# EE492
# Senior Design Project Final Report

# DEEP LEARNING APPROACH ON SPECIFIC EMITTER IDENTIFICATION USING UMOP CHARACTERISTICS

**by**
Fehmi Ayberk Uçkun

A report submitted for EE491 (or EE492) senior design project class
in partial fulfilment of the requirements for the degree of
Bachelor of Science
(Department of Electrical and Electronics Engineering)
in Boğaziçi University

February 11th, 2021

**Principal Investigator:** Emin Anarım

# ACKNOWLEDGMENTS

I would like to thank Prof. Dr. Emin ANARIM for his support and understanding throughout this project. I am grateful to Dr. Emrah ONAT for his encouragement and support.

# ABSTRACT

In this paper, a deep learning approach is proposed in order to solve the specific emitter identification (SEI) problem. SEI is a technique which distinguishes emitters, including same class emitters, using nonlinearity features known as unintended modulation on the pulse (UMOP) which stem from the nonlinearity of the emitter hardware. 3 different UMOP feature extraction methods, Instantaneous Amplitude, Hilbert-Huang Transform (HHT) and Bispectrum are used and instead of using expert analysis, convolutional neural networks (CNNs) are utilized in order to analyze and classify the processed signal information. Results suggest that image data as a UMOP feature can be too complicated for small networks to learn and current widespread methods such as dimensionality reduction or handcrafting other features like entropy supports this idea.

# TABLE OF CONTENTS

**CHAPTERS**

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

In electronic warfare, knowledge of enemies' electronic capabilities is desired. This includes the interception and analysis of enemy radar signals. Thus, emitter identification is an important issue in electronic intelligence (ELINT) and electronic support measure (ESM) systems. The conventional approach for emitter identification was using "Pulse Descriptor Word" (PDW) parameters such as radio frequency (RF), pulse width (PW), and pulse repetition interval (PRI). However, in modern radar systems, the environment is drastically changed with the increasing number of emitter usage, especially of the same classes, and with the usage of more sophisticated waveforms. This made it difficult to identify different emitters with conventional ways. Then specific emitter identification (SEI) technique is proposed. SEI is a technique that distinguishes different emitters using radiofrequency fingerprints. The RF fingerprint, also known as the "Unintentional Modulation on Pulse" (UMOP), is mainly generated by the nonideal and nonlinear devices in the emitter, which will bring unwanted modulation to the transmitted signals. Also, it is proved that UMOP has characteristics of universality, uniqueness and stability. Since UMOP is unavoidable and unique to individual emitters, the developing SEI technique offers attractive prospects for emitter identification [1].

For SEI, great many UMOP feature extraction techniques have been proposed. All these approaches are trying to find the best compromise between the computational complexity and the feature discriminativeness. They can be mainly divided into time, frequency and transform-domain features. Basic time-domain features include the instantaneous amplitude/frequency/phase. These features are easy to obtain, but at the same time, they are susceptible to noise. Transform-domain features are generally calculated from complicated transformations such as Hilbert–Huang transform, ambiguity function, fractal, and cumulants. Although robust to noise, these features usually face the problems of heavy computational loads and/or high dimensions. Frequency-domain features including Fourier spectrum and power spectral density can offer a compromise, but they require precise frequency alignment [2].

Among them, one of the most considerable approaches is higher-order spectral analysis (HOSA). HOSA was developed in the late 1980s, research has shown its superiority in analyzing non-Gaussian and nonstationary processes, such as modelling of nonlinear systems and signals classification. The higher-order spectrum refers to the multidimensional Fourier transform of higher-order cumulants. Compared with feature extraction methods in the time and frequency domains, the higher-order spectrum has three characteristics: it contains amplitude and phase information simultaneously, it includes the nonlinear component of signals, and the higher-order cumulant of Gaussian noise equals to zero. Owing to these characteristics, it is possible to extract UMOP features using the higher-order spectrum [3].

However, even the least computationally expensive and discriminative approach of bispectrum still has more dimensions than an ideal solution thus several dimensionality reduction techniques like Radon Transform and Surrounding-Line Integral and many more are proposed. After a successful and acceptable UMOP characteristic feature extraction step, classification is realized. The most common approach for classification stayed as the nearest-neighbour algorithm while many improved clustering methods are continuing to develop.

## 1.1 Works on Literature

Many methods are proposed for SEI and discussions are made based on usefulness and universality of the proposed UMOP features. Below, I am listing the information extracted from the literature in order to give necessary background knowledge to the reader and as a guide to fellow researchers who are inclined to do research on this topic.

### 1.1.1 Transient/Steady State Signal

In [4], the authors used the transient part of the signal for feature extraction. They believed transient signals usually have special RFF for SEI because when emitter turned on or switched modes, the signal goes through a transient state which is caused by a combination of effects, such as the characteristics of subsystems and amplifiers which also causes the UMOP in the first place. Although, this approach is logical, detecting transient signals accurately for its shot duration is difficult especially in noncooperative communications [23] and transient features are easily hampered by non-ideal and complicated channel conditions [21]. Moreover, the transient signal shows bad performance for the identification of the transmitters which have the same model and manufacturing lot [24]. However, in contrast to the transient signals, the use of steady-state signals provides more statistically stable RFF features [23] and for transmitters with the same model and same batch, the individual difference with the steady transmitted signal will grow out of uncertainty during the device manufacturing process [24]. Indeed, most of the works on SEI use steady-state signals [1-3] [5-24].

### 1.1.2 Single Hop/Relaying Scenarios

While in literature, works concentrates on single-hop communication scenarios, in [21], the authors tested the proposed method for both single-hop and relaying scenarios. Relays are widely deployed in communication systems. For instance, in satellite communications, the features carried by the received signal are the mixture of the fingerprints of the emitter and the relay. Their work, together with [12] shows that this has a negative effect on SEI.

### 1.1.3 AWGN/Rayleigh/Multipath Channel

While most of the works are using AWGN channel, some works [12] tested other channels also. Their work shows that the performances under the multipath channel approximate those under the AWGN channel, even better than those under the Rayleigh fading channel. This implies that inter-symbol interference introduced by the multipath channel has a similar effect on the distribution in the spectrum images of different emitters; whereas the fading coefficient α of the Rayleigh channel leads to more randomness, which blurs the difference between emitters. Even with this generalization both [12] and [14] claim that their proposed method shows the least decrease in Rayleigh channel.

### 1.1.4 Machine Learning Utilization

Machine learning and deep learning usage vary in the literature. Most of the old works rely on statistical feature extraction and clustering. Some of these works applied clustering using machine learning and some used SVMs as classifiers such as [21]. In the last years, a couple of end-to-end deep learning applications is started to develop [3], [11-12], [16], [22-23]. As a different approach, in [10], authors used CNNs as a feature extractor and then classified the emitters using a clustering algorithm and in [3], authors extracted features statistically and classified with CNNs.

### 1.1.5 Simulated/Real Data

Many works in literature shows the performance of the proposed methods using either real-world data [2-9], [18], [20], [23] or simulated data [10-14], [21-22], [24] but [5], [14] and [16] uses both data types. Successful results on both data types show that real-world environment conditions can be simulated successfully in computer. However, some of the works [2], [4-5], [7], [9], [21] uses limited numbers of emitter examples (less than 5) to measure to performance. This makes the results unreliable. Fortunately, some of the works [1], [3], [6], [10], [16], [23] test the methods with adequate number of emitters. Considering the consistency and uniqueness of UMOP [1], we can trust the results of the other works also. Another problem with the works is that not all the works address the challenge of the same type, same model emitter identification while testing. Therefore, all purposed methods may not be suitable for this problem. In [1], [5-7], [9], [23] authors address this problem even with the real data collection.

### 1.1.6 Extracted Features

The most important difference in the literature is the UMOP feature extraction method. Each claiming their method is superior to others. We already mentioned these features mainly as time, frequency, transform-domain and HOSA features.

#### 1.1.6.1 Time Domain Features

[10] uses only the raw I/Q data to create features with CNNs and claims that by using only raw I/Q as input to the CNN, the proposed approach eliminates the need for pre-processing steps such as synchronization or demodulation typically required by

traditional approaches and their approach does not assume a priori knowledge of any emitters thus suitable to non-cooperative scenarios. Their work shows that their approach performs poorly when incoming signals change their bandwidths and classify the same emitter with different bandwidth signals as different emitters. They also used the same modulation scheme for every data.

[11] presents an approach for identifying emitters using convolutional neural networks to estimate the in-phase/quadrature (IQ) imbalance parameters of each emitter, using only the received raw IQ data as input. Because an emitter's IQ imbalance parameters will not change as it changes modulation schemes, the proposed approach has the ability to track emitters, even as they change the modulation scheme.

#### 1.1.6.2 Frequency Domain Features

[1] uses Instantaneous Amplitude and Frequency (IA/F) as features. [2] claims that the amplitude of spectrum has bilateral symmetry and evaluates the spectrum asymmetry caused by UMOP. They say other feature extraction methods face the problem of the overwhelming influence of intended modulations. [18] uses IA as a base and extract features such as rise time, fall time.

#### 1.1.6.3 Transform Domain Features

[4] claims that as communication security become increasingly important and frequency hopping technique is widely used, an SEI method for frequency hopping signals is meaningful. They use normalized frequency spectrum features extracted from the transient signal using Box-counting fractal dimension. They claim that as spectrum resources become sparse, the frequency hopping technique is widely used, and the frequency domain is more robust to white noise than the time domain. [5] utilizes weak individual characteristics caused by phase noise and applies fractal theory to the feature extraction. They claim that the nonlinear dynamics features such as fractal dimension features, iterated function system and permutation entropy reflect the nonlinear dynamics characteristics of RF power amplifiers and oscillators.

[12] constructs a deep residual network for learning the visual differences reflected in the Hilbert spectrum images. [13] also uses HHT with empirical mode decomposition (EMD). [14] uses HHT but claims that Intrinsic Time-scale Decomposition (ITD) is more suitable for non-stationary signals than EMD. [21] uses HHT as well.

#### 1.1.6.4 Higher-Order Spectral Domain Features

[3] uses Bispectrum and Radon Transform for feature extraction. They claim that higher-order spectrum is more useful than other features because of 3 reasons; it contains amplitude and phase information simultaneously, it includes the nonlinear component of signals, and the higher-order cumulant of Gaussian noise equals to zero. Also, they claim that waveform approaches (such as IA/F) have the advantage of lower

computational costs, but they are easily affected by environmental noise. They also perform poorly when analyzing non-Gaussian and nonstationary processes. For the transform domain features (such as HHT, Wavelet), they have computational complexity with high dimensions. Moreover, kernel and basis function choices affect success considerably.

[7] uses high order cumulant matrices for feature extraction. [8] uses bispectrum with surrounding-line integral for dimensionality reduction. Claims that other integral approaches perform worse. [23] uses bispectrum with dimensionality reduction and lastly [24] again uses bispectrum with local integral. [9] uses linear skewness and kurtosis to extract the non-linearity of the signal caused by UMOP. [16] uses gray histograms of bispectrum and further incorporates a priori statistical characteristics of the wireless channel via GAN.

## 2. METHODOLOGY

In this paper, 3 most prominent feature extraction methods from each domain, excluding time domain, are used in order to test the success of CNNs on each domain features.

### 2.1 Instantaneous Amplitude (IA)

This feature is calculated by taking the upper envelope of the signal pulse. Several features from the resulting signal are used as Amplitude UMOP (A-UMOP) features in the literature. These features are [18]:

**i) Rise time duration:** It refers to the power transition from the noise floor to the maximum amplitude.

**ii) Transitory time:** It is the interval after the rise time frame where the amplitude of the signal is not yet constant around the nominal amplitude value.

**iii) Constant amplitude time:** Where the amplitude of the signal is almost constant around the nominal value.

**iv) Fall down time:** It refers to the power transition from the nominal amplitude to the noise floor level.

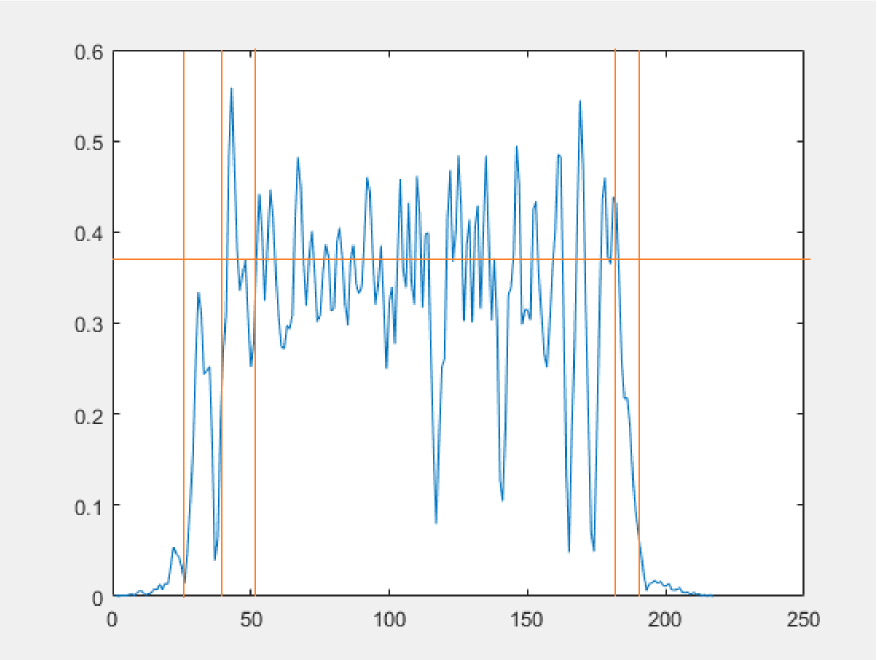

***Figure 1.*** *Upper envelope of a pulse with approximate feature timings*

We used this image as an input to the CNN algorithm instead of handcrafting such features.

## 2.2 Hilbert-Huang Transform (HHT)

HHT is one of the transform domain features which is used in order to analyze nonlinear and non-stationary signals [12]. It is more adaptive compared to traditional transforms such as Fourier [21]. HHT constitutes of 2 steps: empirical mode decomposition (EMD) and Hilbert spectrum analysis.

### 2.2.1 Empirical Mode Decomposition (EMD)

The purpose of EMD is decomposing a signal to the finite number of intrinsic mode functions (IMFs) to represent it with physically meaningful frequency components. To define an IMF following assumptions must be true [21]:

**i)** The number of extrema either equals the number of zero-crossings, or the difference is one at most.

**ii)** At any point, the sum of the upper and lower envelopes is zero.

EMD algorithm involves an iterative sifting process which is applied by the following steps [12]:

**i)** Finding local minima and maxima of x(t).

**ii)** Constructing upper and lower envelopes and finding the mean of envelopes m(t).

**iii)** Subtracting the mean m(t) from signal x(t) and obtaining the residual r(t).

**iv)** Repeat the process until the stopping criteria is satisfied or until r(t) includes no oscillations. r(t) becomes the x(t) of the second loop (first IMF component at the same time).

The process can be summarized as

$$x_{i+1}(t) = x_i(t) - m_i(t) \tag{1}$$

$$s(t) = \sum_{i=1}^{N} x_i(t) + r(t) \tag{2}$$

where s(t) is the signal, x(t) is IMF, N is the number of IMFs that satisfies the criteria and r(t) is the last residual.

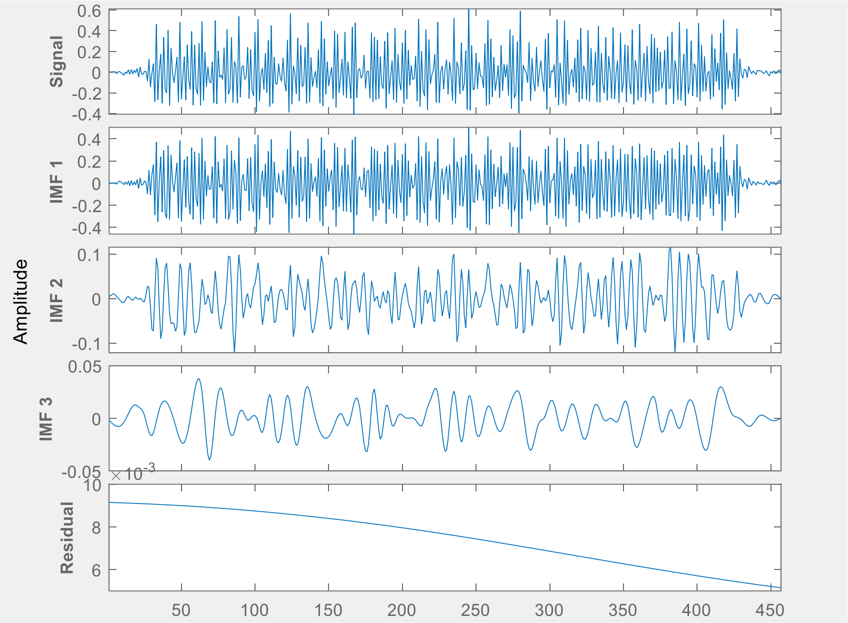

***Figure 2.*** *Empirical Mode Decomposition, showing 3 out of 7 IMFs*

### 2.2.2 Hilbert Spectrum Analysis

The last step uses Hilbert Transform in order to from

$$z_i(t) = x_i(t) + jH\{x_i(t)\} \tag{3}$$

and

$$r(t) = Re\left\{\sum_{i=1}^{N} a_i(t)e^{j\int w_i(t)dt}\right\} \tag{4}$$

Where $z_i(t) = a_i(t)e^{j\theta_i(t)}$, $a_i(t)$ is the instantaneous amplitude, $\theta_i(t)$ is the instantaneous phase and $\omega_i(t) \equiv d\theta_i(t)/dt$. Resulting frequency-time distribution plot called as Hilbert Spectrum H(w, t).

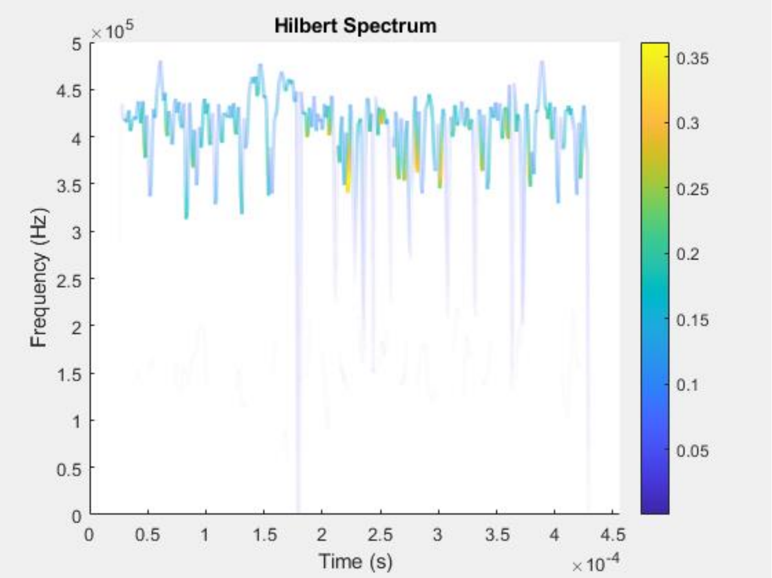

***Figure 3.*** *Hilbert Spectrum*

We gave the below grayscale converted 8-bit Hilbert spectrum image as the input to the CNN algorithm instead of handcrafting features such as correlation and entropy.

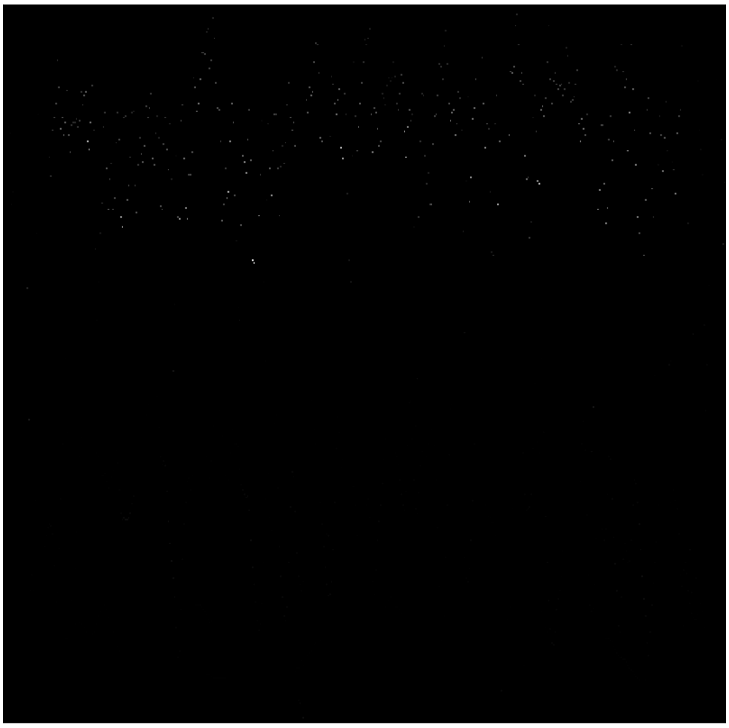

***Figure 4.*** *Grayscale Hilbert spectrum image used as input to CNN*

## 2.3 Bispectrum

Bispectrum is calculated as the two-dimensional Fourier transform of the third-order cumulants in HOSA theory. It is an important tool for the detection and characterization of non-Gaussian signals. It has the good properties of keeping the phase information with the amplitude and suppressing the noise [8]. It has the lowest order and the fewest calculations among other higher-order spectrums [3].

Bispectrum defined as

$$B_X(k_1, k_2) = \frac{1}{N} X(k_1) X(k_2) X^*(k_1 + k_2) \qquad (5)$$

Where X(k) is the Discrete Fourier Transform (DFT) of x(n) and N is the period.

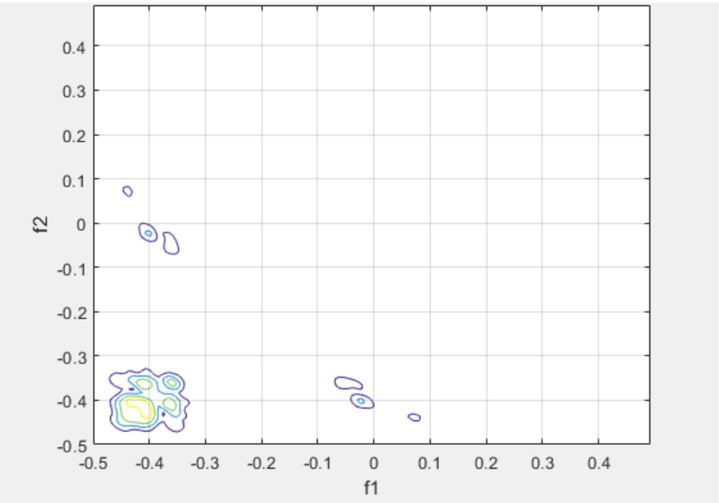

***Figure 5.*** *Bispectrum estimated via the direct FFT method*

We used this image as an input to the CNN algorithm instead of applying dimensionality reduction transforms/integrals in order to handcraft features.

## 2.4 Convolutional Neural Networks (CNNs)

Realizing SEI has two key problems: feature extraction and classification. Both of which can be solved via deep learning techniques. With the increasing computational power of computers, deep learning has become one of the most prominent topics of our decade. Usage of Neural Networks (NN), a human brain neural cells-like processing units, showed increasingly successful results in many areas including electronic intelligence and started to replace long-proposed mathematical solutions to problems.

CNNs are deep learning algorithms which are specially constructed for data which have grid-like topology such as images. Images are 3-dimensional matrices with RGB pixel values. Development of CNNs was necessary due to the struggle of traditional networks against this type of data. These problems [25]:

**i)** Giving the matrix into a fully connected network by flattening, results in a huge number of parameters to train.

**ii)** Flattening results in a loss of spatial information between data.

CNNs constitutes of mainly 2 layers:

**i)** Convolutional Layer

**ii)** Pooling Layer

At the convolutional layer, kernel filters are applied to the matrix. These filters are multi-dimensional like the input data (like 5x5x3). Purpose of this layer is to find local features of the matrix. Pooling layer is applied in order to reduce the number of parameters in the network. The assumption made is this process reduces highly correlated neighbouring values and gives the ability to learn more general features of the matrix.

# 3. SYSTEM MODEL

This paper is adopted the Taylor polynomial signal model as it is widely used in the literature [12], [14], [16], [21] to represent unique nonlinear characteristics of each emitter. The used discrete-time signal model can be represented as

$$r_k(n) = \sum_{l=1}^{L} \propto_l \, s_k(n)^l + n_k \tag{6}$$

Where $r_k(n)$ is the output signal of the emitter, L is the order of the polynomial, $\propto_l$ are the coefficients of the Taylor polynomial for that specific emitter, $n_k$ is the additive white gaussian noise and $s_k(n)$ is the modulated signal formulated as

$$s_k(n) = x_k e^{j2\pi n\frac{f_c}{f_s}} \tag{7}$$

Where $x_k$ is the modulated baseband signal and $e^{j2\pi n\frac{f_c}{f_s}}$ is the carrier.

## 3.1 Data Generation

For the simulations, data for 5 different emitters are generated using the third-order Taylor polynomial signal model and the coefficients are given in Table 1. All data is created using MATLABR2020b. Several frequency values and generation techniques are tested. Sampling frequency values $F_s = [5.10^9, 10^6]$ and carrier frequency values $F_c = [2000.10^6, 420.10^3]$ values are used respectively. QPSK Modulation used together with a raised cosine filter which has the roll-of factor of 0.35 and symbol span 8. Each symbol is represented with 8 points and each segment constitutes 457 symbols. To train the algorithm two different datasets are created. The first dataset is created without noise and each emitter included 2000 segments. The second created using Gaussian noise at SNR values [10:2:20] and for each SNR, emitters included 500 segments. 40% of the data used for validation and 60% per cent used for training.

| ***Coefficients***<br>***Emitters*** | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ |
|---|---|---|---|
| $E_1$ | *1* | *0.5* | *0.3* |
| $E_2$ | *1* | *0.08* | *0.6* |
| $E_3$ | *1* | *0.01* | *0.01* |
| $E_4$ | *1* | *0.01* | *0.4* |
| $E_5$ | *1* | *0.6* | *0.08* |

***Table 1.*** *Taylor Polynomial Coefficients for each emitter*

The aforementioned IA, HHT and Bispectrum features are generated as both greyscale and RGB images and resized in the network to different resolutions [250x250] and [300x300] to test the resolution effect on learning. Bispectrum of the signal generated using MATLAB HOSA Toolbox *bispecd* function.

## 3.2 Network Architecture

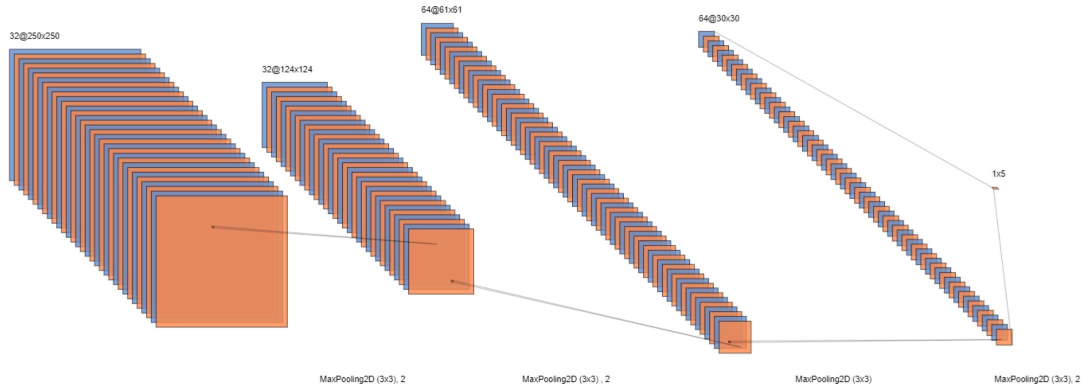

***Figure 6.*** *The network architecture used in the project*

The CNN network is created using Tensorflow 2.4 and trained using NVIDIA GeForce GTX 960M GPU. Adam and RMSProp optimizers are used with α=0.05 initial learning rate with a reduction after 15 epochs without any validation loss improvement by a factor of 0.1. Categorical cross-entropy is chosen as the loss function and accuracy metric is tested. Batch size 15 and epoch size 200 used for the training.

A [0-1] normalization preprocessing layer is applied to the data before entering to network. To prevent overfitting, different regularization methods including batch normalization, dropout, spatial 2d dropout and activity regularization are tested. Rectified Linear Unit (ReLU) and the advanced version of it LeakyReLU is used for the activation function between convolutional layers and softmax activation function is used at the last fully connected output layer.

## 4. EXPERIMENTS

For the experiments, 3 different data types are generated: Upper envelope image, Hilbert spectrum image and Bispectrum image. These images are generated in 2 different resolution and with 2 different signal setups as described in chapter 3. Without loss of generality, it is acceptable to say that all data types gave the same result.

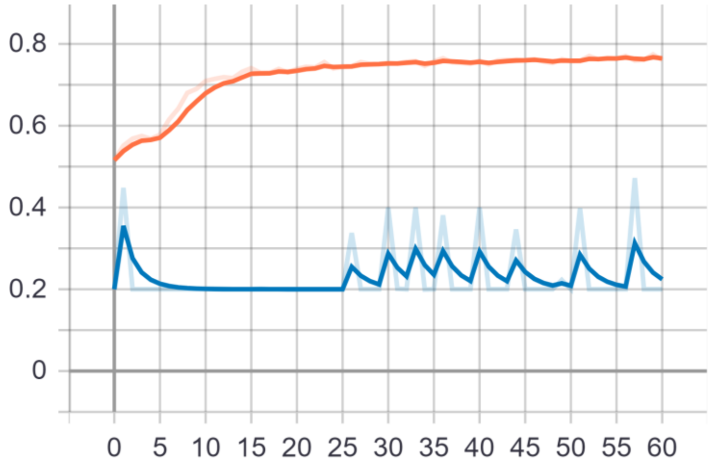

***Figure 7.*** *Accuracy graph per epoch with no noise data*
***Orange** is training accuracy and **blue** is validation accuracy*

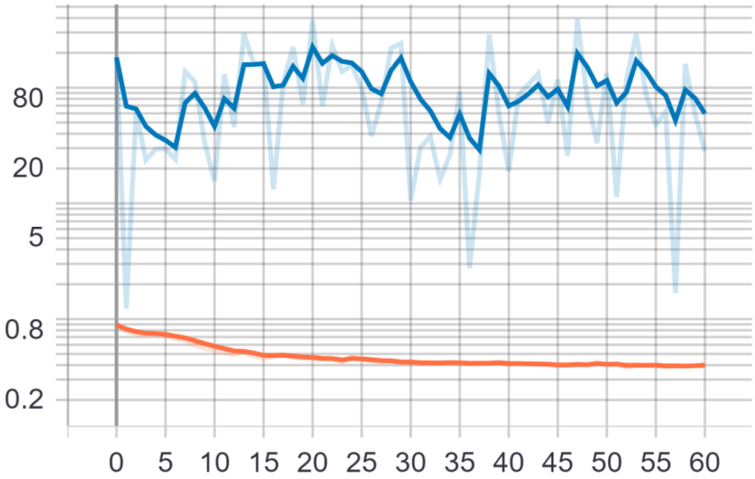

***Figure 8.*** *Loss graph per epoch with no noise data*
***Orange** is training loss and **blue** is validation loss*

When tested with no noise data, all networks overfitted to the training data and could not learn the general representation of the UMOP features inside the image. Reasons can vary. Dataset can be too small to teach every variation to the network or problem can be too complicated for the network to learn and the network is just memorizing. Because of the physical limitations of the test PC’s GPU, tests are made with only 20 samples at each batch. Considering that weights are updated at each batch and we have 5 different emitters, at each batch class distributions probably were not balanced thus variations were not enough for the network to learn properly. Regulations are applied to the network in order to eliminate overfitting and data size is increased but improvement could not be achieved. Around 80% training accuracy achieved but validation accuracy was only 20%.

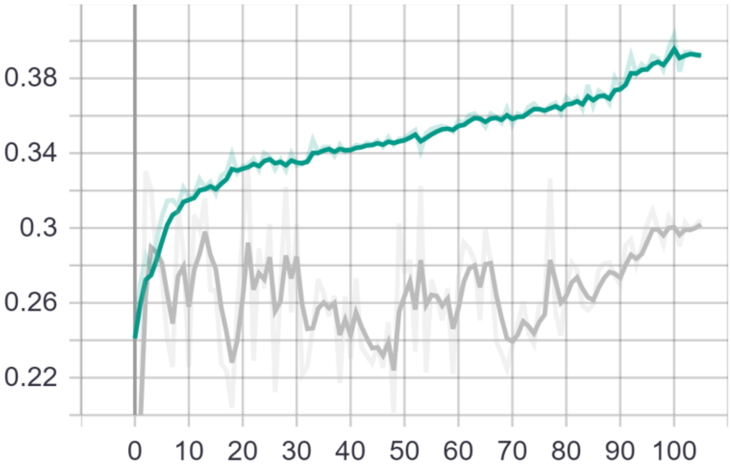

***Figure 9.*** *Accuracy graph per epoch with noisy data*
***Green*** *is training accuracy and* ***gray*** *is validation accuracy*

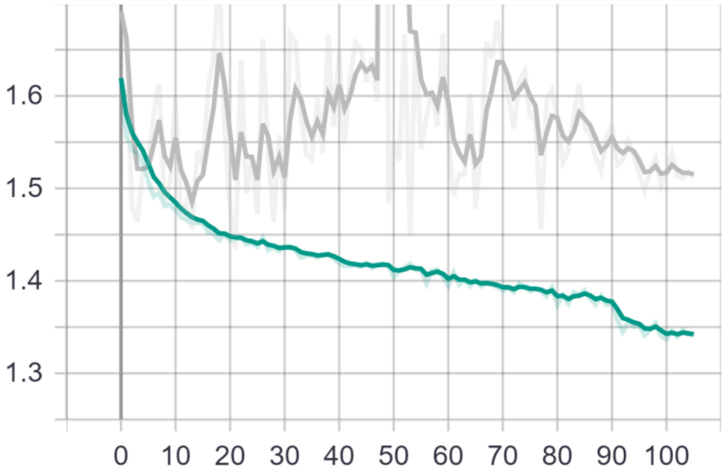

***Figure 10.*** *Loss graph per epoch with noisy data*
***Green*** *is training loss and* ***gray*** *is validation loss*

When tested with noisy data between 10-20 SNR values, overfitting is eliminated but this time learning was slow. Generally, both training and validation accuracies could not go higher than 40% within 200 epochs but in some network structures with HHT image input both accuracies were able to achieve around 60% per cent accuracy. If trained for high epochs, like >1000, the network has the potential of achieving considerably good results. However, the test system was running out of memory and Google Colab is not allowing long training times. Moreover, results could suggest that image representations as 2D or 3D data can be too complicated to learn for small networks with small, simulated data. Because of the author's resource limitations, bigger networks with bigger data could not be realized in this project as with other mentioned possible experiments.

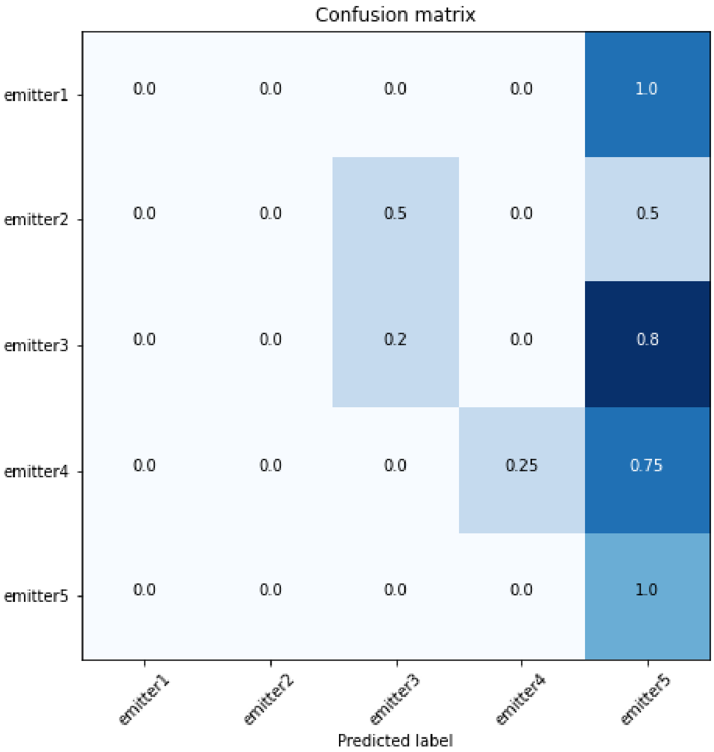

***Figure 11.*** *Confusion matrix of noisy validation data*

For the data, signal modelling and representation of it, as an image could be the wrong choices. Real-world data must be preferred for the realistic presence of nonlinearities. Instead of using whole image data with lots of unnecessary information a more compressed version of the feature representation ought to be considered.

## 5. CONCLUSION

In literature, works mostly focus on dimensionality reduction for Bispectrum images and handcrafted features like entropy for HHT images and rise time, overshoot etc. for time-domain images. Results in this project support that, UMOP feature learning from high dimensional representations like images of bispectrum, HHT etc. is hard to achieve thus implementing extra steps for feature extraction can be sensible. However, these results do not completely prove that the proposed method is unusable but further research and action must be considered with better equipment and real data.

### 5.1 Realistic Constraints and Discussion

Several constraints affect the results of this project. First one is the equipment quality and availability. Because we are experiencing the exceptional problems of a Covid-19 pandemic, I was not able to use the facilities of the Boğaziçi University, including its internet speed and university computers with high GPU speed and memory. Using my own means, I used 8Mbit internet connection and an Nvidia 960M Laptop GPU for the trainings. These conditions resulted in the ability to train only a small size network, ability to use only a small size dataset and training the algorithm for a small time and thus for the small number of epochs.

The second one is the time limit. This is a one-term project, and the success of the project depends on the ability to generate meaningful data and train a deep learning network. Several events regarding the university significantly hindered the progress of the project.

Third and maybe the most important one is my ability to reach expert knowledge. Because we are not in the university, my communication with other students and with my professors are limited, I was not able to ask everything, I needed in time and I lost lots of time in the process.

### 5.1.1 Social, Environmental and Economic Impact

This project planned to have an impact on the budgets reserved for the R&D projects in the industry. Results confirm that for successful follow-up research economics constraints must be discussed. Further successful research can improve the defence capabilities of a country and its ability to secure wireless networks. Also, new progress is useful for cognitive radio and self-organizing networks.

### 5.1.2 Cost Analysis

Results showed that to train a successful deep learning algorithm which works on images, one should have a solid GPU. With an improved GPU, the network can accept bigger size images, can accept them in higher batch sizes, can hold more parameters in memory and can train the network for more batches. This type of GPU (for example an RTX 2060) is around 12k TL in markets today. There are two other alternative methods to this which are not feasible also in my experience. Hiring a GPU from the internet, which can cost also a huge amount of money because we do not know how many trainings and trials we need for this problem. The second one is using free Google Colab but results also showed that training with a small dataset can easily result in overfitting and without a really strong internet connection uploading ~1GB of new data for every new data we generate is a time-consuming approach.

Moreover, probably using real-world data would bring more success to the proposed algorithms thus a real-world data collection mechanism is required. This can cost us a considerable amount depending on the university equipment and the support and the willingness of authors in the references to share their own data.

### 5.1.3 Standards

This project follows IEEE, IET, EU and Turkish standards without compromising form engineering code of conduct.

## 5.2 Future Work

These results do not necessarily mean the failure of the algorithm. More research on the discriminativeness of the data can be done but more necessarily, the proposed method should be applied on a proper deep learning model with suitable parameters and sizable data size. These methods must also be applied to a real-world dataset. Moreover, there are other feature extraction methods which are mentioned in this project but not implemented yet. These methods should be tested with their suitability for deep learning applications.

# BIBLIOGRAPHY

[1] R. Xiaohu, Y. Haohuan, Liu Zheng, H. Zhitao, W. Fenghua, J. Wenli, “An experimental study on secondary radar transponder UMOP characteristics”, *IEEE 2016 European Radar Conference (EuRAD)*, London, 2016, pp. 314-317.

[2] X. Ru, Z. Liu, Z. Huang and W. Jiang, "Evaluation of unintentional modulation for pulse compression signals based on spectrum asymmetry," in IET Radar, Sonar & Navigation, vol. 11, no. 4, pp. 656-663, 4 2017, doi: 10.1049/iet-rsn.2016.0248.

[3] Yipeng Zhou, Xing Wang, You Chen, Yuanrong Tian, "Specific Emitter Identification via Bispectrum-Radon Transform and Hybrid Deep Model", Mathematical Problems in Engineering, vol. 2020, Article ID 7646527, 17 pages, 2020. https://doi.org/10.1155/2020/7646527.

[4] Degang Sun, Yiwei Li and Yanyun Xu, "Specific emitter identification based on normalized frequency spectrum," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2016, pp. 1875-1879, doi: 10.1109/CompComm.2016.7925027.

[5] L. Wu, Y. Zhao, Z. Wang, F. Y. O. Abdalla and G. Ren, "Specific emitter identification using fractal features based on box-counting dimension and variance dimension," 2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Bilbao, 2017, pp. 226-231, doi: 10.1109/ISSPIT.2017.8388646.

[6] H. Ye, Z. Liu and W. Jiang, "Comparison of unintentional frequency and phase modulation features for specific emitter identification," in Electronics Letters, vol. 48, no. 14, pp. 875-877, 5 July 2012, doi: 10.1049/el.2012.0831.

[7] A. Aubry, A. Bazzoni, V. Carotenuto, A. De Maio and P. Failla, "Cumulants-based Radar Specific Emitter Identification," 2011 IEEE International Workshop on Information Forensics and Security, Iguacu Falls, 2011, pp. 1-6, doi: 10.1109/WIFS.2011.6123155.

[8] Tao-wei Chen, Wei-dong Jin and Jie Li, "Feature extraction using surrounding-line integral bispectrum for radar emitter signal," 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 2008, pp. 294-298, doi: 10.1109/IJCNN.2008.4633806.

[9] S. Deng, Z. Huang and X. Wang, "A novel specific emitter identification method based on radio frequency fingerprints," 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), Beijing, 2017, pp. 368-371, doi: 10.1109/CIAPP.2017.8167241.

[10] L. J. Wong, W. C. Headley, S. Andrews, R. M. Gerdes and A. J. Michaels, "Clustering Learned CNN Features from Raw I/Q Data for Emitter Identification," MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, 2018, pp. 26-33, doi: 10.1109/MILCOM.2018.8599847.

[11] L. J. Wong, W. C. Headley and A. J. Michaels, "Specific Emitter Identification Using Convolutional Neural Network-Based IQ Imbalance Estimators," in IEEE Access, vol. 7, pp. 33544-33555, 2019, doi: 10.1109/ACCESS.2019.2903444.

[12] Y. Pan, S. Yang, H. Peng, T. Li and W. Wang, "Specific Emitter Identification Based on Deep Residual Networks," in IEEE Access, vol. 7, pp. 54425-54434, 2019, doi: 10.1109/ACCESS.2019.2913759.

[13] Z. Zhou, J.-K. Zhang, and T. Zhang, "Specific Emitter Identification via Feature Extraction in Hilbert-Huang Transform Domain," *Progress In Electromagnetics Research M*, Vol. 82, 117-127, 2019. doi:10.2528/PIERM19022502

[14] D. Ren and T. Zhang, "Specific emitter identification based on intrinsic time-scale-decomposition and image texture feature," 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, 2017, pp. 1302-1307, doi: 10.1109/ICCSN.2017.8230320.

[15] L. Cain, J. Clark, E. Pauls, B. Ausdenmoore, R. Clouse and T. Josue, "Convolutional neural networks for radar emitter classification," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2018, pp. 79-83, doi: 10.1109/CCWC.2018.8301627.

[16] J. Gong, X. Xu and Y. Lei, "Unsupervised Specific Emitter Identification Method Using Radio-Frequency Fingerprint Embedded InfoGAN," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2898-2913, 2020, doi: 10.1109/TIFS.2020.2978620.

[17] T. J. O'Shea, N. West, M. Vondal and T. C. Clancy, "Semi-supervised radio signal identification," 2017 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, 2017, pp. 33-38, doi: 10.23919/ICACT.2017.7890052.

[18] S. D'Agostino, "Specific emitter identification based on amplitude features," 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, 2015, pp. 350-354, doi: 10.1109/ICSIPA.2015.7412216.

[19] Chen, Xi, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets." *Advances in neural information processing systems* 29 (2016): 2172-2180.

[20] S. D'Agostino, G. Foglia and D. Pistoia, "Specific Emitter Identification: Analysis on real radar signal data," 2009 European Radar Conference (EuRAD), Rome, 2009, pp. 242-245.

[21] J. Zhang, F. Wang, O. A. Dobre and Z. Zhong, "Specific Emitter Identification via Hilbert–Huang Transform in Single-Hop and Relaying Scenarios," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1192-1205, June 2016, doi: 10.1109/TIFS.2016.2520908.

[22] L. J. Wong, W. C. Headley and A. J. Michaels, "Estimation of transmitter I/Q imbalance using convolutional neural networks," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2018, pp. 948-955, doi: 10.1109/CCWC.2018.8301715.

[23] L. Ding, S. Wang, F. Wang and W. Zhang, "Specific Emitter Identification via Convolutional Neural Networks," in IEEE Communications Letters, vol. 22, no. 12, pp. 2591-2594, Dec. 2018, doi: 10.1109/LCOMM.2018.2871465.

[24] L. Xuan-min, Y. Ju and Z. Ya-jian, "A New Method Based on Local Integral Bispectra and SVM for Radio Transmitter Individual Identification," 2010 WASE International Conference on Information Engineering, Beidaihe, Hebei, 2010, pp. 65-68, doi: 10.1109/ICIE.2010.305.

[25] F. A. Uçkun, H. Özer, E. Nurbaş and E. Onat, "Direction Finding Using Convolutional Neural Networks and Convolutional Recurrent Neural Networks," 2020 28th Signal Processing and

Communications Applications Conference (SIU), Gaziantep, Turkey, 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302448.