1. Introduction

Soon after the first radio communication system was invented in the late 19th century, the problem of radio interference from users using the same frequencies arose. As early as the 1820s, the United States made a radio spectrum frequency allocation chart of 0–60 MHz and initiated construction of a spectrum monitoring system. Subsequently, countries around the world began establishing monitoring systems for radio spectrum management. China has a national radio-monitoring system that covers all wavelengths between 20 and 3,000 MHz. However, the system is not automatic and is incapable of real-time spectrum analysis (Lu et al., 2017). Due to ever-increasing wireless applications, bandwidth demand is fast overwhelming the finite spectrum. To better manage the radio spectrum, many novel schemes have been proposed for the design of monitoring systems. Examples of state-of-the-art schemes include the permanent cloud-based system of systems for spectrum monitoring (Cooklev, 2015), the intelligent radio regulatory architecture based on edge computing (Huang et al., 2018), and “Electrosense” (Rajendran et al., 2018), which adopts a crowdsourcing paradigm to collect and analyze spectrum data using low-cost sensors. In addition, artificial intelligence (AI) has experienced rapid development in the last decade, and performance gains from using deep models have fostered progress. Apart from fields in language, voice, and vision (Fogel & Kvedar, 2018; LeCun et al., 2015) that have been intensively studied, deep learning has also shown its efficacy in spectrum signal analysis, such as identifying signals and modulation types at a low signal-to-noise ratio (Rajendran et al., 2018; Zhang et al., 2018). AI in radio monitoring is promising for bringing system automation and intelligence to spectrum management.

In China, illegal broadcasting occurs in almost every city. In 2018, national radio regulatory agencies investigated and dealt with 2,251 illegal broadcasting cases, ranking as the utmost concern among all radio interferences. Because the FM broadcasting band (87–108 MHz) is adjacent to the civil aircraft communications band (118–137 MHz), illegal broadcasting is a potential threat that can interfere with civil aviation navigation signals. However, the current approach to identify illegal broadcasting is by manual monitoring, which is not only time consuming but also hardly exploits the features of radio-monitoring systems. In this paper, we use a DCNN model to classify the acoustic features of speech and noise files collected from FM radio to identify FM broadcasting signals and then detect the illegal signals by verification with a database of legally registered radio stations.

2. Method and Process Description

A block diagram of the illegal broadcast identification process is shown in Figure 1. First, we scan and demodulate each channel with a bandwidth of 0.1 MHz one by one in the frequency range of 87–108 MHz to obtain the corresponding audio files in the Waveform Audio File (WAV) format. These files are stored and marked with the center frequency. Then, the acoustic features are extracted from the audio files.
together with the corresponding labels to generate a training dataset, which is then used to train the DCNN model with four convolutional layers (details are given in the following sections). The trained model is then evaluated for a test dataset to recognize all the FM broadcasting signals and save the corresponding frequencies. The illegal signals are identified by comparing all identified FM broadcasting signals with a radio frequency station database that contains a list of all authorized broadcast stations.

3. Feature Extraction

The training audio files consist of four categories: voice, noise, mute, and strong noise with weak speech. There are a total of 20,896 audio files in the training dataset, of which 10,445 are tagged as “voice” and the rest are tagged as the other three categories. The entire dataset is split into 80% for training and 20% for testing. To extract the acoustic features from an audio file, audio data are converted into a binary...
Figure 3. Block diagrams for obtaining the acoustic features of (a) PLP; (b) MFCCs.

Figure 4. DCNN structure diagram used by the classification and recognition modules.

Figure 5. t-SNE visualization of the DCNN learned features using different audio representations of the input data. (a) MFCCs; (b) FBank; (c) SDC; (d) PLP.
spectrum matrix. As shown in Figure 2, the left column illustrates speech waveforms, and the right side provides the corresponding spectral maps. From top to bottom, the panels correspond to the four kinds of audio samples, respectively. Because directly training the DCNN model using the original spectral matrix is time consuming, we first extract the acoustic features from audio data to facilitate the DCNN model training and testing. Here, four feature extraction methods are used: mel-frequency cepstral coefficients (MFCCs) (Zheng et al., 2001), mel-scale filter bank (FBank), shifted delta cepstral (SDC) (Torres-Carassquilo et al., 2002), and perceptual linear prediction (PLP) (Hermansky, 1990). Block diagrams for obtaining PLP and MFCCs are shown in Figure 3. SDC is mined by means of MFCC, and the benefit of SDC is that it is further strong under loud information. The FBank feature is the speech feature obtained by removing the discrete cosine transform in the last step of MFCCs and retains more original voice data than the MFCC features (Luo et al., 2018).

### 4. Classification Identification

The DCNN structure diagram is illustrated in Figure 4. The design of the convolutional feature extractor is based on the well-known LeNet architecture (Lecun et al., 1998). Our network uses four convolutional layers, where each layer is followed by batch normalization (Ioffe & Szegedy, 2015) and a ReLU activation function (Nair & Hinton, 2010). The kernel sizes and the number of filters for each convolutional layer are \((3 \times 5, 32), (3 \times 5, 64), (3 \times 3, 128)\), and \((3 \times 3, 128)\), respectively. The last convolutional layer is followed by three full connection layers, and the last layer has two output units serving as the signal and noise classifiers. Figure 4 provides a schematic overview of the network architecture. To handle variable-length audio data, we used global average pooling between the convolution layer and the full connection layer.

To explore the performance of the network, we used t-distributed stochastic neighbor embedding (t-SNE) to visualize the feature output from the last convolutional layer, as shown in Figure 5. t-SNE is an embedding technique that is commonly used for the visualization of high-dimensional data in scatter plots (van der Maaten & Hinton, 2008). The results show much better class separability for the FBank feature than for the other three acoustic features.

In this work, the DCNN takes FBank processed data from the audio file as input. Table 1 shows the comparison results of the recognition error rate of the DCNN trained with different data sizes based on 4,179 test samples.

| Training data (h) | Error rate (%) |
|-------------------|----------------|
| 0.5               | 1.2443         |
| 1                 | 0.3111         |
| 2.5               | 0.1675         |
| 5                 | 0.0957         |
| 10                | 0.0718         |
| 12                | 0.0479         |

Table 1: Testing results of the DCNN

Figure 6. System architecture.
samples. When the training is not enough, the error rate is slightly larger, as for the half-hour training data, and 52 samples are mistakenly classified. With more training data, the error rate drops accordingly. When the training data increase to 5 h, the error rate starts to converge with little change, and the model becomes stable.

5. System Design and Implementation

Leveraging the established DCNN, we implemented an FM radio-monitoring system consisting of 14 nodes, of which one node is deployed at Kunming and the other nodes are deployed at 13 counties in Honghe Prefecture of Yunnan Province. The system architecture is shown in Figure 6. The spectrum-sensing node consists of an RF receiver, a GPS receiver, a data processing and analysis module with edge computing features, and a wire/4G router that is used for communication between the sensing node and the backend cloud. A hardware block diagram of the sensing node is shown in Figure 7. The specifications of the RF receiver are provided as follows: frequency range: 1 kHz–2 GHz, bandwidth: 10 MHz, analog to digital conversion: 14 bit, typical noise figure: 3.3 dB @ 100 MHz, frequency tolerance: 0.5 ppm. The trained DCNN model is integrated into the spectrum-sensing node. In the working mode, the sensing node keeps scanning and demodulating each channel in the frequency range of 87–108 MHz and stores every baseband signal as a 5-minute-length audio file, which is then directed to the DCNN model for FM broadcast identification. In addition, a sensor node provides application programming interfaces (APIs) for the cloud to access the data at the node site, including a list of the detected FM broadcasts and the corresponding audio files. The cloud pulls the FM broadcast list through the APIs for illegal broadcasting identification. The spectrum-sensing node also provides web interfaces that allow users to access the results through a browser.

The system is designed using browser–server (BS) architecture and has been running steadily for half a year. Figure 8 shows the interface of the system, which is accessed from the radio private network through a browser. Here, Region is an online map, from which we can see each deployed sensor node. By clicking on the node icon, the broadcast stations monitored by the spectrum-sensing node at the corresponding location are displayed in Region, and the numbers of authorized and unlicensed broadcasts in each measurement are counted and shown above the list. Region shows the statistical results of the broadcast duration of each FM radio station monitored by the corresponding node in 1 day. By clicking the button in the upper-right corner, we can obtain a detailed statistical report of the authorized and unlicensed broadcast quantities and broadcast durations monitored by each sensing node. Region is the entrance to the station database.

Figure 7. Spectrum-sensing node hardware block diagram.

Figure 8. The interface of the FM broadcast monitoring system.
where after login authentication we can add, delete, and modify the authorized broadcast stations recorded in the database.

6. Conclusions

In this paper, a method of identifying FM broadcasts based on speech acoustic features and a DCNN is proposed. For comparison, four acoustic features, including PLP, MFCCs, FBank, and SDC, are extracted from audio files to train the DCNN model. The results show that for the FBank feature, when the training dataset duration is 12 h, the error rate can reach 0.0479%, that is, nearly 100% identification accuracy. Illegal broadcasting signals are then identified by comparing the results with an authorized radio station database. An illegal broadcast monitoring system based on 14 spectrum-sensing nodes designed using BS architecture is deployed in Kunming and Honghe Prefecture of Yunnan Province. This system relies on spectrum-sensing nodes integrated with the trained DCNN model to imitate the process in which radio-monitoring staff listen to FM broadcasts and record the frequency of illegal broadcasts. A use case demonstrates the potential of integrating the Internet of things (IoT) and AI technologies for intelligent and automatic radio monitoring.

Data Availability Statement

The dataset, code, and pretrained model for FM station identification can be obtained from Github (https://github.com/Judonlee/FM_Radio_Detecting).

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61963037, 61863035, and 11564044.

References

Cooklev, T., Darabi, J., McIntosh, C., & Mosaheb, M. (2015). A cloud-based approach to spectrum monitoring. IEEE Instrumentation & Measurement Magazine, 18, 33–37.

Fogel, A. L., & Kvedar, J. C. (2018). Artificial intelligence powers digital medicine. npj Digital Medicine, 1, 5.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. Journal of the Acoustical Society of America, 87, 1738–1752.

Huang, M., Lu, Q. N., Yang, J. J., & Chen, J. C. (2018). An intelligent radio regulatory architecture based on edge computing. China Radio, 12, 53–54.

Ioffe, S., and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521, 436–444.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceeding of IEEE, 86, 2278–2324.

Lu, Q. N., Yang, J. J., Jin, Z. Y., Chen, D. Z., & Huang, M. (2017). State of the art and challenges of radio spectrum monitoring in China. Radio Science, 52, 1–7. https://doi.org/10.1002/2017RS006409

Luo, Y., Liu, Y., & Zhang, Y. (2018). Speech bottleneck feature extraction method based on overlapping group lasso sparse deep neural network. Speech Communication, 99, 56–61.

Nair, V., and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In International Conference on Machine Learning.

Rajendran, S., Meert, W., Giustiniano, D., Lenders, V., & Pollin, S. (2018). Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. IEEE Transactions on Cognitive Communications and Networking, 4, 433–445.

Torres-Carassquilo, P. A., Singer, E., Kohler, M. A., Greene, R. J., Reynolds, D. A., and Deller, Jr. J. R. (2002). Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. In Proc. ICSLP, 899–921.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.

Zhang, S., Chen, S., Yang, L., Zhu, J., Luo, Z., & Hu, J. (2018). Big data processing architecture for radio signals empowered by deep learning: concept, experiment, applications and challenges. IEEE Access, 6, 55907–55922.

Zheng, F., Zhang, G., & Song, Z. (2001). Comparison of different implementations of MFCC. Journal of Computer Science and Technology, 16, 582–589.

10.1029/2019RS006885