International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.14, No.1 (Aug-2023)

http://dx.doi.org/10.12785/ijcds/140131

A Segmentation based Classification Model for Primary User Detection Using Deep Learning Techniques

K. Muthumeenakshi[1]

[1]*Department of ECE, Sri Sivasubramniya Nadar College of Engineering, Chennai, India*

Received 01 Jul. 2022, Revised 07 May. 2023, Accepted 12 May. 2023, Published 01 Aug. 2023

Abstract: Designing wireless communication systems with an aim to improve the utilization of the existing frequency bands involves the novel idea of cognitive radio (CR). CR allows sharing of licensed spectral band of primary users (PU) with the secondary users (SU) if and only if the licensed user is not subject to harmful interference. To sense the availability of PU spectrum, spectrum sensing is employed as a primary task of CR. Traditional signal processing techniques for spectrum sensing have the problem of false alarm or missed detection and may cause interference to PU. Thus, to further expand the ability of learning for CRs and to support an efficient PU sensing, artificial intelligence, machine learning or deep learning techniques can be applied. This paper proposes an efficient and well performing segmentation cum classification algorithm based on deep learning techniques for PU sensing. The spectrogram of the PU's transmission signal pattern for different scenarios was classified using Res-Net 50 model. To further improve the accuracy, a region proposal-based Res-Net50 model is proposed. The performance evaluations validate the effectiveness of the proposed model.

Keywords: Cognitive Radio, Deep Learning, Spectrum Sensing, Classification, Segmentation.

1. INTRODUCTION

CR, a novel technology addresses the problems in wireless technology and computational intelligence [1]. Radio devices capable of learning and adaptation to changes around its radio surroundings is referred to as CRs. It is observed that due to the massive growth in the areas of wireless networks and communication, the number of nodes and devices used has increased to a larger extend. Due to this increase in the usage of devices, scarcity issues in the available spectrum have become predominant. The problem of scarcity in spectrum can be mitigated with the use of CRs which helps to improve the utilization of the licensed spectrum, by providing the SUs dynamic spectrum access [2]. To accomplish all these cognitive tasks, the CR should detect all types of RF activities by being aware of its RF surroundings. This requires periodic spectrum sensing by CR.

CRs sense and learn about its RF surroundings through spectrum sensing technique. Through spectrum sensing, CR discovers opportunities for access to free PU spectrum and allows multiple CRs to share the PU spectrum provided that the latter is not subjected to harmful interference. Commonly employed spectrum sensing methods include matched filter technique, energy detection technique, cyclostationary based feature detection technique, etc., These efficacies of these techniques depend on the level of noise which may result in high false alarm or missed detection. Hence, this work involves applying machine/deep learning models to spectrum sensing problem, which can further improve the correctness of spectrum sensing. The proposed system model is based on residual deep learning that builds a classifier model to classify the PU's communication pattern into ten classes. The algorithm uses a segmentation cum classification model trained on spectrogram images to detect the nature of spectrum occupancy and classifying them into 10 unique classes. It would be useful for SU if the trained model to be deployed in a dynamic environment, to detect the spectrum occupancy.

2. LITERATURE REVIEW

Machine/deep learning techniques and CR are two vast domains comprising a lot of literature useful for developing this work. CRs sense the spectrum occupation pattern of PU and use its spectrum when it is detected vacant as long as there is no significant interference. Existing spectrum detection techniques, although effective and robust, can be substituted with machine/deep learning models for better classification. With various learning techniques available for sensing the spectrum, an efficient technique and robust technique must be chosen. Some of the works of literature that were found helpful for developing the proposed work are discussed below.

A comprehensive literature survey for the use of machine/deep learning algorithms for CR classification with

E-mail address: muthumeenakshik@ssn.edu.in

their merits and demerits is discussed in [3]. The authors also point out some challenging learning problems in CRs with possible solutions. In [4], the detection of cognitive radio spectrum occupancy is studied using machine learning techniques like linear regression, naïve Bayes algorithm, support vector machine (SVM), decision tree algorithm and hidden Markov models. Finally, a new Firefly based SVM algorithm is proposed which attains the best classification accuracy. A cooperative spectrum sensing using supervised machine learning approaches (weighted K nearest neighbor and SVM) and unsupervised machine learning techniques (Gaussian mixture model and K-means clustering) is proposed in [5]. SVM based classifier is found to outperform other machine learning techniques. Another new and improved cooperative spectrum sensing using SVM is proposed in [6] with reduced overhead and improved detection performance. In [7], a deep cooperative sensing model with small sized CNN is introduced which takes real or binary values individual sensing outcomes of SUs. A learning model with CNN - long short term memory model is proposed in [8] The model is proved to perform well irrespective of the noise model assumptions.

In [9], the authors introduced an alternative approach for radio signal detection and localization using spectrogram data trained with CNN for bounding box regression. Their approach is found to outshine performance of human level by computer vision for object detection techniques, similar ones have been used for radio applications. It is done by the basic with labeled training data with broadband spectrogram image annotated with bounding boxes and masks. Authors reported an increased performance in sensitivity when compared to conventional energy detection based sensing methods.

The authors of [10] investigate algorithms for multiple signal detection with classification of modulation schemes using a deep learning framework. Details such as the signal modulation format, start-stop time and carrier frequency can be obtained from their technique. A single shot multibox detection scheme for signal detection and multi-input CNNs for modulation recognition is built which achieves better performance than the existing methods.

In [11], yet another perspective on classification using spectrograms is provided. The work in this literature is to provide the licensed access of the 3.5GHz radar spectrum to the commercially available users in the United States. The rules are that the commercial systems should leave the spectrum when the sensors identify the radars operated by U.S. military. This work used 14,000 spectrograms over which they investigated 13 different methods for detection. They proved that machine learning algorithms outperformed the traditional signal detection algorithms.

The authors of the research work proposed in [12] developed a best-performing technique that merge several low-level features and high-level image context for object detection applications. They combined region proposals with CNN, referred to as R-CNN, regions with CNN features that achieved 30% more performance than the previous techniques.

A radio access technology classification problem for opportunistic spectrum access using machine learning techniques is presented in [13]. They use a public spectrograms dataset for training and developed a prototype to create data for testing and reported an overall accuracy of 96%.

The dataset used in this paper is obtained from [14] and [15] which contain the spectrogram data that illustrate the PU's scenario into various classes. Spectrograms show the time frequency variation of the signal graphically. These spectrograms, which are 64x64 gray scale images are used as inputs to the proposed deep learning model.

In this paper, a simple and scalable PU algorithm for detection is proposed with the following primary objective:

- To develop a classifier cum segmentation model that can classify the PU transmission patterns and detect the presence channel occupancy in the spectrum using a suitable semantic segmentation algorithm.

The remaining segments of the paper are structured as follows: Section 3 provide the dataset description. Section 4 explains the model used for PU sensing comprehensively. Section 5 describes the model's performance and analysis in detail. Section 6 provides the conclusion and discussions.

3. PU DATASET DESCRIPTION

The realistic PU spectrum occupancy with unique values of packet duration, hopping pattern, inter packet delay, spectrum occupancy and bandwidth is captured and reflected as spectrograms. Each spectrogram is a 64x64 gray scale image which cover 4 channels for 50 msec occupying 10 MHz bandwidth (4 x 2.5MHz channels). The dataset obtained for 10 mutually exclusive classes each class and the details of each class are described in Table I.

There are 71,200 training spectrograms and 17800 testing spectrograms from a total collection of 89000 spectrograms. Each class of the dataset is marked according to its corresponding occupancy in the spectrum. Class 0 to class 4 show absence of PU in some channels, while classes 5 to 9 show PU presence in all channels. Sample spectrogram images from each class for varying SNR values are shown in Figure. 1.

4. SYSTEM MODEL

System model for PU classification is developed using a region proposal based Resnet-50 deep learning framework. First, the use of deep residual model Resnet-50 for PU detection is explained. The performance of the model is discussed in the next section and compared with a simple 2-layer CNN. To further improve the accuracy of the model, segmentation cum classification model is proposed. The

TABLE I. Dataset Description

Class	Description	Inter-packet delay	Training dataset	Testing dataset
Class 0	Single channel (random)	5 ms	8000	2000
Class 1	Single channel (random)	10 ms	8000	2000
Class 2	Two random channel hopping	5 ms	18000	4500
Class 3	Four random channels hopping	10 ms	3200	800
Class 4	Four channels synchronous	5 ms	18000	4500
Class 5	Two random channel synchronous	5 ms	3200	800
Class 6	Four channels synchronous	2 ms	3200	800
Class 7	Four channels (delays - Poisson distributed)	20 ms (mean)	3200	800
Class 8	Four channels (delays - Poisson distributed)	10 ms (mean)	3200	800
Class 9	Four channels (delays - Poisson distributed)	5 ms (mean)	3200	800

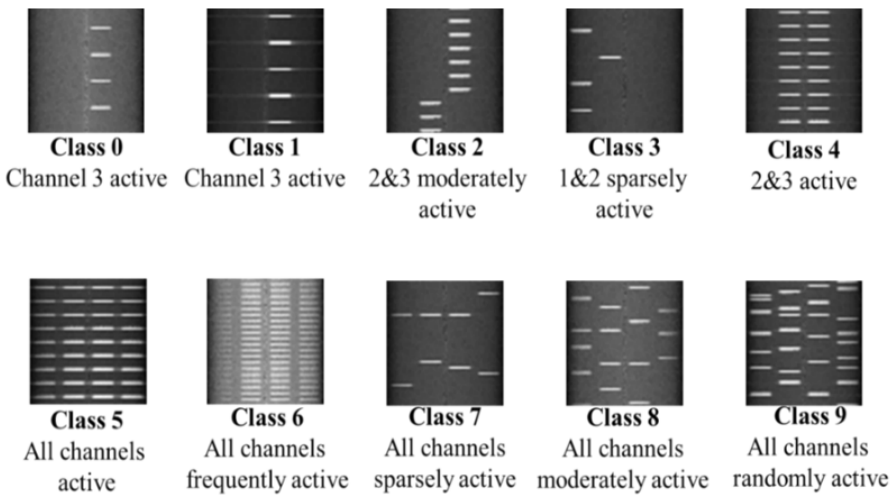


Figure 1. Sample Spectrogram Images for all PU classes at different SNR values

detailed description which is explained whose performance is compared is resnet-50 model.

A. *Deep Residual Model for Spectrum Sensing*

Deep neural networks generally have more training complexities but have shown improved performance for image classification problems. Owing to the complexity involved and issues like vanishing gradient, it usually takes more time to train deeper neural networks and takes a lot of computational resources. ResNets are deep residual networks that are well demonstrated as a successful model for image classification tasks. ResNets makes the training process faster and easier. In addition, they also achieve improved accuracy compared to other neural networks.

Figure. 2 describes the Resnet-50 system model used for PU occupancy detection by secondary users. The input image is fed to the convolution layers of the model and becomes interpolated to appropriate image size. The images are considered as matrices or vectors of the pixel. The model learns by passing the images through each and every layer, and an optimized weight vector is finally obtained. The final convolution layer Conv 5 x includes four 512 layers and four 2048 layers. The output from the final layer is flattened to 1x8192 vector. Then the dense fully connected layer having one thousand nodes and ReLU function for non linear activation is followed. A final output node has multi-class output with a SoftMax activation function. A 1x10 label vector is generated having probabilities of multiple target classes. This vector of appropriate probabilities is used for classification when the test input image is provided as input.

B. *Segmentation Cum Classification Model*

To further enhance the accuracy of detection, segmentation cum classification model is proposed. Deep learning-based model called DeepLab is the model used for image segmentation. One of the key advantages of using DeepLab is that it is compatible with the image classification model, ResNet-50. In other words, the segmentation model can use ResNet-50's weights, thereby not impacting on the computation complexity and robustness. Even though, DeepLab uses pre-trained weights it is not trained on a pixel level. For that purpose, the segmentation model has to train only on the region proposals of a particular image. For training

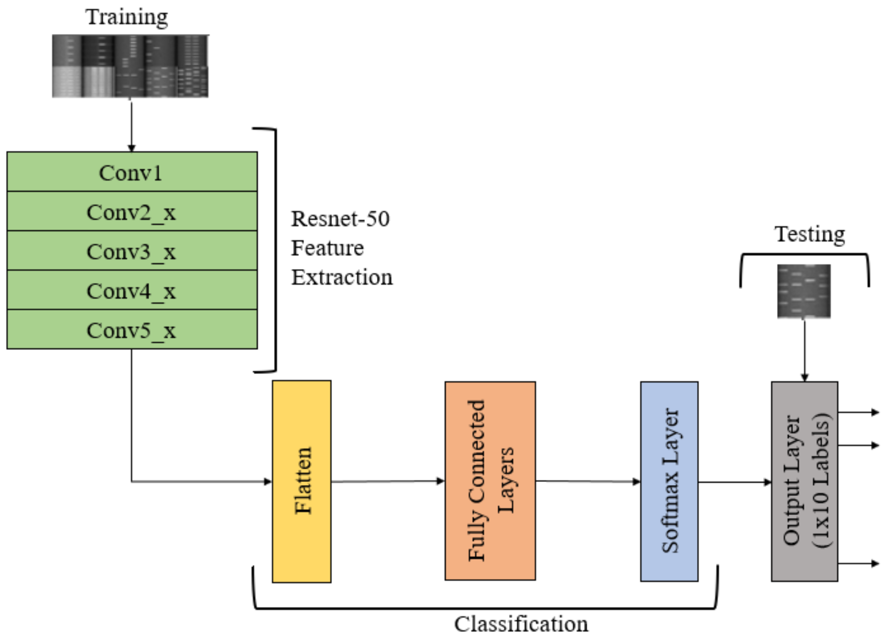


Figure 2. Resnet-50 Model for Spectrum Sensing

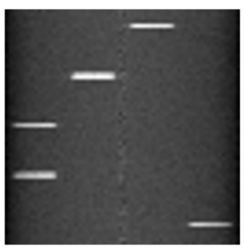


Figure 3. Original Image

Figure 4. Masked Image

Figure 5. Pre-processed Image

the DeepLab model on our spectrogram dataset, the data has to be converted to TFR Record format, which is a simple XML format consisting of information about a particular image, its mask, its region of interest in a simpler manner such that the model understands it easily.

C. Creating Masked Images

The masked image of a particular image is a replicated binary image containing white pixels only in the regions of interests. The image masks are created so that the model learns the region proposals quickly. The image masks are created using software called pixel annotation tool. This software is used to annotate images manually and quickly in directories. The method is pseudo manual because it uses the watershed algorithm of OpenCV.

In Figure 3-5, a sample original image available in the dataset, the masked image of the sample and on the rightmost the pre-processed image is shown. Each class images in the dataset are separately annotated using the pixel annotation tool. Each image is initially masked for the spectrum occupancy and then the masks are transferred to other similar images in the same class. The output from a single image generates a colored mask, a watershed mask, and a regular mask. The mask images are applied with a pre-processing step called Otsu's binarization threshold which is a histogram-based thresholding function from python OpenCV2 These images are then given as input for the training of DeepLab model. With this step, the prerequisites of training a DeepLab model are satisfied. To obtain more accurate results DeepLab works in two phases. The first stage is to extract the region of interest (RoI) or RoIs in the plural form. An RoI is an area of the input image that may contain an object. The second stage is classification, where each RoI is resized to fit the convolutional neural network as discussed in [16].

D. Region Proposals

Regions of interest are generated through the region proposal network (RPN). To generate RoIs, the RPN uses convolutional layers. The RPN accepts an input image and output the RoIs. Every RoI encloses a bounding box and a class score probability value. The CNN is used to extract a feature volume to generate those numbers. The feature volume is then used to generate the regions, coordinates and probabilities. The architecture of the RPN model is shown in Figure 6. The architecture of the RPN and its step-by-step process shown in explained below:

- Step-1: The RoIs are generated in the original image and are converted into the feature map coordinate system.
- Step-2: Resize each RoI such that it fits the input of the fully connected layers.
- Step-3: Apply the fully connected layer. The map is flattened to obtain a feature vector.
- Step-4: Apply two different convolutional layers. One handles the classification (called cls), and the other handles the refinement of the RoI (called rgs).

The final output is a list of RoIs after post-processing. At this step, no information about the class of the object is generated, only about its location. During the next step, classification, we will classify the objects and refine the bounding boxes.

E. Classification

The second part of the DeepLab model is classification. The model accepts two inputs, RoI list from RPN. The feature volume is computed from the input image and the final bounding boxes are obtained. The classification part can work with any feature volume corresponding to the input image. However, as feature vector weights have already been computed in the previous classification model (ResNet-50), they are simply reused here This technique has two benefits sharing the weights, sharing the computation.

The Deep Learning model used for spectrum sensing is shown in figure 7. The figure shows how the region proposals from RPN are being used in the ResNet-50 model. Thus, making it into a segmentation model similar to DeepLab. The feature extraction and the classification part of this model are briefly explained in section 4. The scope of this section is to discuss how a simple classification model can act as a segmentation model.

As described in 4.1.1 the output from RPN are class scores and the bounding box locations, it also says that the refinements of ROIs are handled by a regressor. The proposed model, in a similar manner, has a regressor for refining the bounding box locations. The location of a single bounding box is given by [x, y, w, h], where x and y are the coordinate points of the left top corner of the bounding box, whose height and width are given by y+h and x+w respectively. Since the proposed RCNN model uses a classifier and a regressor after the full connection layers, it acts similarly to the DeepLab model and produces segmentation outputs. The advantage of doing so is it reduces the computation complexity of the model, thereby concentrating more on accuracy and robustness.

F. Training the Model

Because of its unique architecture, the proposed model cannot be trained as a regular CNN. If each of the two parts of the network were trained separately, the feature extractors of each part would not share the same weights. Therefore, a new training regimen for this model was developed, it is discussed below:

- Step-1: Train the RPN so that it predicts acceptable RoIs.
- Step-2: Train the classification part using the output of the trained RPN. At the end of the training, the RPN and the classification part have different convolutional weights since they have been trained separately.
- Step-3: Replace the RPN's CNN with the classification's part CNN so they now share convolutional weights. Freeze the shared CNN weights. Train the RPN's last layers again.
- Step-4: Train the classification's last layer using the output of the RPN again.

At the end of this process, a trained network is obtained with the two parts sharing the convolutional weights. The training of this particular model is carried on 20,000 spectrograms each class containing 2000 images and is tested using 5,000 spectrograms. A separate set of 1000 images not appearing in both training and testing sets is used for validation. The model trained for 25 epochs, for a batch size of 32 with a step size of 625. The proposed ResNet-50 based segmentation model runs on Nvidia GTX 1050 graphics processing unit (GPU) card and implemented on the open-source TensorFlow object detection API.

5. PERFORMANCE EVALUATIONS

The performance of a deep learning model will be based on how well the system predicts the given test image. The performance metrics such as training accuracy, validation accuracy, confusion matrix, top-one accuracy, top-five accuracy, are generally used for analysis. For every single epoch, training is done for a batch size of optimal

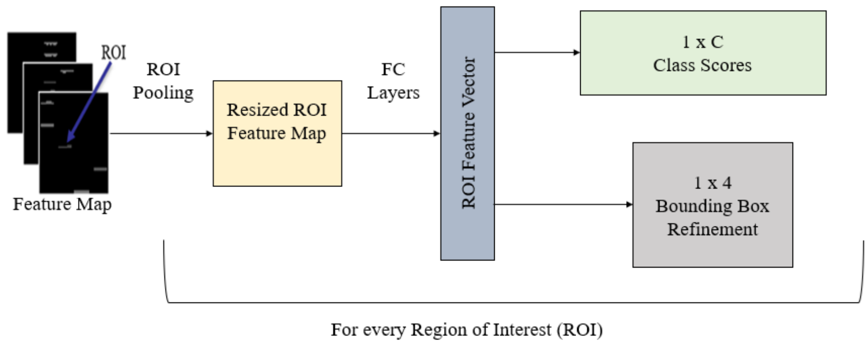


Figure 6. Architecture of the Region Proposal Network

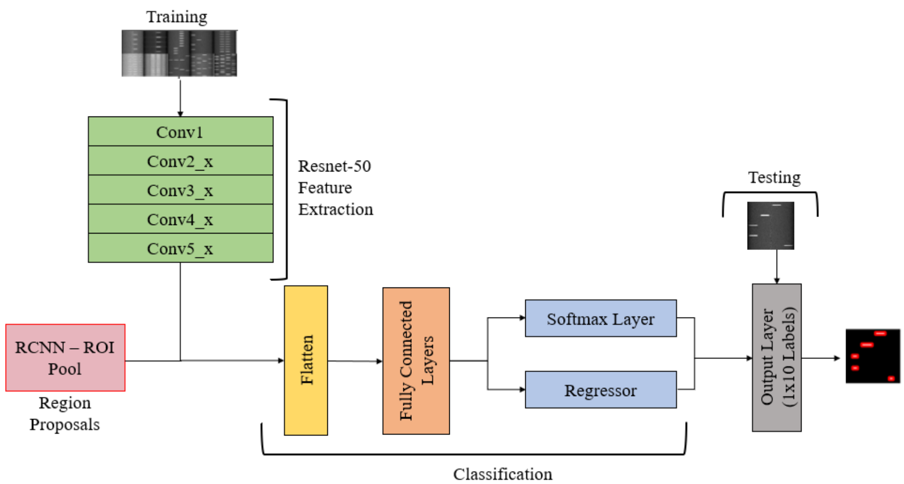


Figure 7. Deep Learning Model for PU Spectrum Sensing

value. At the end of each epoch, the accuracy of training and testing are obtained. The model is trained using 20,000 spectrograms and is validated on 5,000 spectrograms for about 15 epochs. The results obtained are shown in Table II in which a comparison between 2-Layer CNN and ResNet-50 is carried out.

There are two types of performance metrics available for appraising the predicted classes in multi-class categorical classification. Top-1 Accuracy From a set of label vectors obtained the highest probability is chosen and it is predicted as the label. Top-5 Accuracy From a set of label vectors obtained 5 maximum probability values are chosen and predicted as output in increasing order of probabilities.

A confusion matrix of a set of training data is a performance indicator of any classification model which illustrates the correct and incorrect predictions. It gives a summary of prediction results for known true values as in the training data for the classification problem. The correct and incorrect predictions as represented by the confusion matrix with count values for each class is shown in Figure 8.

From the confusion matrix, the important point that is observed is that the prediction is perfect for class 6 and classes 3 and 9 has relatively weak prediction performance i.e. more of those classes spectrograms were incorrectly labelled as other classes. Loss function, in general quantifies how well a model performs in prediction of the classified output The model is said to be performing better in predicting the classes correctly if the loss function is minimum. The main parts that defines the losses in a model are training and validation. The difference between them is that training loss is obtained during the epochs whereas the validation loss is obtained after the epoch. The training loss is continually reported over the duration of the entire epoch; the validation metrics are computed for the validation set only when the current training epoch is over. It is seen that the gaps are much smaller between the training and loss values. Figure 9. shows the shifted training and validation losses graph across the epochs.

A good fit for a classification model can be identified by

TABLE II. Comparison of Resnet-50 and 2 Layer CNN

Performance	Resnet-50 model	2 Layer CNN Model
Training Accuracy (%)	82.24	71.45
Validation Accuracy (%)	86.07	72.32
Testing Accuracy (%)	92.24	74.32

TABLE III. Comparison of Resnet-50 and Proposed RCNN based Resnet-50

Performance	Resnet-50 model	RCNN based Resnet-50
Training Accuracy (%)	82.24	81.81
Validation Accuracy (%)	86.07	87.80
Testing Accuracy (%)	92.24	93.12

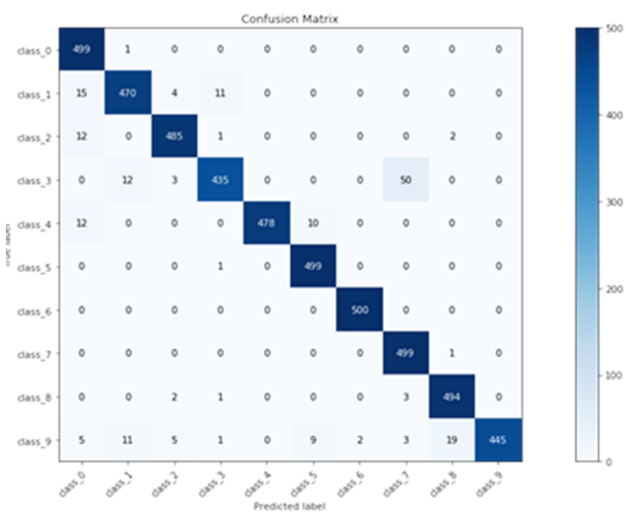


Figure 8. Confusion Matrix of the classifier model

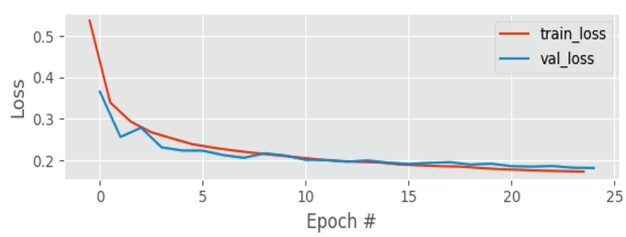


Figure 9. Training and validation loss

functions such as training and validation loss. As shown in Figure 7, it can be inferred that the validation loss is lesser than the train loss, which means that the model does not overfit or underfit.

Table III summarizes the accuracy metrics of the ResNet-50 and RCNN based ResNet-50 models. It is seen that training accuracy in the RCNN based model is affected due to the increased strain on the segmentation part. The following sections will discuss about some metrics on the segmentation part of the model.

A. *Performance of the Segmentation Model*

Because of its unique architecture, the proposed model cannot be trained as a regular CNN. If each of the two parts of the network were trained separately, the feature extractors of each part would not share the same weights. Therefore, a new training regimen for this model was developed, it is discussed below:

The semantic segmentation is a task which is used to foresee the class of each pixel of a given image. The prediction output should match the input's spatial resolution (height and width) with a channel extent equal to the total number of classes to be predicted. Each channel consists of a binary mask which labels areas where a specific class is available. Segmentation-centric metrics like pixel accuracy, mIoU (Jaccard Index) and Dice coefficient (F1 score) are discussed.

Pixel Accuracy: It is the percentage of pixels in an image that are correctly classified. The initial segmentation accuracy obtained was 93%. The issue with this metric is that, only for one class, the pixel accuracy was 93% of the original image. So, if the model classifies all pixels as that class, 93% of pixels are accurately classified and 7% are not correctly classified. This leads to an issue referred to as class imbalance. This happens because the spectrogram dataset contains only a few regions having spectrum occupancy. This issue can be solved by not taking the black pixels into considerations, which provided a very low value, i.e., 42% accuracy. Hence, much better metrics like mIoU and Dice coefficient are evaluated.

Mean Intersection Over Union (IoU): In semantic segmentation, the IoU otherwise referred to as theJaccard Index is one of the most popularly used performance metrics. IoU is defined as ratio of the common area overlapping between the predicted segmentation and the ground truth image to the area of union between the predicted and the ground truth. This range of values for this metric is from 0 to 1 (0 to 100%) with 0 meaning no overlap and 1 meaning fully overlapping segmentation. For multi-class segmentation, the average or mean IoU is considered which is calculated by

TABLE IV. Performance of Segmentation Model

Merics	Values	Time(sec)
Mean IoU	0.834	204
Dice Coefficient	0.822	202

averaging the IoU values of each class. It is defined as follows in equation (1)

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| + |A \cap B|}$$

$|A|$ and $|B|$ are the cardinality of each set; that is, the number of elements they each contain. $|A \cap B|$ is the intersection of the two sets, and therefore the numerator $|A \cap B|$ represents the number of elements they have in common. Similarly, $|A \cup B|$ is the union of the sets, and therefore the denominator $|A \cup B|$ represents the total number elements the two sets cover together.

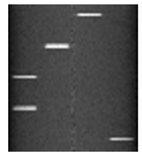
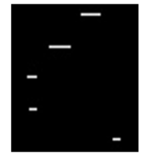
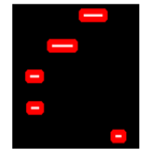
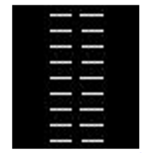
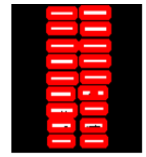
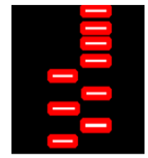
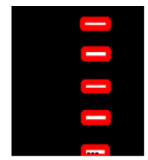
Mean IoU is evaluated using TensorFlow which uses Keras as a backend to run the IoU part of the metric and then it is averaged by the frontend. The model performed well and obtained mean IoU of 0.834 in 204 secs. This metric proved more efficient than pixel accuracy and did not pose any class imbalance issues.

Dice Coefficient: The Sorensen–Dice coefficient measures how well two sets overlap. The Dice coefficient is a metric that is similar mean IoU. They are positively correlated, i.e., if model X is found to be better performing than model Y at segmenting an image, then it will be the same for the other. The value of Dice coefficient ranges from 0 to 1, with 1 signifying the maximum similarity between predicted segmentation and ground truth. It is expressed in the equation (2) as shown below.

$$Dice(A, B) = \frac{2|A \cap B|}{|A \cup B|} = \frac{2|A \cap B|}{|A| + |B| + |A \cap B|}$$

In semantic segmentation, Dice is, therefore, used to measure how well the predicted mask for each class overlap the ground-truth mask. For one class, the numerator then represents the number of perfectly classified pixels, and the denominator represents the total number of pixels belonging to this class in both the predicted and ground-truth masks. As a metric, the Dice coefficient thus does not depend on the relative number of pixels one class takes in images and thus it is not affected by the class imbalance issue. Table V shows the mean IoU and Dice coefficient of the segmentation model.

TABLE V. Predicted Output for RCNN based Resnet-50

Class Name	Input Sample	Ground Truth	Predicted Output
Class 0			
Class 2			
Class 3			
Class 6			

B. *Prediction Results*

The prediction results of a model is a direct comparison of predicted outputs with their corresponding ground truth and original input. The predicted output produces an image similar to the input image with bounding boxes around the spectrum occupied region. Table V provides the details of predicted outputs RCNN based ResNet-50. The table depicts all the best performing classes, provided a few poorly performing classes were present which did not affect the overall performance of the model.

6. Conclusions

Spectrum sensing is a primary key functionality of CR whose accuracy is very crucial. Traditional spectrum sensing techniques has some limitations and constraints, so machine learning techniques were proposed to predict the occupancy of PUs. In this paper, a ResNet-50 model based on residual learning was proposed to predict the transmission patterns of PUs. The prediction results show that the proposed ResNet-50 model was more accurate and robust in classifying the PUs transmission classes. In addition, it is observed that the ResNet-50 model correctly classified the more related classes into their proper individual classes. The proposed classifier cum segmentation model, RCNN based ResNet-50 was tested and validated on the same dataset. The model was able to precisely detect as well as classify the spectrogram for the spectrum occupancy. Various metrics were also used to appraise the model.

References

[1] I. Mitola, Joseph, "Cognitive radio," *PhD Dissertation, Institutionen för teleinformatik*.

[2] I. e. a. Akyildiz, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, pp. 2127–2159, 2006.

[3] M. Bkassiny, Y. Li, and S. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys Tutorials*, vol. 15, pp. 1136–1159, 2013.

[4] F. Azmat and S. N, "Analysis of spectrum occupancy using machine learning algorithms," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 6853–6860, 2016.

[5] K. Thilina, K. Choi, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 2209–2221, 2018.

[6] Z. Li, W. Wu, X. Liu, and P. Qi, "Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks," *IET Communications Journal*, vol. 31, pp. 2485–2492, 2018.

[7] L. Woongsup, K. Minhoe, and C. Dong-Ho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 3005–3009, 2019.

[8] X. Jiandong, L. Jun, Fang, and L. Xuanheng, "Deep learning-based spectrum sensing in cognitiveradio: A cnn-lstm approach," *IEEE Communication Letters*, vol. 68, pp. 2196–2200, 2020.

[9] T. O'Shea, T. Roy, and T. Clancy, "Learning robust general radio signal detection using computer vision methods," *Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 829–832, 2017.

[10] X. Zha, H. Peng, X. Qin, G. Li, and S. Yang, "A deep learning framework for signal detection and modulation classification," *Proceedings of the IEEE Conference on SENSORS*, vol. 19, pp. 4042–4048, 2017.

[11] W. Max, Lees and D. H. Paul, "Deep learning classification of 3.5-ghz band spectrograms with applications to spectrum sensing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, pp. 224–236, 2019.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–58, 2014.

[13] F. Erik, F. S. Joao, and L. A. D. Francisc, Paisana, "Radio access technology characterisation through object detectionradio access technology characterisation through object detection," *Computer Communications*, vol. 168, pp. 12–19, 2021.

[14] P. Francisco, S. Ahmed, K. Maicon, A. Pedro, T. Justin, B. Christian, P. Andre, and D. Luiz, "Context-aware cognitive radio using deep learning," *International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–12, 2017.

[15] W. Felix, P. Francisco, S. Rajendran, S. Ahmed, A. Pedro, M. Sebastian, K. Sebastian, V. d. B. Bertold, and P. Sofie, "Dyspan spectrum challenge: Situationalawareness and opportunistic spectrumaccess benchmarked," *IEEE Transactions On Cognitive Communications And Networking*, vol. 3, pp. 550–562, 2017.

[16] J. Vikneshwar, K. Muthumeenakshi, and S. Radha, "Classification of primary users using deep residual learning," *Advances in Automation, Signal Processing, Instrumentation, and Control, Lecture Notes in Electrical Engineering*, vol. 700, 2021.

K. Muthumeenakshi serves as an Associate Professor in the Department of Electronics and Communication Engineering, Sri Sivasubramaniya Nadar College of Engineering. She received her B.E. in Electronics and Communication Engineering in 1999, M.E. in Applied Electronics in 2004 and Ph.D. in 2015. Her research interests include works on Signal processing, Spectrum sensing and dynamic spectrum access issues in cognitive radios and RF Energy Harvesting techniques.