Enhance Classification Accuracy of Wireless Interference Identification by using Transfer Learning

Thida Shwe
Department of Computer Engineering
Ajou University
Suwon, South Korea
tds@ajou.ac.kr

Young-June Choi
Department of Computer Engineering
Ajou University
Suwon, South Korea
choiyj@ajou.ac.kr

Rajib Paul
Department of Computer Engineering
Ajou University
Suwon, South Korea
rajib@ajou.ac.kr

***Abstract*—The potential growth of wireless devices from different wireless technologies allocated in 2.4 GHz ISM band causes the scarcity of channel resource a severe problem now. Many techniques were used to mitigate the interference occurrence among those different technologies. As the type of solution keeps evolving, Deep Learning has become a popular approach used to solve the interference identification problem lately. In this paper, we try to improve the efficiency of WII classification by adopting a very popular accuracy-enhancing approach namely Transfer Learning (TL) to enhance the interference classification ability of existing WII classification method. The simulation performance of our work proves the significance of TL in the area of wireless signal classification.**

***Index Terms*—Wireless Interference Identification, 2.4GHz band, CNN, Transfer Learning**

I. Introduction

The 2.4 GHz ISM (Industrial, Scientific, and Medical) band is famous for the deployment of different wireless technologies without license needed and free of cost. As the amount of deployed technologies and devices grows exponentially day by day, the resource allocation and collision management have become unavoidable problems. There are several mechanical or hardware based techniques such as [1][2][3][4] were proposed to mitigate the interference between Wi-Fi, Bluetooth, and Zigbee. A sub branch for interference mitigation work, identifying the source of the interference, WII (Wireless Interference Identification) is also becoming an area where researchers dig into lately. Moreover, Deep Learning (DL) has become one of the most popular methods to solve a different kind of problems, there are also several DL based solutions were proposed to solve the interference identification problem as in [5]-[7].

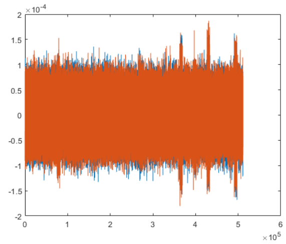
The authors in [5] proposed a DL method to identify the different technologies allocated in the band by using CNN training on the IQ signals dataset which was composed of data from 15 channel indices, 10 of Bluetooth, 3 of Wi-Fi and 2 of Zigbee. In [6], the authors reconfigured the work of [5] by enhancing the method of sampling such as band, SNR and sample selection. In addition, they also extend the type of DL techniques that used to classify the classes from not only CNN but also to ResNet, LSTM and CLDNN and achieve a relatively good result. In [7], the work was again improved by choosing single SNR value for training and also a Boostrap aggregating based algorithm to select the most suitable SNR value at which the best channel classification accuracy can be achieved. All the papers mention above have one common feature: they have well trained neural network model with special ready pre trained weight, and it could be a significant advantage if these weights could be properly made use.

In this paper, we reuse a well trained model from the previous WII classification approach in paper[6] as our pre-trained weight to create a better classification model by using TL. In TL, a previous well trained model's weight is exported to learn a new task in a new model. There are two main features of TL: Fine Tuning and Feature Extraction. In Fine Tuning, the chosen layers of the models can be selected to be retrained again in order to fit the new task better. Whereas, Feature Extraction approach allows the freezing of existing model weight parameter and reusing it by editing usually the last layer of the model according to the features of the new dataset [8]. Unlike [5] and [6], trying to classify among the 15 classes, we manage to classify only between two classes: whether the data belongs to Wi-Fi or Bluetooth. In this paper we re-implement the existing approach [6], and get the initial pre-trained weight. We make a model to classify the type of technology basing on recorded IQ signal values from [9] and achieve a relatively good result. We compared the classification result of model learning from scratch and using TL and illustrate the comparisons in the result section.

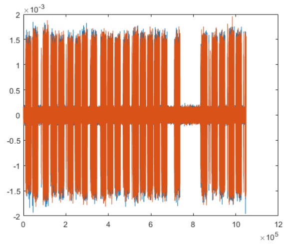
II. Problem Setup

A. Data Preprocessing

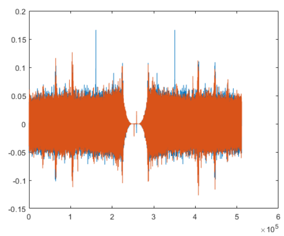
The dataset used in this paper includes inphase and quadrature signal of WiFiand Bluetooth generated by [9]. The complex signals of both WiFi and Bluetooth were collect at center frequency 2.4415 GHz, totally 511488 of Bluetooth and 1279488 of WiFi IQ signals were recorded. In [16], there are 15 channel indices classes (except for the 2422 MHz, 2425 MHz, 2427 MHz and 2430 MHz are overlapped for



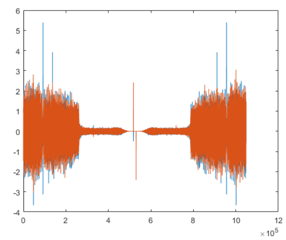
(a) Time domain Bluetooth data



(b) Time domain WiFi data



(c) Frequency domain Bluetooth data



(d) Frequency domain WiFi data

Fig. 1: Visualized Bluetooth and WiFi dataset

two different type of wireless technologies due to the typical ISM band feature) of three technologies: WiFi, Bluetooth, and Zigbee used to be classified. However, in our experiment, we have classified the IQ signals from only two classes WiFi and Bluetooth collected by using the same center frequency, so the classification is more likely a signal content based.

The IQ signals were originally collected in time-domain, we then transform the signal into a frequency based one by using MATLAB and tried to classify both of them by using TL. As described above, the data size of WiFi is more than twice larger than the Bluetooth, for the sake of fairness, we chopped the sample of both WiFi and Bluetooth into 511488 equally. We then reshape the sample size of each of them into (1, 3996, 128, 2), creating 3396 samples of size 128x2 matrix as the input to the neural network just like in [5][6], labeling the WiFi dataset as 0 and Bluetooth dataset as 1. Among the 3396 samples, we set 60% of them, that is 2398 samples as training set and the other 40%, 1598 samples as the testing set. The visualization of the dataset is illustrated in Fig.1.

B. Neural Networks

CNN was mainly used as the neural network throughout our experiment, there are three kinds of CNN network in this paper. Firstly, the reference CNN model which is the model from [6]. Secondly, basic CNN model, same model as reference CNN model but with the last layer being fine tuned to classify our own dataset. And lastly, the TL model, the model we built from pre-trained reference CNN model, used to classify the same dataset as basic CNN model does but with a better initialized pre-trained weight. The architectures of these three models are shown in Fig. 2.

- Reference CNN model, we recovered the CNN from [6] and evaluate the model with the original data provided by the authors. The resulted accuracy is mainly used as a benchmark to compare with the following models we build to classify our new dataset [9] on our new models.
- Basic CNN model, the model we generate exactly same as CNN in [6] but with fine tuned last linear layer, we change the original 15 classifiers into 2 (whether the signal is from WiFi or Bluetooth). The result is also used to compare the performance of the proposed TL model
- TL model, we build this model by using the reference CNN model as the pre-trained model by initializing its weight to the TL model. After loading the initial pre-trained model with existing weight, the convolutional layers are frozen being used as feature extractors, leaving only the linear layers to be trainable to adapt our new dataset features, the last Softmax layer was also fine tuned from 15 into 2 classes. We train the same dataset as the basic CNN model on this TL model. The architecture of these three models are in Fig. 2.

III. Simulation set up and Result

A. Simulation set up

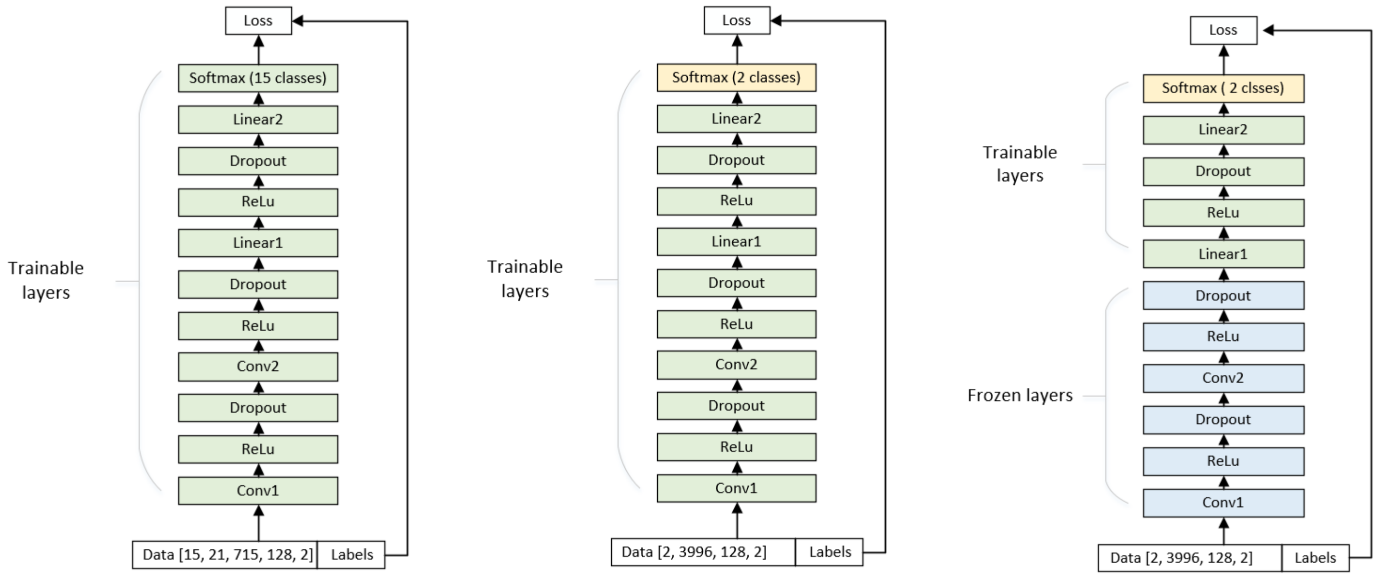
The experiment was run on TITAN Xp from NVIDIA of 12GB on Ubuntu 18.04.4 LTS. we implemented models on the Pytorch platform and preprocess our data by using Matlab. The data was trained for 100 epochs on both basic CNN and TL models, for the evaluation 20 epochs of each were run to visualize the result with less amount of running time. Batch size for Basic CNN model was set to 512 while that of TL model was set to 256. The Dropout rate of both basic CNN model and TL model are 40% for all layers in basic CNN and the first linear in TL model. The learning rate, criterion, optimizer used for both TL model and basic CNN are le-5, CrossEntropyLoss and Adam optimizer respectively. The hyperparameters of three CNN models were listed in Table I.

In the reference CNN model from [6], the dataset to be classified contains data of three categories: WiFi, Bluetooth and Zigbee, however, in our scenario as mentioned in SectionII (A), only contains data of two types WiFi and Bluetooth. Moreover, in our experiment, we eliminate the SNR interference to the data, make the data simpler just to test whether TL could do well on classifying clean data first.

In basic CNN model, the targeted classification objects are the 15 different channel indices (some of them are identical like 2422 MHz, 2425 MHz, 2427 MHz and 2430 MHz) of WiFi, Bluetooth, and Zigbee from where the data was collected. Whereas, in our case, as the data of both WiFi and Bluetooth were collected from the same center frequency 2.4415GHz, the interference type was mainly classified by basing only on the data content itself.

B. Results

For evaluation, we use three kinds of models: the reference CNN model, basic CNN model which is same as reference CNN model but trained with our dataset to classify only 2 classes, WiFi and Bluetooth, and the TL model initialized with reference CNN model's pre-trained weight and fine-tuned to be retrained on our dataset to classify also 2 classes WiFi and



(a) Reference CNN model (b) Basic CNN model (c) TL model

Fig. 2: Model architecture showing the frozen and trainable layers

TABLE I: Hyperparameter list

Parameter Term	Reference Model	Basic Model	TL Model
Conv Layer	256 3*1, 256 3*2	256 3*1, 256 3*2	-
Dense Layer	31744*1024, 1024*15	31744*1024, 1024*256, 256*2	-
Activation Method	ReLu, Softmax	ReLu, Softmax	ReLu, Softmax
Dropout	0.6	0.4	
Optimizer	Adam	Adam	Adam
Learning rate	le-4	le-5	le-5
Criterion	BCELoss	CrossEntropyLoss	CrossEntropyLoss
Classes Name	WiFi, Bluetooth, Zigbee	WiFi, Bluetooth	WiFi, Bluetooth
Classification Object	Channel Indices	Interference Type	Interference Type
Dataset Size	[15, 21, 715, 128, 2]	[2, 3996, 128, 2]	[2, 3996, 128, 2]

Bluetooth. The dataset we use include the time and frequency domain of the complex signal from the same dataset generated by [9].

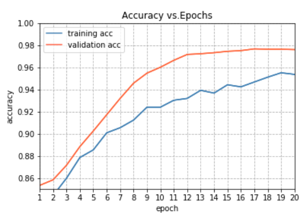
Firstly, we re-implement the reference CNN model with the source code and dataset provided by the authors in [6]. Then we achieve a similar accuracy given in the paper, 89.62% and this will be used as a benchmark against our following results gained from basic CNN model and TL model.

Secondly, we implement the basic CNN model by adopting the same network architecture by just modifying the data training and testing sizes, and the last Softmax layer from the original 15 into 2. Then we tested the model with both the time and frequency domain dataset gaining the accuracies 85.51% and 95.06% respectively. The reason for this improvement on frequency domain compare to time domain is the modification of IQ data into frequency domain. The distinguishing features of the signal become more significant and make the classification goes easier.

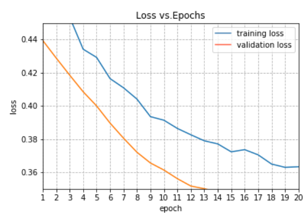
Finally, we used the reference CNN model as the pre trained model and built the TL model. The convolutional layers of TL model were frozen by setting the require gradient to false, leaving only the last two linear layers to be trainable. Thus the gradient will be updated in these two layers during back propagation and the weight can be change along with the training to fit to the new dataset characteristics, and then we fine tuned the last linear layer into a 2 classes Softmax classification layer. Similarly, we run our TL model on both the time and frequency domain datasets for evaluation.

The accuracy of classification we gained from basic CNN model is 85.51% which is quite close to the accuracy from reference CNN model while the accuracy we gained from TL learning model run over time domain IQ dataset is 97.65% which is almost 12% higher than both that CNNs training from scratch. For the frequency domain dataset, the basic CNN model already achieved a relatively good classification accuracy 95.06%, however with better pre-trained weight, the classification accuracy also improved by almost 1% from 95.06% to 95.87%, even though the improvement is not as significant as in time domain dataset, still, the effectiveness of TL concept can be seen clearly.

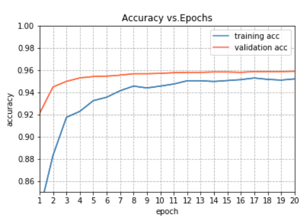
Fig.3 illustrates the accuracy and loss of the TL model run over both the time and frequency domain dataset with 20-epochs. In Fig. 3(a) and Fig. 3(c), the validation accuracy of TL model is higher than the training accuracy. In Fig. 3(a) the validation accuracy starts right from around 88% which is near to the highest accuracy of the basic model and keeps growing up, which vividly express the concept of TL using the existing knowledge and learning from a high position to even higher. And in Fig. 3(b) and 3(d), the loss function of each TL model over time and frequency domain are working well in a descending manner as well.



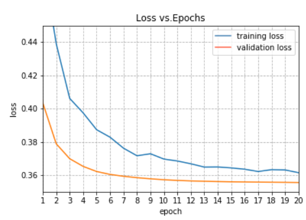
(a) Accuracy of TL model over time domain data



(b) Loss of TL model over time domain data



(c) Accuracy of TL model over fft domain data



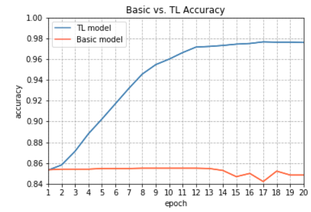
(d) Loss of TL model over fft domain data

Fig. 3: TL accuracy and loss over time and frequency domain IQ data

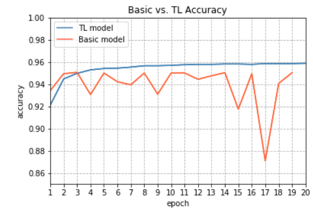
We have also compared the accuracy of basic CNN model and TL model classifying both the time and frequency domain data by retrieving weight of the basic CNN model and try running 20 epochs each, the results are shown in Fig. 4(a) and Fig. 4(b) respectively. In the time domain scenario, the accuracy of basic scenario is relatively stable in the picture, as it only fluctuates in a very small range of accuracy. In contrary, in the frequency domain scenario, the fluctuation is relatively remarkable, getting 95.06% as peak performance, but TL model still maintained a superior performance than the basic CNN model did in overall. Table II shows the accuracy of with and without applying TL to the CNN model classification.

TABLE II: Accuracy of models with and without TL

Model	Time Domain Data	Frequency Domain Data
TL model	97.65%	95.87%
Basic CNN model	85.51%	95.06%



(a) Accuracy over time domain data



(b) Accuracy over frequency domain data

Fig. 4: Comparison of TL and basic CNN model accuracy

IV. Conclusion

In this paper, we tried to enhance the WII approach by applying TL upon a regular neural network learning basics and improved the signal classification accuracy. The accuracy we achieved by using TL is more than 12% higher than the basic model. We achieved a relatively good result in time domain IQ signals however, there are more scope to improve the performance. Through our experiment, the validity of TL concept was successfully analyzed.

References

[1] Q. Pang and V. C. M. Leung, "Channel Clustering and Probabilistic Channel Visiting Techniques for WLAN Interference Mitigation in Bluetooth Devices," in IEEE TRANSACTIONS ON ELECTROMAGNETIC COMPATIBILITY, BOL. 49, NO. 4, NOVEMBER 2007.

[2] W. Lee and H. Kim, "Channel Quality Estimation for Improving Awareness of Communication Situation in the 2.4 GHz ISM Band," IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 17, NO. 9, SEPTEMBER 2018.

[3] W. Sun, J. Koo, S. Byeon, W. Park, S. Lim, D. Ban, and S. Choi, "BlueCoDE: Bluetooth Coordination in Dense Environment for Better Coexistence," in IEEE International Conference on Network Protocols (ICNP), Toronto. ON. Canada, 10-13 OCT. 2017.

[4] Q. Zhang, G. Chen, L. ZHao, and C. Chang, "Piconet Construction and Restructuring Mechanisms for Interference Avoiding in Bluetooth PANs," in Journal of Network and Computer Applications, Volume 75, November 2016, Pages 89-100.

[5] M. Schmidt, D. Block, U. Meier, "Wireless Interference Identification with Convolutional Neural Networks," Emden. Germany, INDIN, 24-26 July 2017.

[6] X. Zhang, T. Seyfi, S. Ju, S. Ramjee, A. E. Gamal, and Y. Eldar, "Deep Learning for Interference Identification: Band, Training SNR, and Sample Selection," Cannes. France, SPAWC, July 2019.

[7] X. Wang, S. Ju, X. Zhang, S. Ramjee, and A. E. Gamal, "Efficient Training of Deep Classifiers for Wireless Source Identification using Test SNR Estimates,", 26 Dec 2019.

[8] Dipanjan (DJ) Sarkar, https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a/, Nov 15, 2018

[9] Hoffbeck, Joseph P. and Andrew Melton. "Rf Signal Database For A Communication Systems Course." (2006).