A Deep Convolutional Neural Network Based Transfer Learning Method for Non-Cooperative Spectrum Sensing

BIPUN MAN PATI, MEGUMI KANEKO, (Senior Member, IEEE), AND ATTAPHONGSE TAPARUGSSANAGORN

1Asian Institute of Technology, Pathum Thani 12120, Thailand
2National Institute of Informatics, Tokyo 101-8430, Japan

Corresponding author: Bipun Man Pati (bemaanpati@gmail.com)

ABSTRACT In this article, we investigate machine learning methods for enabling high performance non-cooperative spectrum sensing, for future cognitive radio systems. The fulfillment of sensing requirements is crucial for ensuring an efficient reuse of the scarce spectrum by unlicensed users, without causing harmful interference to primary users. In this work, we propose a deep convolutional neural network-based transfer learning framework for non-cooperative spectrum sensing in TV bands, applicable across various locations, wireless environments and even frequency assignments. Specifically, we design a four-layer convolutional neural network for limiting the computational costs while satisfying the sensing requirements, and apply transfer learning by freezing the first two convolutional layers. The performance of the proposed method is evaluated against benchmarks, based on over 29,000 spectrograms collected in UHF TV band from a recent measurement campaign. The experiments show that thanks to transfer learning, the proposed method is able to detect TV signals with high accuracy despite a significantly reduced amount of data, thereby providing a high adaptability to various locations, environments, and frequencies. Furthermore, the proposed method with transfer learning not only guarantees the sensing requirements but also realizes up to 94% reduction of training time of the network, as well as 20% reduction of the required sensing time, compared to the case without transfer learning.

INDEX TERMS Spectrum sensing, cognitive radio, deep learning, transfer learning.

I. INTRODUCTION

Future wireless communication systems are facing severe technological issues due to the rapid increase of mobile data traffic despite the lack of spectrum resources. During the past years, Cognitive Radio (CR) [1] has been extensively investigated as a solution for alleviating the problem of spectrum underutilization caused by the static allocation of RF spectrum, by means of spectrum sharing among the licensed Primary Users (PU) and the unlicensed Secondary Users (SU). The spectrum sharing techniques can be classified as underlay, overlay, and interweave models [2]. In the underlay model, the SU’s transmission on the PU’s band is allowed under power constraints to avoid interference to PUs. In the overlay model, SUs are required to have full knowledge of PU signals so as to avoid any interference, whereas in the interweave model, SUs can make use of frequency bands that are not being used by PUs. For this purpose, the SU is required to perform spectrum sensing to determine whether a particular band of frequency is being used by the PU or not. Conventional spectrum sensing methods include energy detection, cyclostationary detection, matched filter detection [3], and covariance-based detection [4].

Such CR capabilities have been integrated into the standard for Wireless Regional Area Network (WRAN) defined by IEEE 802.22 [5], with the aim to carry broadband access to hard-to-reach and sparsely populated rural areas. It is a point-to-multipoint network topology that operates in the TV band for taking advantage of the favorable radio propagation characteristics of lower frequencies while avoiding interference towards PUs. In addition to this, the IEEE 802.11af [6] standard allows Wireless Local Area Network (WLAN) to operate in the TV White Space (TVWS).

Recently, machine learning-based techniques have been leveraged for spectrum sensing, as they are capable of implicitly and efficiently learning the surrounding wireless
environment. Furthermore, these techniques are shown to be more adaptive to the dynamic changes of the environment, as compared to conventional methods [7]. So far, most works on spectrum sensing using machine learning have focused on manual feature extraction. For instance, authors in [8] and [9] propose an Artificial Neural Network (ANN) and a Convolutional Neural Network (CNN) for spectrum sensing respectively, using features based on energy and cyclostationarity of the received signal. ANN-based spectrum sensing using classical energy detection and likelihood ratio statistics is proposed in [10]. One important drawback of such manual feature extraction is that the whole network would need to be retrained by collecting a whole new set of data whenever the input signal feature changes. Automatic feature extraction has also been investigated in the literature for spectrum sensing by means of deep learning [11], [12]. However, the trained model obtained in these works is highly dependent on the specific location and environment where the data was collected. Hence, the Deep Neural Network (DNN) model needs to be retrained each time the location and environment undergo changes. This requires new data sets to be collected in each of the new locations and wireless environments. As training requires considerable amounts of time, data, and computational resources it is impractical to collect new data sets and to retrain the model every time such changes occur.

Therefore, in this work, we consider the spectrum sensing problem in the interweave scenario, whereby a single SU performs sensing through deep learning with automatic feature extraction. The goal of our proposed method is to alleviate the problem of collecting new data sets and to retrain the DNN whenever the input signal features undergo changes. Firstly, we design a CNN architecture with automatic feature extraction, that enables us to predict the frequency bands unoccupied by PUs within the TV band of interest. The proposed CNN is cost-efficient and well-suited for a variety of input signals, namely the different spectrograms spanning different locations, Signal-to-Noise Ratio (SNR) environments, and frequency bands. The proposed CNN method is further enhanced by means of transfer learning, that enables a swift adaptation of the proposed neural network to location, environment, and frequency changes, namely with low computational complexity as well as minimal additional sensing time. In particular, the complexity reduction achieved through transfer learning will entail less training time when location and environment are changed, while the reduction in sensing time will directly increase the data transmission time, thereby improving the overall throughput of the system.

The main contributions of this article are summarized as follows:

1) We propose an interweave spectrum sensing method based on a CNN architecture for extracting the signal features for detecting the PU signals, whereby the proposed CNN is designed to handle a variety of spectrograms over different locations, SNR environments, and frequency bands.

2) To enable high quality predictions across different environments while minimizing the required amount of training data, computational complexity and sensing time, transfer learning is exploited in the proposed CNN. To the best of the authors’ knowledge, this is the first work exploiting transfer learning to enable spectrum sensing across different locations, SNR environments, and frequency bands.

3) The proposed method is assessed through an experimental setup, whereby real-world data sets obtained by on-site measurements have been gathered. The experimental results demonstrate the efficacy of the proposal in terms of prediction performance and show that training and sensing times can be effectively reduced compared to benchmark schemes.

The remainder of the paper is organized as follows: Section II provides the related work. Section III then describes the system model of the study. The reference approaches and proposed methods are discussed in Section IV and V, respectively. Section VI presents the details of the measurement setups, locations, and the obtained results. Finally, Section VII concludes this work and gives directions for future research.

II. RELATED WORK

It has almost been a decade since the Federal Communications Commission (FCC) has allowed the use of an unused broadcast TV spectrum for secondary usage [13]. Many measurement campaigns have been carried out since then. However, most of the measurement campaigns to study the spectrum occupancy in the Ultra High Frequency (UHF) frequency band allocated to TV broadcasting, had been carried out in developed countries, using highly specialized equipment [14]–[18]. The introduction of the low-cost spectrum analyzer, i.e., RFExplorer [19] has enabled carrying out spectrum measurements in an affordable, portable, and easy to use manner. Since the introduction of such a low-cost spectrum analyzer, many researchers have carried out spectrum measurements around the world. For example, [20] carried out spectrum measurement in Trieste, Italy, from 400 MHz up to 800 MHz frequency band, while [21] performed spectrum occupancy measurement in UHF TV band in Venezuela. The use of such a low-cost spectrum analyzer enables us to gather a huge amount of data, paving the way towards deep learning approaches.

More recently, deep learning has been applied to the spectrum sensing problem, which is treated as a classification problem where a classifier decides the state of the frequency channels being sensed as either busy or free. On the basis of cooperation among SUs, spectrum sensing techniques can be classified as either cooperative or non-cooperative. Most of the works on spectrum sensing employing deep learning have focused on Cooperative Spectrum Sensing (CSS). In [22], the authors proposed a CSS approach using an energy detector to allow the cognitive radios in the same band to
In more details, let us consider a frequency band time, across any location, environment and frequency bands. The allocation of channels at location A to PU1 and at location B different across the different locations. The overall situation is used by the PU at a particular location and at a particular time across the different locations. The problem here is that PU2 are different. The objective of the SU is to be able to detect channels occupied by PUs with minimal sensing towards PUs. The allocation of channels to the PUs is different across the different locations. The problem here is that may reuse all channels provided there are no interferences.

memory (LSTM), and Fully connected CNN (FCNN). However, the experiment in [12] is only performed in a laboratory environment with a 15 cm distance between the transmitting and receiving antennas.

In [11], the so-called SPN-43 radar detection is carried out in 3.5 GHz band spectrograms and shows that a three-layer CNN architecture offers a superior tradeoff between accuracy and computational complexity. However, [11] performs detection on two measurement sites using spectrograms without exploiting transfer learning. Similarly, [12] designed a DNN architecture composed of CNN, Long Short Term Memory (LSTM), and Fully connected CNN (FCNN). However, the experiment in [12] is only performed in a laboratory environment with a 15 cm distance between the transmitting and receiving antennas.

III. SYSTEM MODEL

The overall system model consists of a CR network whereby PUs make use of a set of channels at a particular location, and a SU performs non-cooperative spectrum sensing, i.e., the SU may reuse all channels provided there are no interferences towards PUs. The allocation of channels to the PUs is different across the different locations. The problem here is that the SU needs to determine the channels that are not being used by the PU at a particular location and at a particular time across the different locations. The overall situation is depicted in the example of Fig. 1. The SU can move within the same location A or to a different location B, where the allocation of channels at location A to PU1 and at location B to PU2 are different. The objective of the SU is to be able to detect channels occupied by PUs with minimal sensing time, across any location, environment and frequency bands.

In more details, let us consider a frequency band $f_1$ Hz to $f_2$ Hz, where each channel has a bandwidth of $B$ Hz. The total number of channels is given by $T_n = \frac{f_2-f_1}{B}$. Let $\mathcal{N}_c = \{1, 2, \ldots, T_n\}$ be the set of all channels. PUs occupy a certain number of channels in $\mathcal{N}_c$ at a particular location $l \in L$. Let $\mathcal{N}_c^o$ be the set of channels being occupied by PUs, such that, $\text{Card}(\mathcal{N}_c^o) < \text{Card}(\mathcal{N}_c)$, and $\mathcal{N}_c^o \subset \mathcal{N}_c$ at a particular location $l$, where Card(·) denotes the cardinality of the set.

When $\text{Card}(\mathcal{N}_c^o) < \text{Card}(\mathcal{N}_c)$, the spectrum is underutilized. The problem here is to determine whether the received signal at a particular channel, location, and environment is being occupied by PUs or not. This problem can be represented as a binary hypothesis testing at each channel. Let $y_{\mathcal{N}_c}^{l,e}(t)$ be the received signal at SU for a certain period of time $t = 0 \rightarrow T$, then the received signal can be written as:

$$y_{\mathcal{N}_c}^{l,e}(t) = \begin{cases} x_{\mathcal{N}_c}^{l,e}(t) + w_{\mathcal{N}_c}^{l,e}(t) & \text{H1} \\ w_{\mathcal{N}_c}^{l,e}(t) & \text{H0} \end{cases}$$

where $x_{\mathcal{N}_c}^{l,e}(t)$ is the sample of the PU signal at channel $\mathcal{N}_c$ in location $l$ in environment $e$ at time $t$, $w_{\mathcal{N}_c}^{l,e}(t)$ is the Additive White Gaussian Noise (AWGN) at any channel in the set $\mathcal{N}_c$ in location $l$ in environment $e$ at time $t$, H1 is the hypothesis that PU signal is present, and H0 the hypothesis that PU signal is absent. In Eq. (1), parameter $l$ represents the location where the signal is received, namely urban, suburban or rural (see section VI-A for more details). Environment $e$ refers to the indoor and outdoor scenarios at a particular location where the signal is received. We model the received signal taking into account these parameters so that we can analyze the effects of received signal quality for spectrum-sensing across different environments at a particular location. Parameters $l$ and $e$ are used in Section V to describe our proposed algorithm. The performance of the sensing algorithm is measured in terms of Probability of Detection ($P_d$), and Probability of False Alarm ($P_{fa}$) [27], defined as

$$P_d = P[H1|H1],$$

$$P_{fa} = P[H1|H0],$$

where $P_d$ is the probability of detecting the PU signal given that it is actually present and $P_{fa}$ is the probability of detecting the PU signal given that it is actually absent. The classification performance of the deep CNN needs to satisfy the two sensing requirements, i.e., $P_d$ and $P_{fa}$ as specified by the IEEE 802.22 standard. Assuming the positive class as busy channel and the negative class as free channel, $P_d$ and $P_{fa}$ correspond.
to True Positive (TP) and False Positive (FP), respectively, in terms of confusion matrix. The value of the other two parameters in confusion matrix, i.e., False Negative (FN) and True Negative (TN) in terms of probabilities, can be calculated by $1 - P_d$ and $1 - P_{fa}$, and referred as probability of miss detection ($P_{md}$) and probability of correct rejection ($P_{cr}$), respectively. The sensing algorithm requires to have high $P_d$ in order to avoid interference to PU, and low $P_{fa}$ to ensure high spectral efficiency. In order to protect PUs, the FCC has set some requirements on spectrum sensing. For example, in IEEE 802.22 WRAN CR standard, the requirements for the above metrics are $P_d \geq 90\%$, $P_{fa} \leq 10\%$. In addition, the sensing time denoted $T_{sens}$, which is defined as the time required by the PU to sense the channel, should satisfy $T_{sens} \leq 2$ seconds [28], [29].

IV. REFERENCE APPROACHES

In traditional spectrum sensing methods, the first step after receiving the signal is the computation of test statistics, as shown in Fig. 2(a). The computation of test statistics is predefined by the designer and depends upon the knowledge the algorithm has about the PU. For example, in an energy detection algorithm, no knowledge of PU is assumed, and the test statistics correspond to computing the energy of the received signal. After the computation of test statistics, it is compared to a predefined threshold parameter, and the final decision is made about the state of the channel.

The spectrum sensing problem in (1) can be formulated as a classification problem in the context of machine learning, as shown in Fig. 2(b). In particular, if the received samples are arranged in a time-frequency spectrogram, the problem can be cast as an image classification problem into two hypotheses, i.e., H0 or H1. The feature extraction process can be done either manually or through deep learning, in particular, CNN, for automatic feature extraction. Manual feature extraction can only extract the features known to the designer, i.e., it cannot extract hidden features of the data. Finally, the Machine Learning (ML) algorithm such as, SVM, KNN, GMM, etc. which has been trained using labeled data, enables to output the sensing decision. Hence, the computation of test statistics in the traditional method is replaced by feature extraction in the ML paradigm. Furthermore, the offline threshold computation, which required some assumptions in the traditional approach (e.g., known noise variance in the case of energy detector), is now based on the available data for training the learning parameters in the novel ML-based approach.

V. PROPOSED METHOD

In this section, we explain the proposed CNN architecture with linear SVM for spectrum sensing and the exploitation of transfer learning by the proposed method.

A. PROPOSED CNN BASED SVM

The proposed system consists of two main parts as shown in Fig. 3:

(a) a deep CNN used for the automatic feature extraction from the spectrogram plots,

(b) a linear SVM classifier that is trained for the classification task using features extracted from the deep CNN component.

In Fig. 3, time-frequency representation called spectrogram of the received signal in (1) is created at each location $l$ and environment $e$ for all channels. The spectrogram of a busy channel is quite different from that of a free channel. Thus, by appropriate training, a deep CNN can effectively learn to classify these spectrograms. In Fig. 3, the first part of the proposed method, i.e., the deep CNN, extracts features from these spectrograms. However, before any feature extraction can occur, the deep CNN model itself needs to learn features from the spectrograms. For that, the spectrograms are labeled manually using the ground truth data explained in detail in Section VI. After labeling, we divide the spectrograms into three sets, i.e., training, validation, and testing. The training set and the validation set trains the deep CNN for a specific number of images and epochs. After training, the different layers of the deep CNN learn features according to the spectrogram plots at a particular location and environment. Next, the features learned from the deep CNN are used to train the linear SVM. The test set evaluates the performance of the linear SVM to determine whether it fulfills the first two sensing requirements, i.e., $P_d \geq 90\%$ and $P_{fa} \leq 10\%$. Note that the test set is composed of spectrograms that are unseen by the deep CNN and linear SVM during the training phase. If the trained linear SVM classifier’s performance on the test set satisfies the first two sensing requirements, the trained model, i.e., the deep CNN along with the linear SVM classifier, is used further for transfer learning. However, if the linear SVM classifier does not satisfy the two sensing requirements, we retrain the deep CNN by increasing either the number of images or epochs until the requirements get fulfilled by the linear SVM. We now explain the details of the deep CNN.

The designed deep CNN architecture is inspired by the AlexNet [30] architecture, since it was the first CNN-based winner of ImageNet Large Scale Visual Recognition.
Challenge (ILSVRC) and given its low complexity as compared to other CNN architectures. As shown in Fig. 4, our CNN consists of four convolution layers for feature extraction. This design is chosen to further reduce the computation complexity by exploiting various hyperparameters as described in Section V. Each convolutional layer is followed by a batch normalization, ReLu, and a maxpool layer. We denote the convolutional layer and maxpool layer as $\text{conv}_i$ and $\text{maxpool}_i$, respectively for the $i^{th}$ layer, where $i \in \{1, 2, 3, 4\}$. Each convolutional layer has $F_i$ filters of size $f_i \times f_i$ and each maxpool layer has pool size of $P_i$. The total weights and bias of $i^{th}$ convolutional layer are denoted by $W_i$ and $B_i$, respectively. Furthermore, each convolutional and maxpool layer has stride, and padding, denoted at layer $i$ as $CS_i$ and $CP_i$, respectively. Similarly, we denote the stride of maxpool layer as $MS_i$. The stride is the number of pixels the filter moves after each operation for both the convolution or max-pool and padding is the number of zeros added to either side of the boundaries of the input [31]. The features are extracted from the last maxpool layer, i.e., maxpool_4. Let $x_i \times x_i \times y_i$ denote the size of the input at layer $i$. The output size of the $i^{th}$ convolution layer is denoted as $O_{\text{conv}_i}$ and is computed as

$$O_{\text{conv}_i} = \frac{x_i - F_i + 2CP_i}{CS_i} + 1. \quad (3)$$

As ReLu and batch normalization do not change the output size, it is given for the maxpool layer as

$$O_{\text{maxpool}_i} = \frac{x_i - P_i}{MS_i} + 1. \quad (4)$$

Note that the proposed system is different from that of [11] and [12], where the softmax layer is used as the final layer as a classifier for predicting and minimizing cross-entropy loss. The use of linear SVM and softmax was investigated in [32] and disclosed significant gains on popular deep learning datasets MNIST, CIFAR-10, and the ICML 2013 Representation Learning Workshops face expression recognition challenge. We investigated the classification performance of both the softmax layer and the linear SVM for the proposed method and found that a softmax layer has lower performance than a linear SVM, as shown in our experiments in Section VI-F-1. Thus, in the proposed method, we use a linear SVM as a classifier.

**B. TRANSFER LEARNING**

The initial motivation for transfer learning was to apply prior knowledge about a given task in order to solve a new but similar task instead of learning from scratch, thereby saving significant time and computation burdens. According to [33], given a source domain $D_S$ and learning task $T_S$, a target domain $D_T$ and learning task $T_T$, transfer learning aims at improving the learning of the target predictive function $f_T(\cdot)$ in $D_T$ using the knowledge in $D_S$ and $T_S$, where $D_S \neq D_T$, or $T_S \neq T_T$. There are three different general approaches for transfer learning, i.e., inductive, transductive, and unsupervised [33]. In the inductive approach, the source and target domains are the same, yet the source and target tasks differ from each other. In the transductive case, the source and target tasks are identical, but the corresponding domains are different. In addition, the source domain has many labeled data, while the target domain has none. Finally, in the unsupervised approach, the source and target domains are similar, but the tasks are different, with emphasis on unsupervised tasks in the target domain. Deep learning models are representative of inductive learning, where the algorithm works with a set of assumptions related to the distribution of the training data, known as inductive bias. Since inductive transfer techniques utilize the inductive biases of the source task to assist the target task, we apply transfer learning using this approach.

In the proposed method, we apply the concept of transfer learning by transferring knowledge, i.e., features learned at a particular location and environment across different locations and environments. We first train the deep CNN at a particular location and environment from scratch, i.e., by randomly initializing weights of all the layers of the deep CNN. While
We use the following notations: $l(e) \rightarrow l'(e')$ denotes transfer learning to location $l'$ and environment $e'$ using pre-trained model $l(e)$, with either $l \neq l'$ or $e \neq e'$.

$S_{l,e}$ denotes spectrograms at $l$ and $e$.

$N_{l,e}$ denotes the required number of spectrograms at $l$ and $e$ to satisfy $P_D$ and $P_{fa}$.

$N_{l,l,e}$ is the total number of spectrograms at $l$ and $e$.

To determine $l(e)$ and $N_{l,e}$ we use Algorithm 1. The received signal in Eq.(1) is first pre-processed to create spectrograms $S_{l,e}$. The inputs of Algorithm 1 are $M_{cnn,svm}$ and $S_{l,e}$ and outputs are $l(e)$ and $N_{l,e}$. The main steps in Algorithm 1 are:

1) First, we divide the available spectrograms $S_{l,e}$ into training, validation, and test sets, and then create a trained CNN model at $l(e)$ using $S_{l,e}$ and the initialized CNN $M_{cnn,svm}$ (lines 1 to 2).

2) Second, we compute $P_D$ and $P_{fa}$ for the linear SVM which is trained using features extracted from maxpool4 of $l(e)$ using the test set (lines 3 to 5).

3) Finally, we check whether $P_D$ and $P_{fa}$ fulfill sensing requirements. If so, we save the pre-trained model $l(e)$ and the corresponding value of $N_{l,e}$, otherwise we increase $N_{l,e}$ and re-iterate, until both requirements are fulfilled (steps 6 to 12).

After we have created $l(e)$ using Algorithm 1, we apply transfer learning by using Algorithm 2 to create $l'(e')$ and determine $N_{l',e'}$. The inputs of Algorithm 2 are $l(e)$ and $S_{l',e'}$ and outputs are $l'(e')$ and $N_{l',e'}$. Algorithm 2 can be summarized as follows:

1) First, we divide the available spectrograms at $l'$ and $e'$, i.e., $S_{l',e'}$ into training, validation, and test sets, and then create a trained CNN model at $l'(e')$ using $S_{l',e'}$ and the pre-trained model $l(e)$ by freezing its first two layers and...
Algorithm 1 Proposed CNN Pre-Training Algorithm

Input: $M_{\text{cnn,svm}}$ and $S_{l,e}$
Output: $l(e)$ and $N_{l,e}$
1: Create Training, validation, and test sets from $S_{l,e}$.
2: Train $l(e)$ using training and validation sets from $S_{l,e}$ and $M_{\text{cnn,svm}}$.
3: Extract features from $\text{maxpool}_4$ from $l(e)$ for the test set.
4: Train linear SVM with extracted features from test set.
5: Compute $P_d$ and $P_{fa}$ using Eq. (2) for the test set.
6: if $P_d \geq 90\%$ and $P_{fa} \leq 10\%$ then
7: save $l(e)$ and $N_{l,e}$
8: go to step 12
9: else
10: $N_{l,e} = N_{l,e} + 10$
11: go to step 1
12: end if
13: return $l(e)$ and $N_{l,e}$

Algorithm 2 Proposed CNN Transfer Learning Algorithm

Input: $l(e)$ and $S_{l',e'}$
Output: $l'(e')$ and $N_{l',e'}$
1: Create Training, validation, and test sets from $S_{l',e'}$.
2: Freeze the weights of $\text{conv}_1$, $\text{conv}_2$, $\text{maxpool}_1$, and $\text{maxpool}_2$ of the pre-trained CNN model $l(e)$ and initialize weights of $\text{conv}_3$, $\text{conv}_4$, $\text{maxpool}_3$, and $\text{maxpool}_4$ using Gaussian initialization.
3: Train $l'(e')$ using $S_{l',e'}$ and $l(e)$.
4: Extract features from $\text{maxpool}_4$ from $l'(e')$ for the test set.
5: Train linear SVM with extracted features from test set.
6: Compute $P_d$ and $P_{fa}$ using Eq. (2) for the test set.
7: if $P_d \geq 90\%$ and $P_{fa} \leq 10\%$ then
8: save $l'(e')$ and $N_{l',e'}$
9: go to step 14
10: else
11: $N_{l',e'} = N_{l',e'} + 10$
12: go to step 3
13: end if
14: return $l'(e')$ and $N_{l',e'}$

by Gaussian initialization of the last two layers (lines 1 to 3),
2) Second, we compute $P_d$ and $P_{fa}$ for the linear SVM which is trained using features extracted from $\text{maxpool}_4$ of $l'(e')$ using the test set (lines 4 to 6),
3) Finally, we check whether $P_d$ and $P_{fa}$ fulfill sensing requirements. If so, we save the trained model $l'(e')$ and the corresponding value of $N_{l',e'}$, otherwise we increase $N_{l',e'}$ and re-iterate, until both requirements are fulfilled (steps 7 to 14).

Overall, we identify the following scenarios for transfer learning:

- Transfer learning across different environments at each location, i.e., $l(e) \rightarrow l'(e')$.
- Transfer learning across different environments and locations but with the same frequency assignment, i.e., $l(e) \rightarrow l'(e')$ represented as Case 1 in Fig. 1.
- Transfer learning across different environments and locations with different frequency assignments, i.e., $l(e) \rightarrow l'(e')$ represented as Case 2 in Fig. 1.

To summarize, Algorithm 1 is used to obtain the trained model at a particular location and environment from scratch. The algorithm outputs the number of images required to satisfy the two sensing requirements, i.e., $P_d \geq 90\%$ and $P_{fa} \leq 10\%$, when the model is trained from scratch. The second algorithm is used to obtain a trained model at a particular location and environment using a pre-trained model from another location and environment. This algorithm outputs the number of images required to satisfy the two sensing requirements, i.e., $P_d \geq 90\%$ and $P_{fa} \leq 10\%$ when transfer learning is applied.

VI. EXPERIMENTAL RESULTS

This section describes our experimental setups carried out in Thailand and presents the results for various scenarios.

A. MEASUREMENT SETTINGS

Fig. 5 shows the overall block diagram of the measurement setup. To perform all measurements, RFExplorer spectrum analyzer of model WSUB1G, fitted with an omnidirectional antenna (NA-773) with SMA male connector was used. The antenna is connected directly to the spectrum analyzer, which operates within the frequency range of 240 MHz to 960 MHz, the dynamic range from $-115$ dBm to 0 dBm, ±3 dB overall amplitude accuracy, and the average noise level of $-110$ dBm. The spectrum analyzer is connected to a Raspberry Pi 2 Model B by a Universal Serial Bus (USB) cable. Rfstatic, a bash script, was installed in Raspbian OS for controlling the various configuration parameters of the spectrum analyzer as shown in Table 1 while carrying out the measurements. The laptop is merely used as a monitor for Raspberry Pi.

The measurements were performed at three locations:
1. Bangkok for the Urban Area (UA),
2. Pathum Thani for the Sub-Urban Area (SUA),
3. Mae Sot for the Rural Area (RA),
and two different environments,
- a. indoor (in) with antenna height (above ground level):
  - 1 m,
TABLE 1. Configuration parameters used for the spectrum analyzer.

| Parameter         | Value       |
|-------------------|-------------|
| Frequency range   | 510 to 790 MHz |
| Center frequency  | 650 MHz     |
| Frequency span    | 280 MHz     |
| Resolution bandwidth | 500 KHz   |
| Sweep time        | automatic   |
| Reference level   | -50 dBm     |

b. outdoor (out) with antenna height (above ground level): 5 m (except for SUA(out) for which it was 10 m).

The measurements were performed for seven days except for Mae Sot, where they were for three days. The distance between the transmitter and measurement location (both indoor and outdoor) in Bangkok and Pathum Thani are 5.90 and 37.1 km, respectively, as shown in Fig. 6. Similarly, the distance between the transmitter and measurement location for outdoor and indoor in Mae Sot are 17.7 and 7.5 km, respectively, as shown in Fig. 7. Except for SUA(out) measurements obtained with Line-of-Sight (LOS), all the outdoor measurements have a clear blockage from surrounding buildings. The LOS situation for SUA(out) was further verified using Google Earth Pro, taking into account the height of the transmitter and receiver antennas. The raw data of all the collected spectrum data is available publicly in [34].

B. DIGITAL TV BAND SPECTRUM

In Thailand, the RF spectrum from 510 MHz to 790 MHz is allocated for Digital Terrestrial Television (DTT), with a total of 35 channels each with 8 MHz bandwidth [35]. The channels are numbered from 26 to 60. The channels being used for TV broadcast at UA and SUA are shown in Table 2 and for RA, in Table 3.

The Equivalent Isotropic Radiated Power (EIRP) and the height of the transmitter in Bangkok, which is also serving in Pathum Thani, are 80 dBm and 328 m, respectively, whereas those in Mae Sot are 67 dBm and 100 m, respectively. With the noise floor of -100 dBm, the received SNR levels measured for the best channel for each scenario are shown in Table 4. As observed, the SNR measured at SUA(out) is the largest of all even though it corresponds to the largest distance among our experiments. This is because the measurements at SUA(out) were the only ones conducted with the receiver antenna at the height of 10 m, whereas it was fixed to 5 m for all other outdoor measurements, as mentioned previously. The lowest SNR is observed at SUA(in) as it corresponds to the farthest location, and with the receiver antenna height of 1 m. Furthermore, the measured SNR value of RA(in) is higher than RA(out) because RA(in) measurement was conducted at a distance much closer than the one for RA(out) with respect to the transmitter in Mae Sot as can be seen in Fig. 7. Note that the SNR of the received signal at outdoor environment is higher than that of indoor environment, provided that both the indoor and outdoor environment measurements are carried.
C. SPECTROGRAM CREATION

In total, 29,070 spectrograms were created from the entire measurement campaign. The spectrograms span a 280 MHz frequency range, from 510 to 790 MHz, and a time interval of one minute. Spectrograms are created for each channel. Each spectrogram has dimensions $500 \times 16$ with 500 time-bins of duration 0.11 sec and 16 frequency-bins of length $8 \text{ MHz/16} \approx 500 \text{ kHz}$. Fig. 8 shows some of the busy and free channel spectrograms at SUA for different channels.

We label these spectrograms as being either busy or free from the report available from the National Broadcasting and Telecommunications Commission (NBTC) at each location. The spectrogram generated is used as input of the CNN model. For each location and environment, we use 70% of spectrograms for training, 15% for validation, and 15% for testing. After the CNN has been trained, the last maxpooling layer of the CNN model provides the features used to train the linear SVM classifier. The total number of features extracted is 1600. After training the linear SVM classifier, the confusion matrix is then calculated on the test images.

The parameter values of the proposed CNN are given in Table 5. Each convolution layer is followed by a batch normalization layer to prevent overfitting. A batch size of 10 and a learning rate of 0.01 were chosen, for their high validation accuracy carried out every epoch. Furthermore, we use Stochastic Gradient Descent with Momentum (SGDM) as an optimizer with a momentum equal to 0.9 and the loss function as cross entropy loss. The last maxpool layer gives a total of 1600 features, which are used to train the linear SVM classifier.

D. PERFORMANCE OF CONVENTIONAL SPECTRUM SENSING

We consider an Energy Detection (ED) method from the classical detection theory that is suitable for incoherent detection. Note that coherent detection techniques cannot be
applied with our data as these techniques require In-phase and Quadrature (I/Q) data. The ED is a conventional spectrum sensing algorithm based on the principle that the signal of interest can be detected by computing the total energy of the signal in a given time and frequency range, and comparing it against a pre-defined threshold. If the total energy exceeds the threshold, the channel is detected as busy. The value of the threshold can significantly affect the performance of ED that can be calculated based upon the $P_{fa}$ requirement.

The ED method is applied channel-wise to all the 35 channels of 8 MHz with 500 time-bins at each location and environment. To improve the performance of the ED, we do not use all the samples at each channel but select a fraction of the samples that give the highest SNR. Specifically, we select 10% of the total samples at each channel. The value of the threshold used determines the value of $P_d$ and $P_{fa}$ and is dependent on the location and environment [36]. Therefore, we determine the threshold iteratively such that $P_{fa} \leq 10\%$ at each measurement location and environment [37]. Then, the value of $P_d$ is calculated by dividing the number of busy channels detected by the total number of busy channels. Table 6 shows the $P_d$ of the conventional ED method for all measurement scenarios with $P_{fa} \leq 10\%$.

From Table 6, we can see that the performance of the conventional ED method is highly dependent upon the SNR value, i.e., it performs better at high SNR values as compared to low SNR values, achieving its best detection for SUA(out) with 73.14% and the worst at SUA(in) with 32.00%. Furthermore, if we compare the performance of the ED method at each location but in a different environment, we see that its performance is better in a high SNR environment as compared to low SNR. However, it is unable to jointly achieve both $P_{fa}$ and $P_d$ requirements, in any scenario.

### E. PRELIMINARY EVALUATION OF BENCHMARK CNN

Initially, we evaluate the performance of a benchmark pre-trained deep CNN model for feature extraction, namely AlexNet, where the weights of all convolutional layers except the last three fully connected layers have been frozen. Gaussian initialization was performed, i.e., initial weights were set by sampling from a normal distribution with mean 0 and standard deviation 0.01. To evaluate the performance of this pre-trained deep CNN model, image sizes were down-scaled so as to fit to the one required for AlexNet, i.e., 227 $\times$ 227 pixels. Features were extracted from spectrograms using all fully connected layers, i.e., fc6, fc7, and fc8. The best results were achieved when the features are extracted from the fc6 layer. Note that similar evaluations were made for snore sound classification in [38], whose best results were obtained by extracting features with the fc7 layer of the AlexNet architecture. The performance is analyzed in terms of the minimum number of images required to fulfill the first two sensing requirements, i.e., $P_{d} \geq 90\%$, and $P_{fa} \leq 10\%$ and shown in Table 7. It is observed that the sensing requirements are never met for scenarios SUA(in), UA(out), and RA(out) even if all the spectrograms are used. Table 7 shows that AlexNet is not able to fulfill the first two sensing requirements, i.e., $P_{d} \geq 90\%$, and $P_{fa} \leq 10\%$ even though the SNR level of RA(out) measurement is closer to RA(in) measurement. The reason that AlexNet is not able to fulfill the sensing requirement at RA(out) could be accounted for the fact that: a) AlexNet was trained on real-life objects so the features it has learned may not be suitable for spectrograms and b) we have a limited number of spectrograms for training at RA(out). However, if more spectrograms are collected in RA(out) and the AlexNet is trained on a larger set of spectrograms, the sensing requirements could have been fulfilled. Furthermore, these initial results show that in high SNR cases, namely, SUA(out), UA(in), UA(out), and RA(in), the extracted features by this benchmark CNN can be used for spectrum sensing.

### F. PERFORMANCE ANALYSIS OF THE PROPOSED SYSTEM

Here we give the results of the proposed method with and without transfer learning for all the cases in each scenario.

#### 1) PERFORMANCE WITHOUT TRANSFER LEARNING

We first determine the minimum number of images required to fulfill the first two sensing requirements by training the proposed CNN model from scratch. For this, we use Algorithm 1. Table 8 shows the number of images required to satisfy the sensing requirements for each of the considered six cases. From the obtained results, we see that the proposed system is able to fulfill the sensing requirements for SUA(in) and RA(out) and to reduce the required number of images for SUA(out) and UA(out), as compared to the benchmark performance of Table 7. Moreover, we observe that the least number of images required for fulfilling sensing requirements is for the SUA(out) scenario and the highest for SUA(in) scenario. A similar pattern is seen for UA, where UA(out) requires less number of images than UA(in) for fulfilling the sensing requirements. It may seem that measurements
in outdoor environment require less images as compared to indoor. However, this pattern is not applicable to RA, where RA(out) requires more images than RA(in). Therefore, we conclude that the number of images required to fulfill the sensing requirements is not solely dependent on environment. However, if we take SNR into account to determine the required number of images, we see that SUA(out), which has the highest SNR, requires the least number of images to fulfill the sensing requirements. In contrast, SUA(in), which has the lowest SNR, requires the highest number of images. Similar tendencies are observed in UA and RA, where UA(out) with high SNR requires less images compared to UA(in) with lower SNR, and likewise for RA(in) with high SNR compared to RA(out) with low SNR. Comparing the number of images required in each location for a different environment, we see that SUA(out) requires 97% less images as compared to SUA(in), and UA(out) requires 33% less images than by UA(in), while RA(out) requires 37% more images than by RA(in). Thus, we can conclude that when training the network from scratch, the number of images required to fulfill the sensing requirements mainly depends on the SNR level, irrespective of the location and the environment. Specifically, more images are required to fulfill the sensing requirements in low SNR cases as compared to high SNR cases. Moreover, when the SNR level is very low, the number of required images can be significantly higher, as in the case of SUA(in).

Furthermore, we show that a linear SVM for the proposed method outperforms softmax layer for classification purposes. To make the comparison fair, we train the deep CNN using the same number of spectrograms and with the same parameters. We compare the $P_d$ and the $P_{fa}$ of the softmax layer and the linear SVM for all measurement scenarios and observed that the linear SVM outperforms the softmax layer. For example, Tables 9 and 10 show the confusion matrices on the test set for the softmax and the linear SVM for SUA(in) by training the proposed method using 800 spectrograms.

2) PERFORMANCE WITH TRANSFER LEARNING

Here we provide results and discussions for all the scenarios described in Section III.

In this scenario, transfer learning is applied across different environments, i.e., indoor or outdoor, at the same location. There are in total six different cases, i.e., SUA(in) $\to$ SUA(out), SUA(out) $\to$ SUA(in), and similarly for UA and RA. Table 11 shows the number of images required to satisfy the sensing requirements with and without applying transfer learning for all six cases. We can see that at each location, applying transfer learning from indoor to outdoor environment enables to reduce the required number of images, provided the SNR of the indoor environment is lower than that of the outdoor environment. The three different cases where transfer learning reduces the number of images are: SUA(in) $\to$ SUA(out), UA(in) $\to$ UA(out), and RA(out) $\to$ RA(in). The highest reduction in the number of images is observed for SUA(in) $\to$ SUA(out), i.e., 60% less images using transfer learning from SUA(in) as compared to SUA(out) without transfer learning. For UA(in) $\to$ UA(out), it is observed that UA(out) requires 14% less images as compared to UA(out) without transfer learning. Finally, the lowest reduction is observed for RA(out) $\to$ RA(in), but the proposed transfer learning still enables an 11% reduction of images as compared to RA(in) without transfer learning. These results indicate that applying transfer learning from low SNR to high SNR environments at the same location enables a significant performance improvement. Furthermore, we also see that the highest reduction in the number of images is obtained when transfer learning is applied from low SNR to the highest SNR environments at the same location.

In this scenario, transfer learning is applied across different environments, i.e., indoor or outdoor, at the same location. There are in total six different cases, i.e., SUA(in) $\to$ SUA(out), SUA(out) $\to$ SUA(in), and similarly for UA and RA. Table 11 shows the number of images required to satisfy the sensing requirements with and without applying transfer learning for all six cases. We can see that at each location, applying transfer learning from indoor to outdoor environment enables to reduce the required number of images, provided the SNR of the indoor environment is lower than that of the outdoor environment. The three different cases where transfer learning reduces the number of images are: SUA(in) $\to$ SUA(out), UA(in) $\to$ UA(out), and RA(out) $\to$ RA(in). The highest reduction in the number of images is observed for SUA(in) $\to$ SUA(out), i.e., 60% less images using transfer learning from SUA(in) as compared to SUA(out) without transfer learning. For UA(in) $\to$ UA(out), it is observed that UA(out) requires 14% less images as compared to UA(out) without transfer learning. Finally, the lowest reduction is observed for RA(out) $\to$ RA(in), but the proposed transfer learning still enables an 11% reduction of images as compared to RA(in) without transfer learning. These results indicate that applying transfer learning from low SNR to high SNR environments at the same location enables a significant performance improvement. Furthermore, we also see that the highest reduction in the number of images is obtained when transfer learning is applied from low SNR to the highest SNR level, i.e., from SUA(in) $\to$ SUA(out).

### TABLE 8. Number of images required to fulfill $P_d \geq 90\%$ and $P_{fa} \leq 10\%$, proposed CNN without transfer learning for each location and environment.

| Scenario    | Number of Images |
|-------------|------------------|
| SUA(in)     | 5680             |
| SUA(out)    | 100              |
| UA(in)      | 210              |
| UA(out)     | 140              |
| RA(in)      | 540              |
| RA(out)     | 560              |

### TABLE 9. Confusion matrix of softmax for SUA(in).

| True | Predicted |
|------|-----------|
| busy | 102       |
| free | 18        |
| busy | 26        |
| free | 94        |

### TABLE 10. Confusion matrix of linear SVM for SUA(in).

| Scenario    | Number of Images (with transfer learning) | Number of Images (without transfer learning) |
|-------------|------------------------------------------|--------------------------------------------|
| SUA(in) $\to$ SUA(out) | 40 | 100 |
| SUA(out) $\to$ SUA(in) | 1744 | 3880 |
| UA (in) $\to$ UA(out) | 120 | 140 |
| UA(out) $\to$ UA(in) | 210 | 210 |
| RA(in) $\to$ RA(out) | 560 | 560 |
| RA(out) $\to$ RA(in) | 480 | 540 |

### TABLE 11. Number of images required to fulfill $P_d \geq 90\%$ and $P_{fa} \leq 10\%$, proposed CNN using transfer learning across environments (in $\leftrightarrow$ out), but at the same location.

| Scenario       | Number of Images |
|----------------|------------------|
| SUA(in) $\to$ SUA(out) | 40 |
| SUA(out) $\to$ SUA(in) | 1744 |
| UA (in) $\to$ UA(out) | 120 |
| UA(out) $\to$ UA(in) | 210 |
| RA(in) $\to$ RA(out) | 560 |
| RA(out) $\to$ RA(in) | 480 |
corresponding to SUA and UA environments. Table 12 shows the number of images required to satisfy the sensing requirements with and without transfer learning for all applicable eight cases. We can see that transfer learning reduces the required number of images in three different cases, i.e., SUA(in) → UA(in), SUA(in) → UA(out), and SUA(out) → UA(in). The largest reduction is observed for SUA(in) → UA(in), which achieves a 57% reduction, as compared to UA(in) without transfer learning. Comparing these results to the previous scenario, where transfer learning was applied across environments but at the same location, we can realize that here, the largest reduction does not occur when transfer learning is applied to the highest SNR level case, but when transfer learning is applied from the lowest SNR level (1.8 dB) to the second-lowest SNR level (15.3 dB), i.e., SUA(in) → UA(in). Similarly, there is a reduction of 7% and 28% for SUA(in) → UA(out) and SUA(out) → UA(out), respectively. By contrast to the previous case where transfer learning reduced the required number of images only from low SNR to high SNR levels, here, transfer learning enables such reductions even from high SNR to low SNR levels, i.e., SUA(out) → UA(out). However, the reduction from SUA(in) is lower than that for SUA(out), which may be explained by the fact that the SNR level of SUA(out) is closer to that of UA(out) as compared to SUA(in). Overall, we conclude in this case that transfer learning from low SNR to high SNR always requires an either lower or equal number of images while transfer learning from high SNR to low SNR can achieve reduction provided that their SNR levels are close enough.

**C: TRANSFER LEARNING ACROSS LOCATIONS AND ENVIRONMENTS, DIFFERENT FREQUENCY ASSIGNMENTS**

In this scenario, transfer learning is applied across locations and environments with different frequency assignments, corresponding to SUA ↔ RA and UA ↔ RA. This scenario is represented as Case 2 in Fig. 1. There are a total of sixteen different cases, eight for SUA ↔ RA and eight for UA ↔ RA. Table 13 shows the number of images required to satisfy the sensing requirements with and without transfer learning between SUA and RA environments for all applicable eight scenarios.

### Table 12. Number of images required to fulfill \( P_d \geq 90\% \) and \( P_{fa} \leq 10\% \), proposed CNN using transfer learning across locations (SUA ↔ UA) and environments (in ↔ out), same frequency assignment.

| Scenario          | Number of Images (with transfer learning) | Number of Images (without transfer learning) |
|-------------------|------------------------------------------|---------------------------------------------|
| SUA(in) → UA(in)  | 90                                       | 210                                         |
| SUA(in) → UA(out) | 130                                      | 140                                         |
| SUA(out) → UA(in) | 240                                      | 210                                         |
| SUA(out) → UA(out)| 100                                      | 140                                         |
| UA(in) → SUA(in)  | 3744                                     | 3680                                        |
| UA(in) → SUA(out) | 120                                      | 100                                         |
| UA(out) → SUA(in) | 3744                                     | 3680                                        |
| UA(out) → SUA(out)| 100                                      | 100                                         |

### Table 13. Number of images required to fulfill \( P_d \geq 90\% \) and \( P_{fa} \leq 10\% \), proposed CNN with transfer learning across locations (SUA ↔ RA) and environments (in ↔ out), different frequency assignments.

| Scenario          | Number of Images (with transfer learning) | Number of Images (without transfer learning) |
|-------------------|------------------------------------------|---------------------------------------------|
| SUA(in) → RA(in)  | 560                                      | 540                                         |
| SUA(in) → RA(out) | 560                                      | 560                                         |
| SUA(out) → RA(in) | 420                                      | 540                                         |
| SUA(out) → RA(out) | 560                                      | 560                                         |
| RA(in) → SUA(in)  | 3680                                     | 3680                                        |
| RA(in) → SUA(out) | 40                                       | 100                                         |
| RA(out) → SUA(in) | 3680                                     | 3680                                        |
| RA(out) → SUA(out)| 40                                       | 100                                         |

### Table 14. Number of images required to fulfill \( P_d \geq 90\% \) and \( P_{fa} \leq 10\% \), proposed CNN with transfer learning across locations (UA ↔ RA) and environments (in ↔ out), different frequency assignments.

| Scenario          | Number of Images (with transfer learning) | Number of Images (without transfer learning) |
|-------------------|------------------------------------------|---------------------------------------------|
| UA(in) → RA(in)   | 440                                      | 540                                         |
| UA(in) → RA(out)  | 560                                      | 560                                         |
| UA(out) → RA(in)  | 540                                      | 540                                         |
| UA(out) → RA(out) | 560                                      | 560                                         |
| RA(in) → UA(in)   | 160                                      | 210                                         |
| RA(in) → UA(out)  | 140                                      | 140                                         |
| RA(out) → UA(in)  | 200                                      | 210                                         |
| RA(out) → UA(out) | 100                                      | 140                                         |

There are three cases where transfer learning provides performance gains, i.e., SUA(out) → RA(in), RA(in) → SUA(out), and RA(out) → SUA(out). In general, transfer learning reduces the required number of images when applied from low SNR to high SNR levels except when it is applied from lowest SNR level, i.e., SUA(in) → RA(in) and SUA(out) → RA(out). There is a 60% reduction of images for the other two cases of low SNR to high SNR levels, i.e., RA(in) → SUA(out) and RA(out) → SUA(out). On the other hand, when transfer learning is applied from high SNR to low SNR, there is only one case, i.e., SUA(out) → RA(in) that reduces the required number of images, by 22%. This case is different from the other three high SNR to low SNR levels in that transfer learning is applied from the highest SNR level (30 dB) to the second-highest SNR level (14.3 dB).

Next, Table 14 shows the number of images required to satisfy the sensing requirements with and without transfer learning between UA and RA environments. There are four cases in this scenario where transfer learning requires less number of images, i.e., UA(in) → RA(in), RA(in) → UA(in), RA(out) → UA(in), and RA(out) → UA(out). For all the low SNR to high SNR cases, transfer learning either reduces the required number of images, or achieves a similar level compared to the case without transfer learning. The highest reduction, namely 29%, is achieved by RA(out) → UA(out). On the other hand, for high to low SNR levels, transfer learning achieves reduction only for UA(in) → RA(in) with a 1dB difference in SNR level, which is the smallest.
TABLE 15. Required Number of images to fulfill $P_d \geq 90\%$ and $P_{fa} \leq 10\%$ with and without transfer learning for the proposed CNN.

| Scenario        | Number of images without transfer learning | Number of images with transfer learning |
|-----------------|------------------------------------------|----------------------------------------|
| SUA(out) → SUA(out) | 100                                      | 40                                     |
| UA(out) → UA(out)    | 140                                      | 120                                    |
| RA(out) → RA(in)     | 540                                      | 480                                    |
| SUA(out) → UA(in)    | 210                                      | 90                                     |
| SUA(out) → UA(out)   | 140                                      | 130                                    |
| SUA(out) → UA(out)   | 140                                      | 100                                    |
| SUA(in) → SUA(out)   | 100                                      | 40                                     |
| RA(out) → SUA(out)   | 100                                      | 40                                     |
| UA(in) → RA(in)      | 540                                      | 440                                    |
| RA(in) → UA(in)      | 210                                      | 160                                    |
| RA(out) → UA(in)     | 210                                      | 200                                    |
| RA(out) → UA(out)    | 140                                      | 100                                    |

In conclusion, for the cases where transfer learning is used across environments at the same location, and across locations and environments with the same frequency assignment, substantial performance gains are achieved compared to without transfer learning, when applied from low to high SNR levels. For high to low SNR levels, transfer learning achieves reductions only when applied from the highest SNR level to the second-highest, otherwise better results are achieved without transfer learning. On the other hand, using transfer learning across environments and locations with different frequency assignments enables significant performance gains in all cases except when applied from the lowest SNR level. When applied from high to low SNR levels, transfer learning enables reduction only from the highest to the second-highest SNR level, otherwise learning from scratch achieves better results. Finally, Table 15 summarizes the scenarios for which transfer learning provides a substantial reduction of the required number of images to fulfill the first two sensing requirements.

G. TRAINING TIME, NUMBER OF EPOCHS, SENSING TIME, AND COMPLEXITY ANALYSIS

In this section, the scenarios identified in Table 15 are further analyzed in terms of training time, number of epochs, and sensing time. Furthermore, we discuss the complexity analysis of the proposed algorithm.

1) NUMBER OF EPOCHS REQUIRED FOR TRAINING

To optimize the number of epochs required for training the network, we first determine the minimum number of images required to fulfill the $P_d$ and $P_{fa}$ requirements for each scenario using transfer learning, i.e., from Table 15. Next, for each scenario we fix the minimum number of required images and train the network in ascending order of the number of epochs, starting from 5 epochs with an increment of 5 epochs. For each number of epochs, the network is trained and analyzed to determine whether it satisfies the two sensing requirements, i.e., $P_d$ and $P_{fa}$. This process is repeated until the minimum number of epochs for which the network satisfies the two sensing requirements is reached. Figs. 9 and 10 show the number of epochs versus $P_d$ and $P_{fa}$ in percentage, for cases RA(out) → UA(in) and SUA(in) → UA(in), respectively. Clearly, when the number of epochs for which the network is trained is small, both requirements are not jointly fulfilled, but they are fulfilled after a sufficient number of epochs, namely after 25 epochs for RA(out) → UA(in) and 20 epochs for SUA(in) → UA(in). By repeating the aforementioned procedure, Table 16 summarizes the number of epochs required for all cases.

TABLE 16. Number of epochs required to fulfill $P_d \geq 90\%$ and $P_{fa} \leq 10\%$ for all cases with minimum number of images.

| Scenario        | Number of epochs |
|-----------------|------------------|
| SUA(in) → SUA(out) | 10              |
| UA(in) → UA(out)    | 45              |
| RA(out) → RA(in)     | 10              |
| SUA(in) → UA(in)    | 20              |
| SUA(in) → UA(out)   | 15              |
| SUA(out) → UA(out)  | 45              |
| SUA(out) → UA(in)   | 40              |
| RA(in) → SUA(out)   | 10              |
| RA(out) → SUA(out)  | 5               |
| UA(in) → RA(in)     | 20              |
| RA(in) → UA(in)     | 5               |
| RA(out) → UA(in)    | 25              |
| RA(out) → UA(out)   | 10              |
for both transfer and non-transfer learning methods. The accuracy for training and validation is computed by the following equation, using the training and validation sets at each epoch, respectively:

\[
\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}.
\] (5)

Figs. 11 and 12 show the curves for training and validation accuracies as well as training and validation losses for without transfer learning in RA(in) and with transfer learning in RA(in) using pre-trained CNN from RA(out), i.e., RA(out) → RA(in), with 480 images and 10 epochs. It is shown that thanks to the proposed pre-training and transfer learning, the validation accuracy curve starts at a higher value and reaches a higher value, i.e., 63% and 91% as compared to the case without using transfer learning, i.e., 50% and 81%. Furthermore, the validation loss curve starts at a low value and reaches a lower value, i.e., 0.76 and 0.25, as compared to the case without using transfer learning, i.e., 1.30 and 0.36. Similar tendencies are observed for all other cases in Table 15.

2) COMPARISONS OF REQUIRED TRAINING TIME

Even though the training phase is done offline, it is still important to reduce the training time of the network as it saves not only time but also computation as well as energy resources. Here, we determine the training times for the selected scenarios in Table 15, where we make use of a Graphics Processing Unit (GPU), as shown in Table 17. We can clearly see the reduction in training time achieved by the proposed CNN based on transfer learning. The minimum percentage of reduction in training time is 67% for SUA(out) → UA(out), and the maximum is 94% for RA(out) → SUA(out) with transfer learning as compared to non-transfer learning.

3) COMPARISONS OF REQUIRED SENSING TIME

Finally, we compute the sensing time taking into account the time required by the pre-trained network for each scenario to extract the features from the test set and the time required by the network to give the final decision about the state of the channel for all test set images. This process is repeated one thousand times, and the average sensing time for each image is computed. The timing measurements were performed solely on the CPU, for fair comparisons. Table 18 shows the sensing time required for all scenarios. Clearly, the sensing time is significantly reduced by using transfer learning, for all scenarios, for instance, up to 21% reduction for scenario SUA(in) → UA(in). This is a notable enhancement offered by our proposed CNN, as these major savings in sensing time entail large overhead reductions. Namely, the proposed method enables much longer data transmission durations, thereby increasing the overall throughput of the CR
system, while ensuring the fulfillment of all spectrum sensing requirements.

4) COMPLEXITY ANALYSIS
As mentioned previously, the proposed deep CNN based linear SVM architecture for spectrum sensing consists of two parts, i.e., the deep CNN for the feature extraction and the linear SVM as a classifier. Training and testing times are essential metrics for any ML models. These times were investigated previously for all scenarios. However, they cannot directly determine the time complexity of the proposed model. Traditionally, the testing time has been considered the more important of the two. However, whenever new data sets are available, the model has to retrain in order to update itself. To determine the training and testing time complexity of the proposed model, we measure the training and testing time as a function of training and testing set size, respectively. Figs. 13 shows the average training time of the proposed algorithm with various training set sizes for SUA(in) with and without the use of GPU for 50 epochs. Similarly, Fig. 14 shows the average testing time of the proposed algorithm without the use of GPU. For the computation of testing time, we train the linear SVM classifier beforehand using features extracted from the CNN architecture and then compute the testing time in increasing order of testing set size.

Fig. 13 shows that the training time scales linearly with the training set size for both with and without GPU. Hence, the proposed algorithm has a training time algorithmic complexity of approximately $O(n)$, where $n$ denotes the number of images of the training set. Similarly, Fig. 14 shows that the testing time complexity also has a complexity of $O(n)$ as the test time scales linearly with the number of images of the testing set.

In addition to analyzing training and testing time complexity, it is essential to measure the ML model’s computational complexity. To measure the computational complexity of the proposed model, we compute the Floating Point Operation (FLOP) count in each convolutional layer of our deep CNN architecture. The FLOP count of the deep CNN depends upon the network architecture, i.e., the number of layers, and the types of the layers. Following [39], we first determine the number of trainable parameters for our deep CNN architecture and then calculate the required FLOP count at each convolutional layer. Table 19 shows the FLOP count for each convolutional layer and the total FLOP count. Thus, the training and the testing time complexity of the proposed method scales linearly with training and testing set sizes, respectively. Furthermore, the GPU can reduce the computational burden of the training.

VII. CONCLUSION
In this work, we have investigated a machine learning-based non-cooperative spectrum sensing technique in a TV band cognitive radio system. We proposed a deep CNN architecture for automatic feature extraction and linear SVM for classifying the occupancy of the TV white band channels based on spectrograms. The proposed CNN makes use of transfer learning, generalizing its applicability across various
locations, wireless environments, and frequencies. The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.

The experiments conducted over three different locations in Thailand have shown that the proposed method could largely satisfy the sensing requirements of CR in all scenarios. Furthermore, we disclosed the benefits of using transfer learning for fulfilling sensing requirements in terms of reducing the number of images required for training, training time, and sensing time. Among other benefits, the reduction in sensing time makes the transfer learning a viable choice for increasing the overall throughput of the CR system.

In the future work, we plan to improve the proposed method to cope with severe deterioration of the channel qualities. Furthermore, we will investigate if the pre-trained method to cope with severe deterioration of the channel throughput of the CR system is a viable choice for increasing the overall throughput of the CR system.
B. Man Pati et al.: Deep CNN Based Transfer Learning Method for Non-Cooperative Spectrum Sensing

BIPUN MAN PATI received the bachelor’s degree in information technology from the Nepal College of Information Technology, Nepal, in 2011, and the master’s degree in telecommunication from the Asian Institute of Technology, Thailand, in 2016, where he is currently pursuing the Ph.D. degree in telecommunications. His master’s degree was sponsored by the Asian Development Bank Japan Scholarship Program. He worked as a Teaching Assistant at the Nepal College of Information Technology, until 2014. His research interests include wireless communications, signal processing, and machine learning.

MEGUMI KANEKO (Senior Member, IEEE) received the Diplôme d’Ingénieur degree from the Télécom SudParis (French Grande École), France, in 2004, the M.Sc. and Ph.D. degrees from Aalborg University, Denmark, in 2007, and the HDR degree (French Doctoral Habilitation for Directing Researches at professor position) from Paris-Saclay University, France, in May 2017. She was a JSPS Postdoctoral Fellow with Kyoto University, from April 2008 to August 2010. From September 2010 to March 2016, she was an Assistant Professor with the Department of Systems Science, Graduate School of Informatics, Kyoto University. She is currently an Associate Professor with the National Institute of Informatics and the Graduate University for Advanced Studies (SOKENDAI), Tokyo, Japan. Her research interests include wireless communications, 5G and beyond, the Internet of Things (IoT), wireless systems, and PHY/MAC design and optimization. She received the 2009 Ericsson Young Scientist Award, the IEEE GLOBECOM 2009 Best Paper Award, the 2011 Funai Young Researcher’s Award, the WPMA 2011 Best Paper Award, the 2012 Telecom System Technology Award, the 2016 Inamori Foundation Research Grant, and the 2019 Young Scientists’ Prize from the Minister of Education, Culture, Sports, Science and Technology of Japan. She serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE COMMUNICATIONS LETTERS, and IEICE TRANSACTIONS on Communications.

ATTAPHONGSE TAPARUGSSANAGORN received the B.Eng. degree from Chulalongkorn University, Bangkok, Thailand, in 1997, the M.Sc. degree in electrical engineering from the Technische Universität Kaiserslautern, Kaiserslautern, Germany, in 2001, and the D.Tech. degree from the University of Oulu, Oulu, Finland, in 2007. He worked as an Engineer at the Telecommunications Transmission Department, Siemens Ltd., Bangkok, for two years. He gained an experience as a Researcher at the Institute of Communications, University of Stuttgart, until 2003. He joined the Centre for Wireless Communications, University of Oulu, in 2003. In 2008, he worked as a Visiting Postdoctoral Researcher at Yokohama National University, Yokohama, Japan. After he returned home from Bangkok, in 2011, he joined the Asian Institute of Technology, Thailand, as an Adjunct Faculty Member. In 2012, he has also joined the National Electronics and Computer Technology Center, Thailand. Since August 2015, he has been a full-time Faculty Member with the Asian Institute of Technology. He is currently an Associate Professor with the Department of ICT. His research interests include signal processing, statistical signal processing (detection and estimation techniques), and artificial intelligence (AI) (machine learning) for various applications, for example, wireless communications, radio channel estimations, pattern recognition, coexistence problems, the Internet of Things (IoT), medical and healthcare ICT, and so on.