CNN for Analyzing UAV Sound

INDEX

01 Introduction
- Subject
- Goal

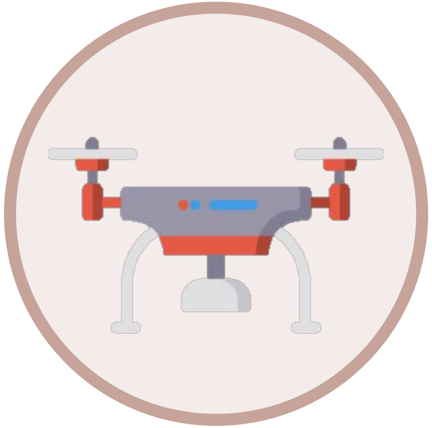
02 Process
- Data Preprocessing
- CNN
- Learning Model
- Training & Testing

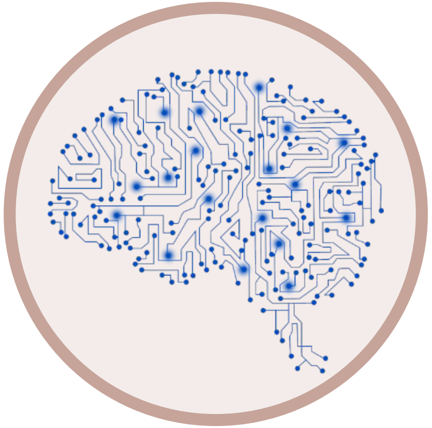
03 Result
- Prediction & Realization

04 Conclusion

Introduction

01 Introduction
Main Subject

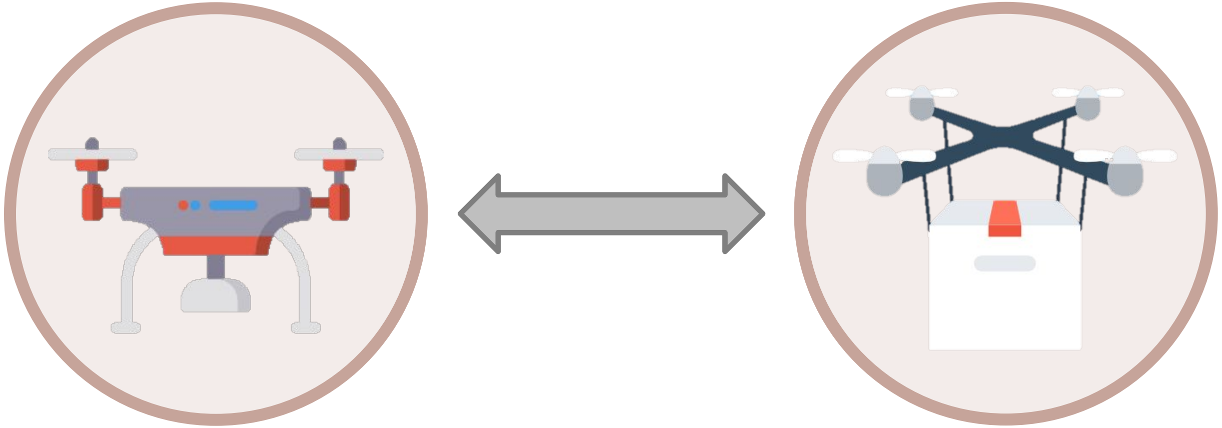




Analysis and utilization of drone sounds

01 Introduction

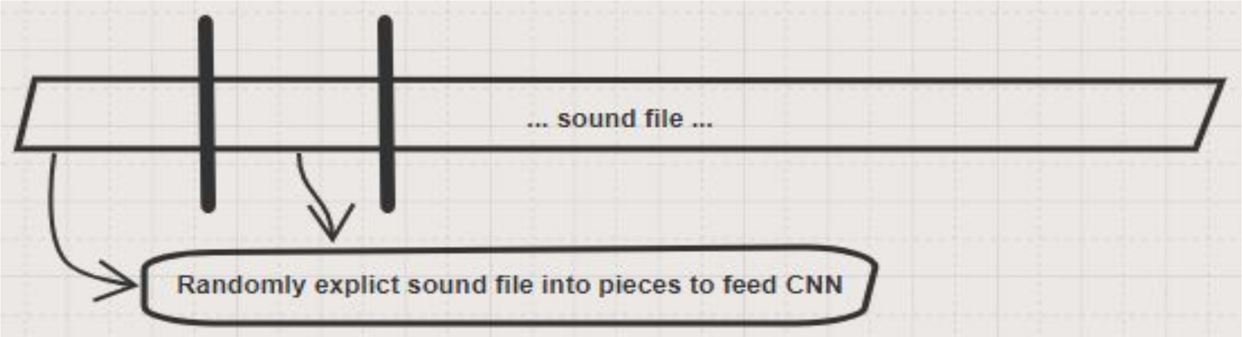
Our Goal



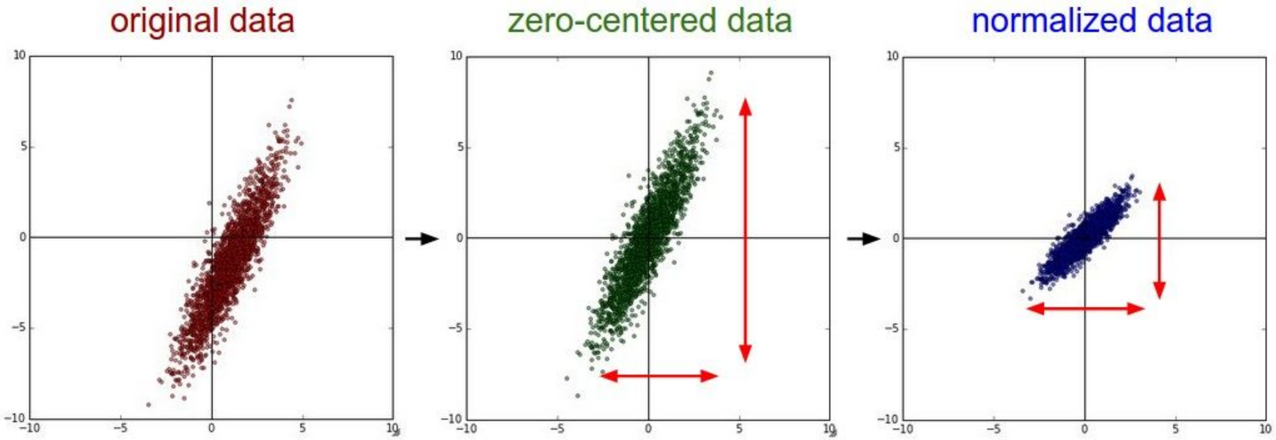
Distinguish unloaded drone and loaded drone

Process

02 Process
Data Preprocessing



Get samples



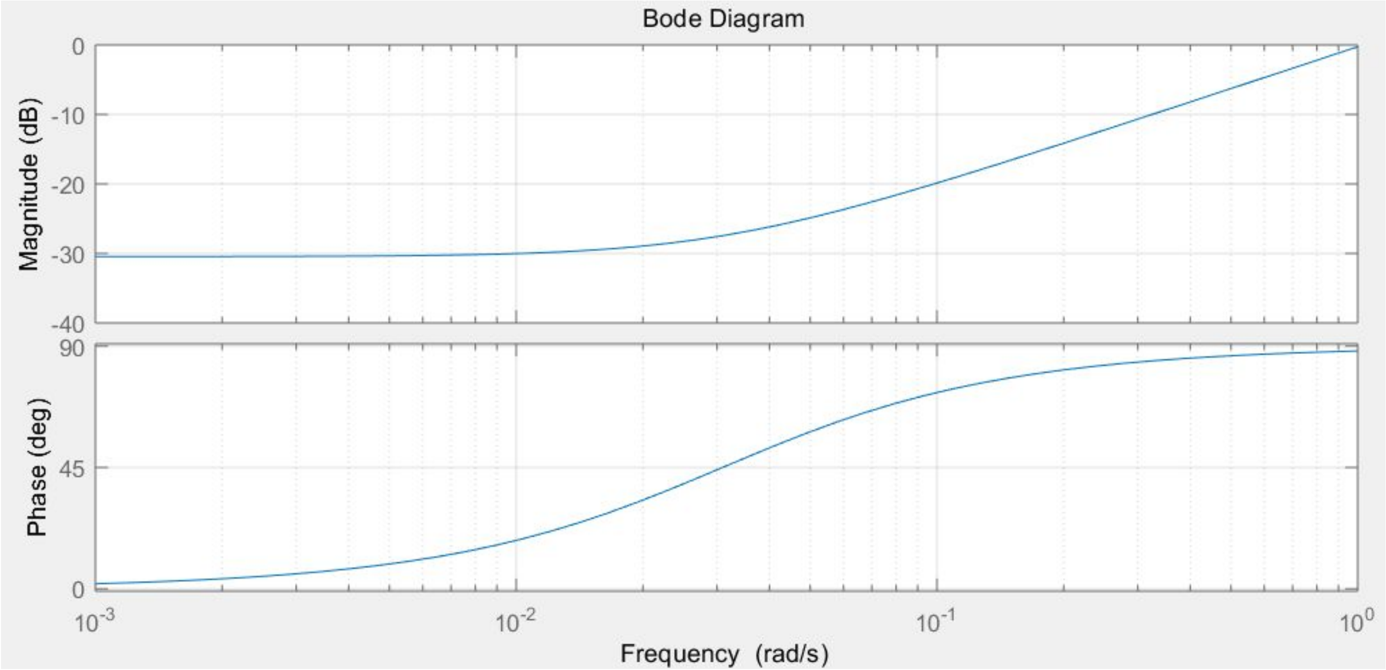
Normalization

Data Preprocessing

Pre-emphasis Filter

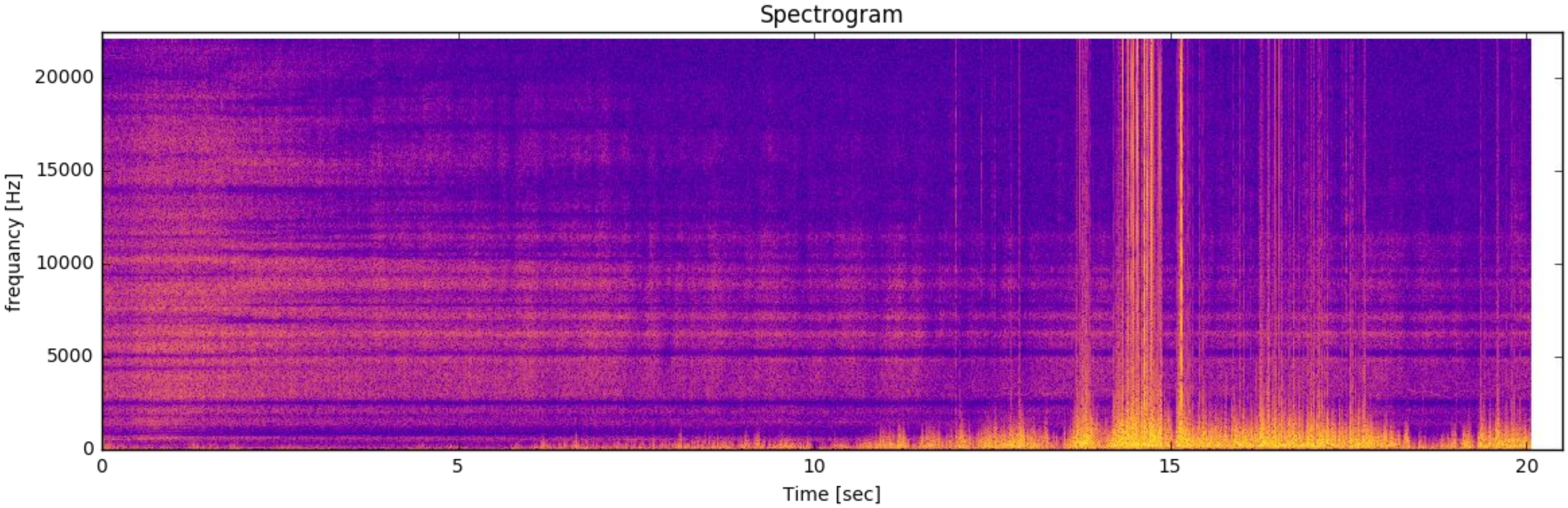
$$y[n] = x[n] - \alpha x[n-1]$$
$$y[0] = x[0] - \alpha$$

Pre-emphasis Filter

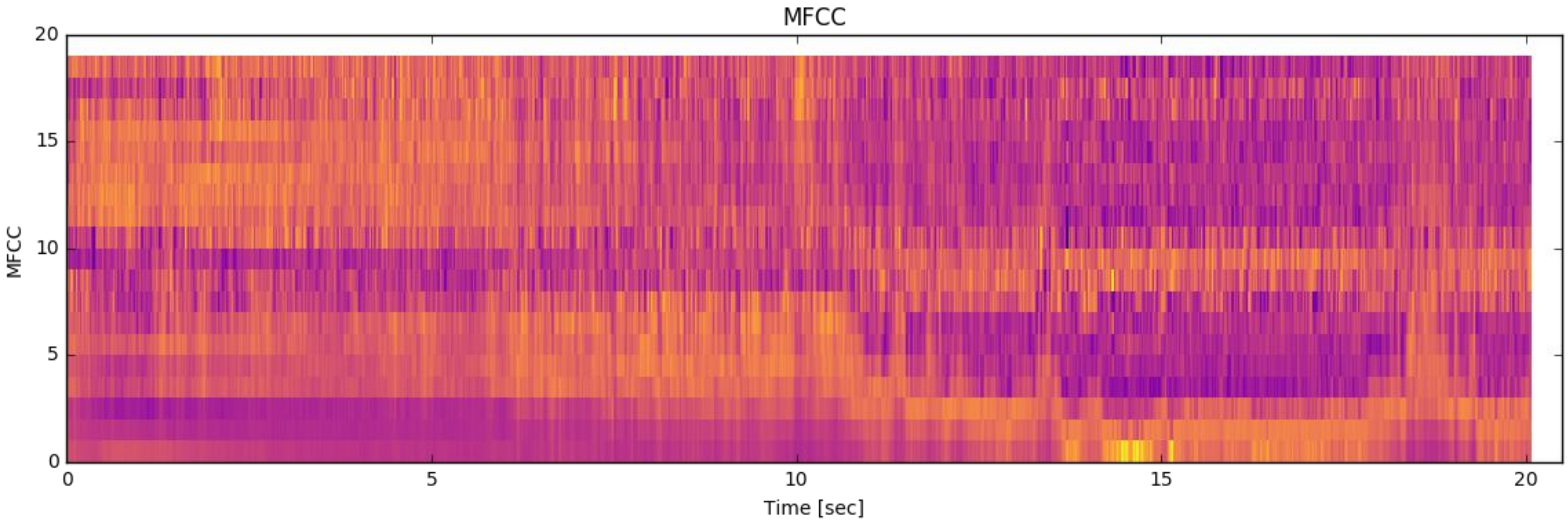


Data Preprocessing

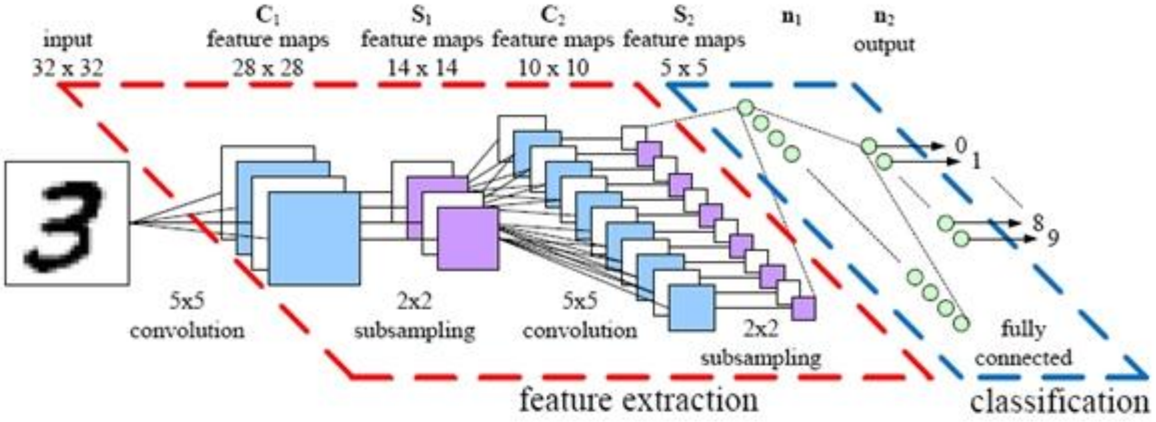
Spectrogram using STFT(Short Time Fourier Transform)



MFCC (Mel-Frequency Cepstral Coefficient)



CNN (Convolutional Neural Network)



- Deep learning
- Use to recognize images

02 Process
CNN

Language

TensorFlow

02 Process

CNN

```
mfccs = mfcc(y=emphasized_signal, sr=fs, S=None, n_mfcc=n_mfcc)
mfccs = sklearn.preprocessing.scale(mfccs, axis=1)

# print(S_norm.shape)
# data_set.append(S_norm.reshape(-1))
test_set = np.vstack((test_set, mfccs.reshape(-1)))
```

- PreProcessing (MFCC)

```
b3 = tf.Variable(tf.random_normal([100]))
L3 = tf.matmul(L2_flat, W3) + b3
L3 = tf.nn.relu(L3)
L3 = tf.nn.dropout(L3, keep_prob=keep_prob)
```

- Tensorflow(CNN)

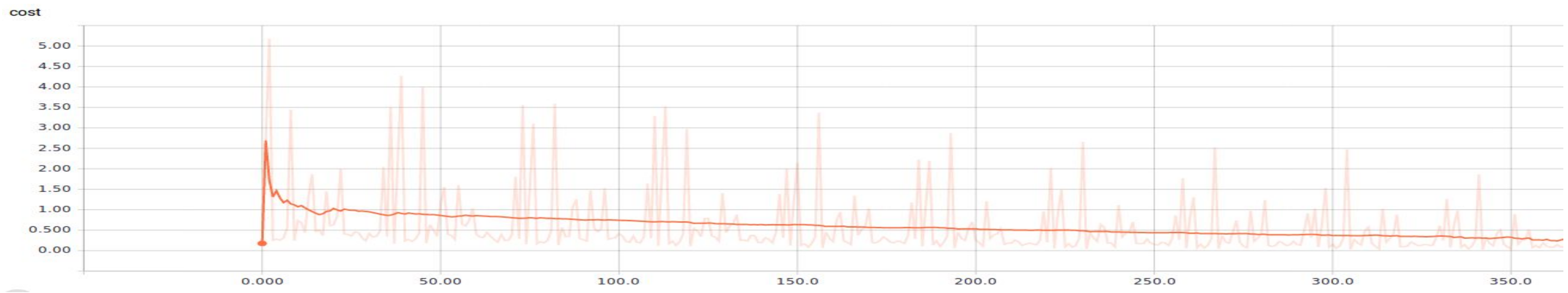
02 Process
Learning Model

1_ Convolutional	window size: 2x4x1 number of filter : 32
1_ Relu	activation function
1_ MaxPool	kernel size : [1,2,4,1] stride : [1,2,4,1]
2_ Convolutional	window size: 2x9x32 number of filter : 64
2_ Relu	activation function
2_ MaxPool	kernel size : [1,1,2,1] stride : [1,1,2,1]

3_ Fully Connected Neural Network	input size: 24192 output size : 100
3_Relu	activation function
3_ DropOut	keeping rate : 70%
4_ Fully Connected Neural Network	input size : 100 output size : 3

The layer of CNN

02 Process
Training & Testing



Result

03 Result
Prediction & Realization

result real data	noise	unloaded	loaded
noise	46766	60	174
unloaded	0	5000	0
loaded	3	0	3997

Accuracy : 99.5%
negative error : 0.005%

Conclusion

Conclusion

In conclusion

- UAV sound can be distinguished through data preprocessing with existing CNN model

- confirmed that the derivation of negative errors can be solved by weighting

[1] Seung Jin Shin, 2016, “Possibility of logistics delivery through drone (drone) and overseas case," Monthly KOTI Magazine on Transport, pp. 45~53.
[2] Dong-Hyun Lee, 2017, "Convolutional Neural Network-based Real-Time Drone Detection Algorithm," The Journal of Korea Robotics Society, Vol. 12, No. 4, pp. 425~431.
[3] Kee-Woong Lee, Kyoung-Min Song, Jung-Hwan Song, Chul-Ho Jung, Woo-kyung Lee, Myeong-Jin Lee and Yong-Kyu Song, 2017, "Implementation of Radar Drone Detection Based on ISAR Technique," The Journal of Korean Institute of Electromagnetic Engineering and Science, Vol. 28, No. 2, pp. 159~162.

Resource

[4] Lin Shi, Yujing He and KyungHi Chang, 2016, "Acoustic-based Classification for Drone Identification using Hidden Markov Model with MFCC Technique," Proceedings of Symposium of the Korean Institute of communications and Information Sciences, , pp. 204~205.
[5] SangGi Hong, GaYoung Kim, HeonGyeom Kim, SangHee Lee, DongHyun Lim, Austin Snair, Lucy Gotwals and Jone C. Gallagher, 2017, "Practically Classifying Unmanned Aerial Vehicles Sound using Convolutional Neural Networks," Proceedings of the Korean Information Science Society, pp. 1856~1858.

Thank you