Queensland University of Technology

FINAL REPORT

AI Signal Classification Challenge Application of Deep Learning On Blind Radio Frequency Analysis

Candidate:
Alexander Iftene N9657533
Team Members:
Waldo Fouche N9950095
Devin Da Silva Martins N9968601
Supervisor:
Veronica Gray
School Name:
Science and Engineering Faculty
Queensland University of Technology - QUT

Contents

List of Figures

1 Abstract

This project aims to produce an artificially intelligent system using deep and machine learning algorithm techniques to optimally identify blind radio frequency signals given from various databases and resources. The application of this system will be to aid Electronic Warfare Officers to identify and react to these signals in the battlefield. The Army have provided the team with a brief and all the resources and databases necessary in order to complete this mission. The team managed to create a machine learned model that is able to predict signals with an accuracy of 97.23%.

2 Introduction

From recent research into automatic modulation classification, the literature indicates that this is a massively complex problem with incredible applications in signals intelligence. When observing the contemporary literature, most fail to delve deep enough to solve the discrepancies existing when the signal to noise ratio is high and or long observations of the signal are available. Unfortunately for the Rapid Capability Office (RCO), this does not bode well for future developments in electromagnetic warfare to help deal with optimising automated signal classification. This project aims to delve deep into machine and deep learning concepts to produce an artificially intelligent system that can classify blind radio frequency signals. This effort invites the development of machine learning algorithms and the supporting processes, methods and tools needed to improve the speed and agility of blind signal identification and classification within the electromagnetic spectrum. The resulting product will hopefully lead to advancements in how Electronic Warfare Officers identify and react to these signals on the battlefield. The Government intent is for solvers to present advanced algorithms and AI implementations with a high degree of classification accuracy and performance (e.g., speed of classification, low CPU resource requirement) that would allow for ease of integration within existing systems. [2] This research goal is to promote innovation and advancement in the area of signal processing. Ideally, if the team were to produce a system with such efficiency and applicability, the Army will grant a $150,000 cash prize for the success. With this incentive in mind, the team has been provided with resources to aid in the training of the system to identify test signals from certain databases.

Signal analysis is a conventionally broad topic with multiple applications across a huge variety of disciplines ranging from medical to defence. Nowadays, there is much work that is being conducted in the medical industry using artificially intelligent signal identification systems to diagnose certain conditions such as heart and brain diseases. Developing technology for any of these departments would create avenues for others to improve and adjust to work for certain areas, which is why this project has such value to the modern day era of technology, specifically to do with artificially intelligent systems. The future of technological development is directing itself towards an artificially intelligent future, where the implementation of deep and machine learning algorithms will aid many endeavours.

Contemporary systems incorporate the assumption that the transmitter and the receiver are cooperative and have a full understanding of the wave-forms that are exchanged. Realistically however, this is not always the case as the modulation or the coding of the waveform may have been changed before transmission. Typical examples of such scenarios involve cognitive radio network and the intervention of a signal for intelligence purposes. There has already been an extensive amount of research done on this issue, for example, a research paper from the late 2000s that delves deep into modulation recognition which is extremely important in communication intelligence, otherwise referred to as COMINT, indicates another methodology mainly using Fourier transformations of different analogue modulation types to set up a recognition procedure. This process was carried out and found that all types of analogue modulation was classified with a success rate of approximately 90% at a signal to noise ratio (SNR) of 10db. [1]

Modulation recognition is a key component in identifying improper demodulators when transferring signals as it may damage the signal context. This is an issue as this can conflict with the deciphering process which converts the demodulated message from its initial ciphered state. This is the main process in converting a non-intelligible signal that cannot be relayed or processed for interpretation, to an intelligible and digestible signal. To add to the importance of modulation recognition, understanding the correct modulation type assists to recognise a possible threat and determine an appropriate course of action, whether that may include jamming the signal or redirecting it.

This paper concluded that the team managed to simulate approximately 12 different analogue modulation signal types at different SNR. The threshold for SNR for correct modulation recognition is around 10dB which is an overall improvement in the SNR threshold at the time. However, the effectiveness of modulation recognition on digital signals is still an ambiguity because the algorithms applied to analog systems would not necessarily translate directly to a more complex signal such as a digitally modulated signal.

The US Army conducted work on deep learning in 2019 for RF signal classification in unknown spectrum environments where they utilize the signal classification results in a distributed scheduling protocol. Wireless networks are prone to in-network and out-network interference such as jammers, path loss, fading and traffic congestion in the network. In order to support a concept called dynamic spectrum access (DSA), a necessity to derive the interference from the spectrum from hidden sources became a mission. This would work by certain users employing signal classification scores to make channel access decisions and share the spectrum with each other while avoiding interference with out-network users and jammers. Their team managed to show that distributed scheduling constructed upon signal classification results provided major improvements to user throughput and out-network user success ratio. The team applied continual learning to a Convolutional Neural Network (CNN) using an Elastic Weight Consolidation (EWC) based loss. This was then used to detect unknown signals using an outlier system that was applied to outputs of the convolution layers Minimum Covariance Determinant (MCD) and k-means clustering methods. The final stage of their process was to filter the interference signals through a blind source separation technique called Independent Component Analysis (ICA). [3]

These neural networks that are being mentioned are the means for the computer to perform a task by analyzing training examples which are hand-labelled in advance. An example amongst others that have been previously mentioned can be object recognition which is one of the most popular uses of machine learning and the neural networks involved to identify certain objects by training these systems using labelled images of the same type. These neural networks are actually modelled loosely on the human brain, a sort of Darwinian attempt by humans to construct a brain like machine for use in many different departments of science and technology. These neural networks operate using nodes which might be connected to several other nodes in the layer above or beneath it from which it receives data or sends data to the aforementioned nodes on different layers. These nodes have weights and these wights have corresponding data attached to them with a certain number that is multiplied over each of the connections associated thereof. If the number exceeds the threshold value, it sends the sum of the weighted inputs to the next connection and the process continues to other nodes on different layers.When the net is being trained all the weights and thresholds are set random values and training data is sent to a variety of layers and it passes through the succeeding ones which is the mathematically computed into an output layer.

3 Materials and Methods

The method of this project was computationally heavy utilising Python, MATLAB and frameworks such as TensorFLow for training the data sets and Simulink to graphically model the radio signals used for training the machine learning model. Using Python's TensorFlow framework, the idea was to be able to input the signals generated from MATLAB using the Simulink environment in order to create neural networks to be used for an accurate model generation.

The exact machine learned model was developed using layers of neural networks derived from Convolutional Neural Networks (CNN) where we used images of signals that were parsed through the model for training using approximately 5 to 10 epochs with a set population depending on the input arguments of data.

4 Results and Discussion

The team decided to use the following signals that were parsed through Simulink and then sent to the TensorFlow framework for modelling. The signals that were primarily used for training the data set were the 4G LTE Uplink and Downlink, 5G Uplink and Downlink, and three different variations of the same wifi signal called WLAN 802.11. An example simulation of the 5G signal spectrogram through the MATLAB Simulink graphical framework is show below to demonstrate the frequency and samples of this specific signal and what characteristics could be dissected using this program.

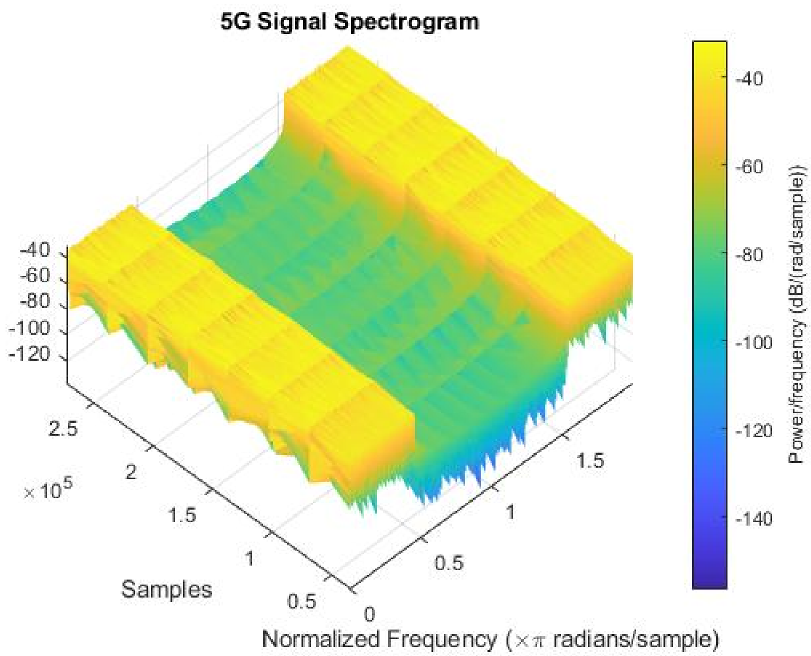


Figure 1: 5G Signal Spectrogram in frequency domain using MATLAB Simulink.

All of the trained signals were generated using MATLAB Simulink and the spectrograms were computed isometrically and in a 2D fashion to graphically simulate a more traditional understanding of the different signal's characteristics, as shown in the diagram below. Here we see the normalised frequency of the signal versus the samples computed in MATLAB which have generated this flat version of the 5G signal spectrogram.

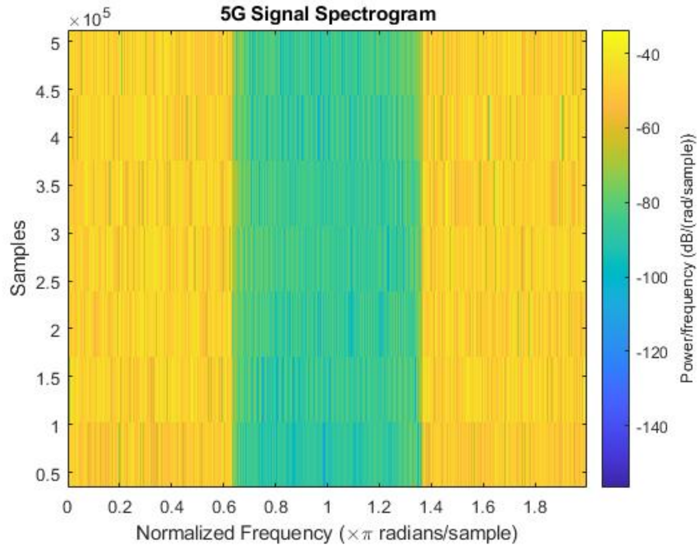


Figure 2: Flat 5G Signal in frequency domain using MATLAB Simulink.

As the model required the team to understand more than just the typical dimensions of a signal, the up-link of the signals were also obtained using MATLAB. Up-link pertains to radio communication service where the up-link is the portion of a feeder link used for the transmission of signals from a station to another. A demonstration of this is depicted below where the up-link graph is modelled in the time domain as opposed to the frequency domain in the previous examples. The reason why the signals had to be converted to image files is because it is a lot easier to parse in image files to train as opposed to .wav or .mat files. These alternative files in their original form cannot be parsed into reliable data sets or stored because neural networks are not designed to compute such varied formats of information especially in the form of signals which are more easily read by programs such as MATLAB and Python's NumPy framework which was actually used to plot the differences in data sets. These sets were loaded in as labels which were then parsed into arrays to be computed later on for modelling via graphical means or for the purpose of training. This is why it was so important to use Simulink to graphically model the desired signals before porting them to our machine learning model for training, otherwise it would have been extremely hard to near impossible to achieve any sort of reliable predictions from our model.

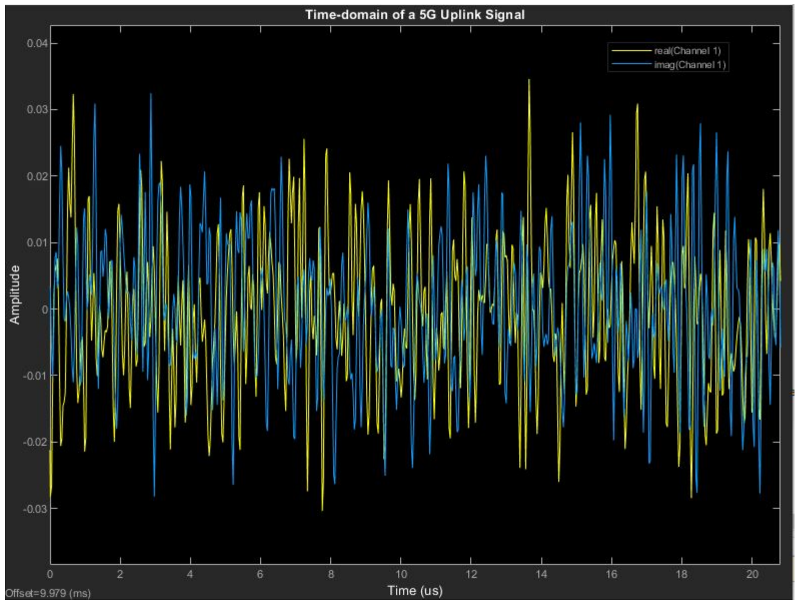


Figure 3: 5G upling Signal modelled in the time domain.

During the pre-processing phase, the data from the signals was parsed through to make them the same size and to convert the images into grey scale, as dealing with images that aren't in grey scale format make it difficult to model from. These signals were then plotted using the framework Matplot which serves as a mathematical framework to plot data types for interpretation, this was used throughout the project to model the pre-processed data and also the final prediction graphs that compare the predictions to the trained accuracy of the models. These images were then flattened in order to be computed through CNNs, in other words the dimensions of the images were changed so that the neural networks could process the data as a single array with simple dimensions. A demonstration of the pre-processed images that were used from all the different selected signals is shown below in Figure 6.

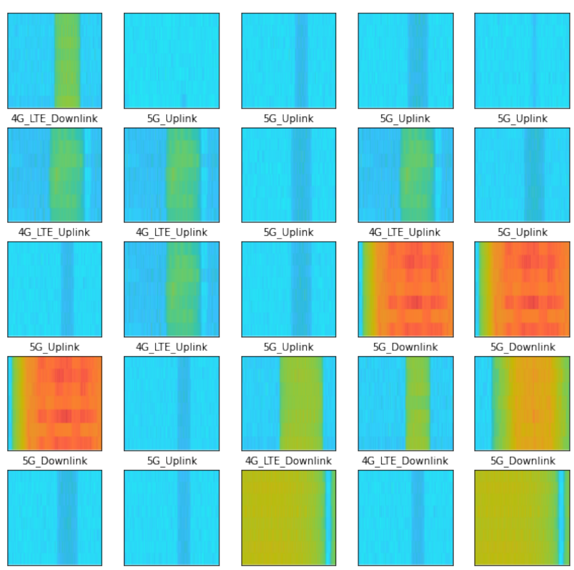


Figure 4: Pre-processed greyscaled images of 5 different types of signals.

Sequential layering of the CNNs was utilised because it is the easiest way to build a model layer by layer, however, as there are layers to this system, each layer is essential in the result of the model. The first 2 layers deal with the input of the images which are seen as 2-dimensional matrices where 32 and 64 nodes are inputted which depend on the size of the dataset and can be lowered or increased accordingly. Because the dataset is relatively small, 32 or 64 nodes for the layering is adequate to produce reliable results. The kernel size of a filter matrix is incredibly important for this step as a kernel is responsible for the amount of times the convolution multiplies over a certain amount of pixels until all the pixels are covered. For this instance a filter matrix of 3x3 was necessary to allow for the rectified linear activation to work well in the neural network. The following output of this model is shown below after computing a dense layer of 7 nodes corresponding to the amount of signals used for training.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 498, 498, 32)      896
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 249, 249, 32)      0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 247, 247, 64)      18496
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 123, 123, 64)      0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 121, 121, 64)      36928
_________________________________________________________________
flatten_1 (Flatten)          (None, 937024)            0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                59969600
_________________________________________________________________
dense_3 (Dense)              (None, 7)                 455
=================================================================
Total params: 60,026,375
Trainable params: 60,026,375
Non-trainable params: 0
_________________________________________________________________
```

Figure 5: Modelling of a sequential layering system using convolutional layers using keras.

The model was then compiled using three parameters; optimizer, loss and metrics. In this instance the optimizer controls the learning rate where it can be adjusted accordingly and so adam was the best choice. Adam is generally a popular optimizer for many cases because it can adjust the learning rate throughout the training of the data sets. The learning rate determines how fast the optimal weights for the model is calculated, this is explained by the fact that slower learning rates can opt for more accurate weights however the time taken to compute is longer, and vice versa for faster learning. Categorical cross-entropy was used as the loss function which is important as it indicates how the model is performing and is the most reliable choice for this instance.

The prediction of the models was computed and outputted graphically to show the differences in the preidctions for each of the signals that were computed previously. Demonstrations of the computation using just the 5 signals and the 7 signals are demonstrated below and the percentage accuracy's show the the model after the processing of the layers are done and then trained.

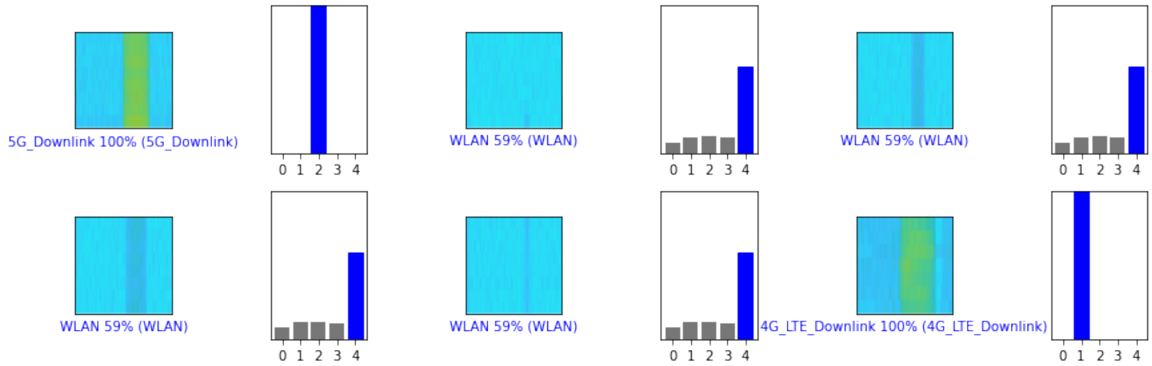


Figure 6: Modelling of the predictions of using 5 signals as the data set.

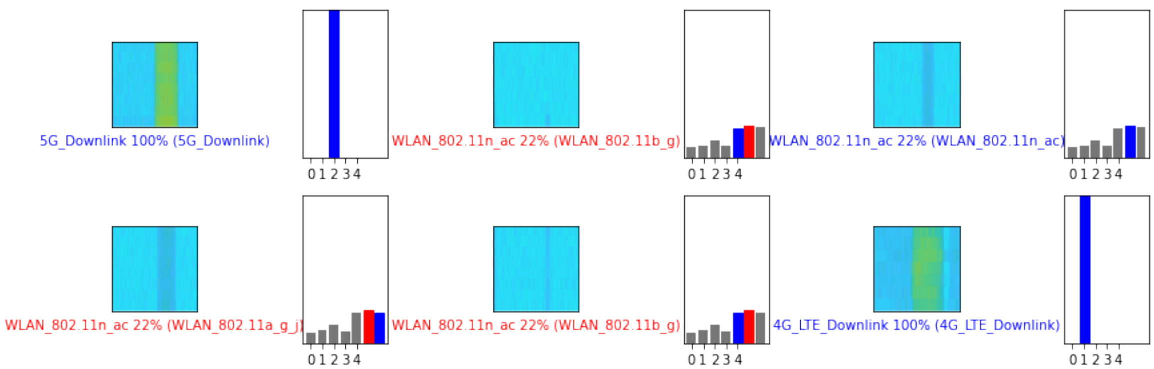


Figure 7: Modelling of the predictions of using 7 signals as the data set.

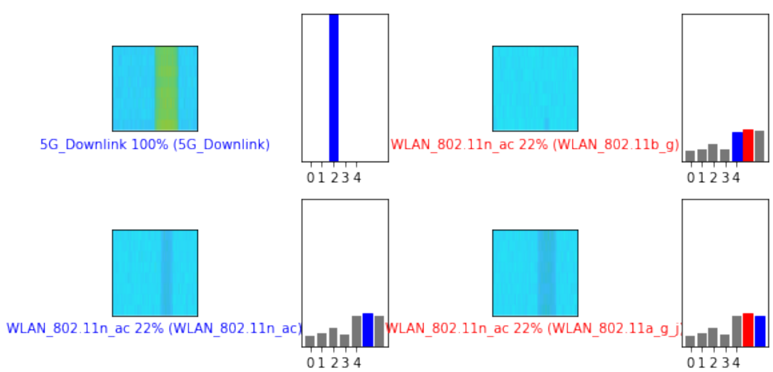
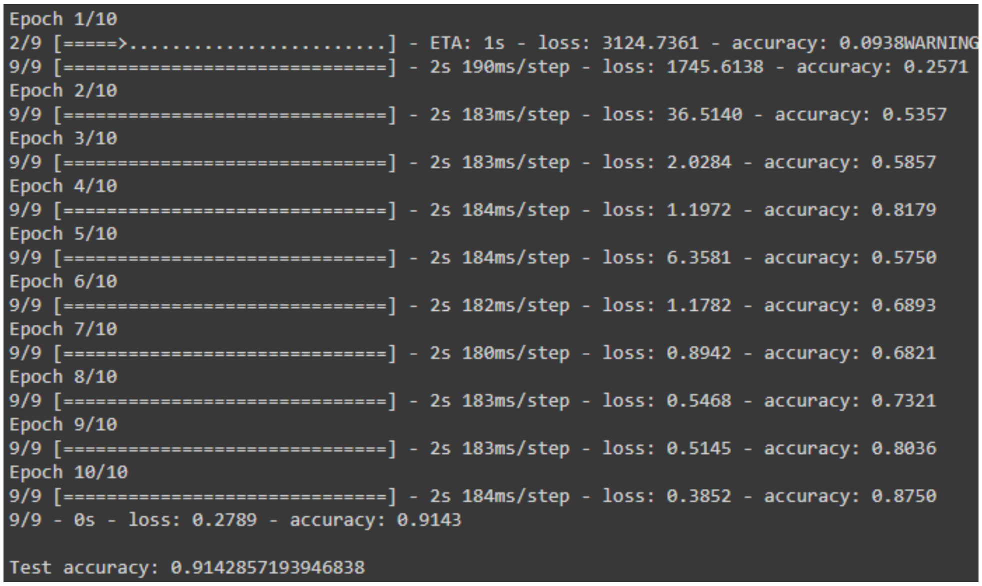


Figure 8: Continued modelling of the predictions of using 7 signals as the data set.

The final trained model was computed using 10 epochs which is the amount of times the model will cycle through the data. The more epochs that run, the more the model will improve and provide a more accurate result. The final results of the trained data set demonstrate an accuracy of approximately 91% using the 7 signals as data sets. In this data set, there were 3 different types of WLAN as previously mentioned in the report. However, when the data set was trained using just the 5 signals and using only one generic WLAN signal, the trained data set was considerably higher and it achieved an output accuracy of 97.23%. Below are demonstrated the differences in the



```
Epoch 1/10
2/9 [=====>........................] - ETA: 1s - loss: 3124.7361 - accuracy: 0.0938WARNING
9/9 [==============================] - 2s 190ms/step - loss: 1745.6138 - accuracy: 0.2571
Epoch 2/10
9/9 [==============================] - 2s 183ms/step - loss: 36.5140 - accuracy: 0.5357
Epoch 3/10
9/9 [==============================] - 2s 183ms/step - loss: 2.0284 - accuracy: 0.5857
Epoch 4/10
9/9 [==============================] - 2s 184ms/step - loss: 1.1972 - accuracy: 0.8179
Epoch 5/10
9/9 [==============================] - 2s 184ms/step - loss: 6.3581 - accuracy: 0.5750
Epoch 6/10
9/9 [==============================] - 2s 182ms/step - loss: 1.1782 - accuracy: 0.6893
Epoch 7/10
9/9 [==============================] - 2s 180ms/step - loss: 0.8942 - accuracy: 0.6821
Epoch 8/10
9/9 [==============================] - 2s 183ms/step - loss: 0.5468 - accuracy: 0.7321
Epoch 9/10
9/9 [==============================] - 2s 183ms/step - loss: 0.5145 - accuracy: 0.8036
Epoch 10/10
9/9 [==============================] - 2s 184ms/step - loss: 0.3852 - accuracy: 0.8750
9/9 - 0s - loss: 0.2789 - accuracy: 0.9143

Test accuracy: 0.9142857193946838
```

Figure 9: Compilation of the learned model of 7 signals using 10 epochs and computing an accuracy of approximately 91%

The team managed to model the differences in training accuracy between the 2 different inputs of signals, using 5 signals and then 7 signals interchangeably to see the discrepancies. What the team had discovered is that the model is very efficient at learning using completely different signals that have different frequency and bandwidth characteristics, however, once multiple signals of the same type are introduced the model finds it difficult to discern the key components and then the ability to train the model reliably becomes a distant reality. A demonstration of the above is shown in the graphs which show the epochs versus accuracy of the trained models and also the differences in loss correspondingly.

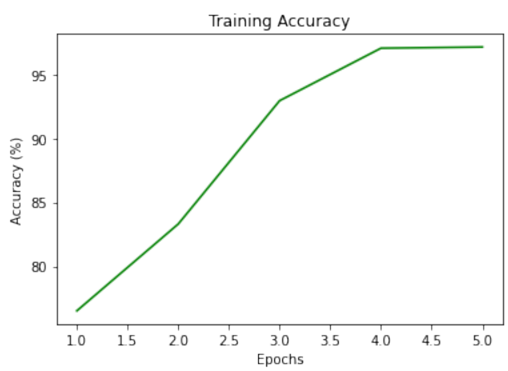


Figure 10: Training accuracy of 5 signal data set with highest accuracy model

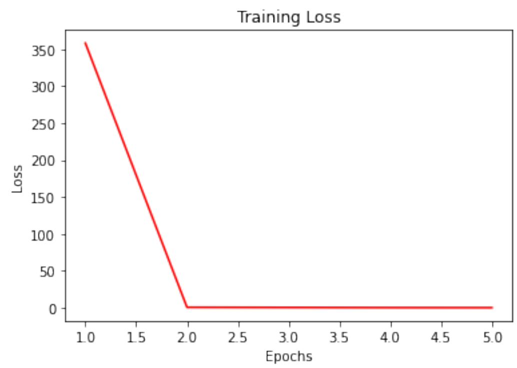


Figure 11: Training losses of 5 signal data set with highest accuracy model

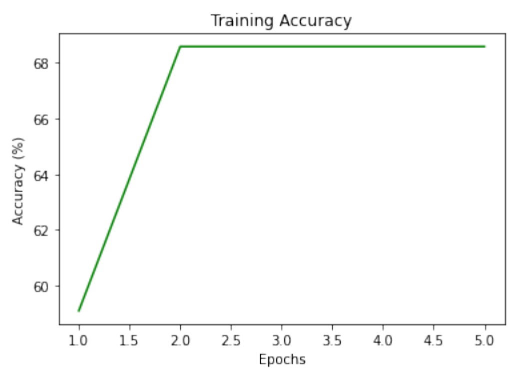


Figure 12: Training losses of 7 signal data set with lowest accuracy model

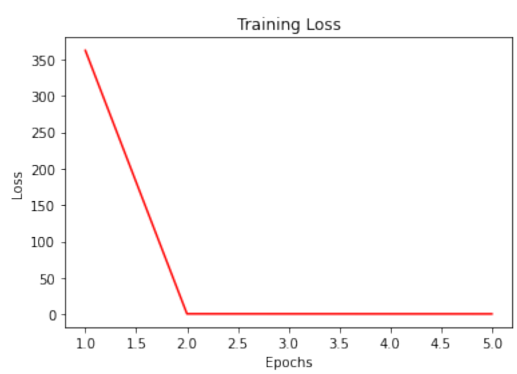


Figure 13: Training losses of 7 signal data set with lowest accuracy model

As it can be seen from the above graphs generated from the finalised machine learned models, the results from the different sets of data are shown inherently in the model's performance as the epochs were cycled through. The higher accuracy model demonstrated a progressive improvement in learning as the epochs were cycled through and peaked at the final stage, adversely, when observing the loss of the model it is seen that as the epochs were cycled through the loss progressively decreased. This is fantastic and is exactly what the team wanted when applying the model to the data set. Alternatively, when looking at the model obtained from the data set with 7 signals, the accuracy reaches a plateau when the epochs reach 2 cycles which is not ideal for a model which is supposed to improve from cycle to cycle theoretically. Observing the loss graph of this data set shares similar characteristics of the previous one which shows promising signs of development. However, the theoretical statement that a machine learned model if compiled correctly should be able to get as close to 100% as possible given the amount of epochs it is able to cycle through. It is clearly not the case with the second data set, and it can be attributed to the fact that as previously stated, the model is not efficient enough at differentiating between similar signal types as opposed to signals which have vastly different characteristics.

5 Conclusion

Conclusively, the trained model has demonstrated that it can achieve an accuracy of 97.23% given a restricted amount of data that varies greatly from each other, but has a lower accuracy of around 91% and sometimes even 60% depending on the training of the data sets. Given this, it can be said that the biggest issue with machine learning derives from the amount of data that can be inputted. Machine learning works best when the amount of data that is being processed constitutes of varied sources and in large amounts, which provides diversity for the training of the model so that it can more accurately predicted other signals. This was the bane of this project, the data given was not enough and made it hard for the team to train the model accurately enough to accommodate for a wide variety of different signals with unique characteristics. The aim of this project was achieved however the process of obtaining data could have been greatly optimised to allow for a more improved and reliable machine learning model.

References

[1] A.K. Nandi and E.E. Azzouz. Automatic analogue modulation recognition. *Signal Processing*, 46(2):211 – 222, 1995.

[2] Army Rapid Capabilities Office. Army signal classification challenge. "https://www.challenge.gov/challenge/army-signal-classification-challenge/", 2020. [Online; accessed 10-August-2020].

[3] Yi Shi, Kemal Davaslioglu, Yalin E. Sagduyu, William C. Headley, Michael Fowler, and Gilbert Green. Deep learning for rf signal classification in unknown and dynamic spectrum environments. *Spectrum*, page 10, 2019.