameters before switching to a new behavior. The transition between pulse parameters can either be random or determined by a predefined pattern.

As an example, Fig. 5 displays the types of agility possible in this data set. This emitter is group-to-group agile in PRI. It emits a group of four pulses before changing to a new pattern. The first group (red, left oval) has a short PRI and the second group (blue, right oval) switches to a longer PRI. In RF, the emitter is pulse-to-pulse agile with a random frequency

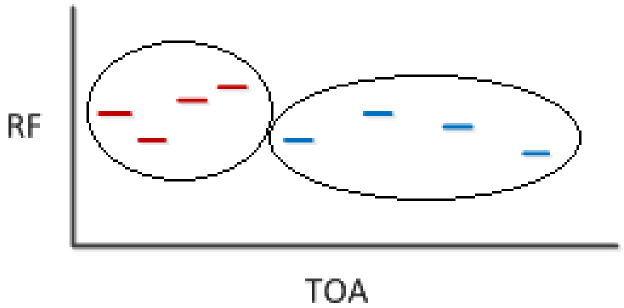
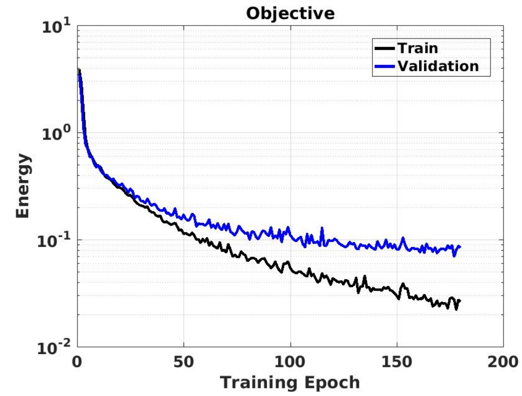


Fig. 5: Example of the types of agility within the emitter dataset. The first pulse group (red, left oval) is an example of a short PRI that switches to the second group (blue, right oval) representing a longer PRI. The groups are circled for grayscale consideration.

TABLE I: List of parameters used for training and testing the CNN.

Parameter	Value
Train/Test Percentage	70% / 30%
Number of Samples	7096
Batch Size	100
Learn rate	0.001, constant
Weight decay	0.0005
Momentum	0.9
Normalization	Min/Max & Zero Mean



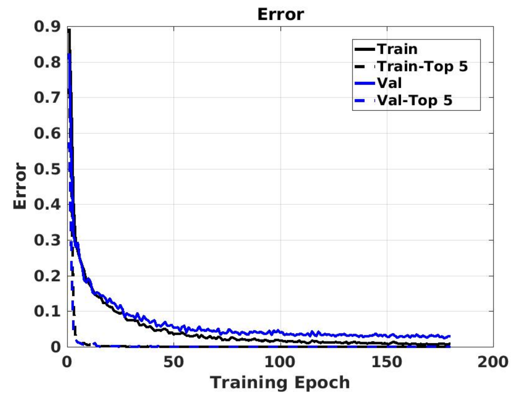


Fig. 6: Objective and error plot for the 3-feature set (log(RF), log(PW), log(PRI)), includes the Top 1 and Top 5 training and validation error.

selection. The emitter parameters are chosen such that there are significant ambiguities in the fixed values, as well as in the agile behaviors.

All of the data used for this evaluation is represented by creating images, as discussed in Section II-A. The image sets are 100 x 100 x 3 and are created using a sample size of $40ms$ and a step size of $20ms$.

B. Results and Discussion

An extensive parametric test was completed to determine the chosen feature set. For simplicity, the result of the three feature logarithm set will be presented and discussed. The CNN model is trained on 70% of the data and validated with 30% of the data with a batch size of 100. A summary of the network parameters are summarized in Table I. The classification framework was developed and implemented with MatConvNet [8] using MATLAB 2016a [9]. The results of the three feature logarithm set are shown in Fig. 6, where 'Objective' is the magnitude of the softmax loss function during training and validation and 'Error' indicates the training and validation error relative to the truth label. The 'top 1' error is the chance that the class with the highest probability is the true correct target. The 'top 5' error is the chance that the true target is one of the five top probabilities.

The results provide a first look into the ability for a CNN to accomplish emitter-by-emitter classification. This technique yielded an equal weighted classification accuracy amongst all 58 classes of 98.7% for the testing dataset. This study does not compare with previous tests because there is currently no work that is comparable.

IV. Conclusion

This paper presents a novel and efficient method to classify EW emitter data from 58 separate emitters. An extensive study was completed to properly represent the data as an image to the CNN. The data is represented as images where TOA is plotted versus the logarthim transform of the PW, RF, and PRI. This simple data representation proved to be the most efficient while also maintaining adequate performance. The results indicate that this method is viable for further development with an impressive 98.7% classification accuracy.

The techniques described are efficient and, once trained, could be deployed for real-time emitter identification. To improve the classification accuracy, further investigation will be required to determine how to present the emitter data to the CNN. There are a variety of techniques that can be employed to better represent the true fingerprint of the system. As an example, highlighting the agility of a signal would capitalize on the CNN's ability to classify complex signals over current methods. In the future, a hybrid architecture can be employed that combines the current CNN with an basic artificial neural network (ANN) that receives the raw RF, PW, and PRI features to further improve classification accuracy. Additionally, further investigation will be required to determine the technique's ability to withstand the presence of noise and signal dropout. Finally, creating a classifier that can handle unknown signals will add functionality to the system that is necessary for a complex and adaptive EW environment.

References

[1] D. L. Adamy, *EW 104: Electronic Warfare Against a New Generation of Threats*. Artech House, 2015.

[2] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[3] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[4] A. Selim, F. Paisana, J. A. Arokkiam, Y. Zhang, L. Doyle, and L. A. DaSilva, "Spectrum monitoring for radar bands using deep convolutional neural networks," *CoRR*, vol. abs/1705.00462, 2017. [Online]. Available: http://arxiv.org/abs/1705.00462

[5] M. Zhang, M. Diao, L. Gao, and L. Liu, "Neural networks for radar waveform recognition," *Symmetry*, vol. 9, no. 5, 2017.

[6] C. Wang, J. Wang, and X. Zhang, "Automatic radar waveform recognition based on time-frequency analysis and convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 2437–2441.

[7] Y. LeCun *et al.*, "Lenet-5, convolutional neural networks," 2015. [Online]. Available: http://yann.lecun.com/exdb/lenet/

[8] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[9] MATLAB, *version 9.0.0.341360 (R2016a)*. Natick, Massachusetts: The MathWorks Inc., 2016.