Report Date: 07/22/2022
To: ematson@purdue.edu, ahsmith@purdue.edu, lhiday@purdue.edu, and lee3450@purdue.edu
From: What is today's lunch?

- Ilmun Ku(mun90505@hufs.ac.kr)
- Seungyeon Roh(shtmddus99@konkuk.ac.kr)
- Gyeongyeong Kim(kky57389@sunmoon.ac.kr)
- Charles Taylor(taylo869@purdue.edu)

Summary

The performance of the Deep Learning(DL) model improved continuously with each data collection trip. The techniques of Deep Learning we utilized were Convolutional Neural Networks(CNN), Recurrent Neural Networks(RNN), and Convolutional Recurrent Neural Networks(CRNN). Among these techniques, CRNN with Mel-frequency Cepstral Coefficient(MFCC) produced the best result.
Also, a methodology draft has been written including the data augmentation part. In the data augmentation section, raw augmentation, including time stretching and pitch scaling, and spectrogram augmentation including time masking and frequency masking were explained.

What 'What is today's lunch?' completed this week:

- **Deep Learning Performance**

  ■ The features of MFCC and Mel produce reasonable results rather than chroma, contrast, and tonnetz. CNN and RNN showed similar performances and CRNN has achieved the best accuracy among CNN, RNN, and CRNN.

4ClassClassification	CRNN	CNN	RNN
MFCC	90.14%	87.16%	87.6%
Mel	91.73%	87.86%	88.23%
Chroma	69.26%	64.19%	68.84%
Contrast	69.52%	65.53%	70.47%
Tonnetz	56.45%	54.17%	50.34%

- **Data Collection Trip**

  ■ 531 audio samples were collected for four classes including noise, unloaded, 1 payload, and 2 payloads.
  ■ The weather condition is the same as the figure below.

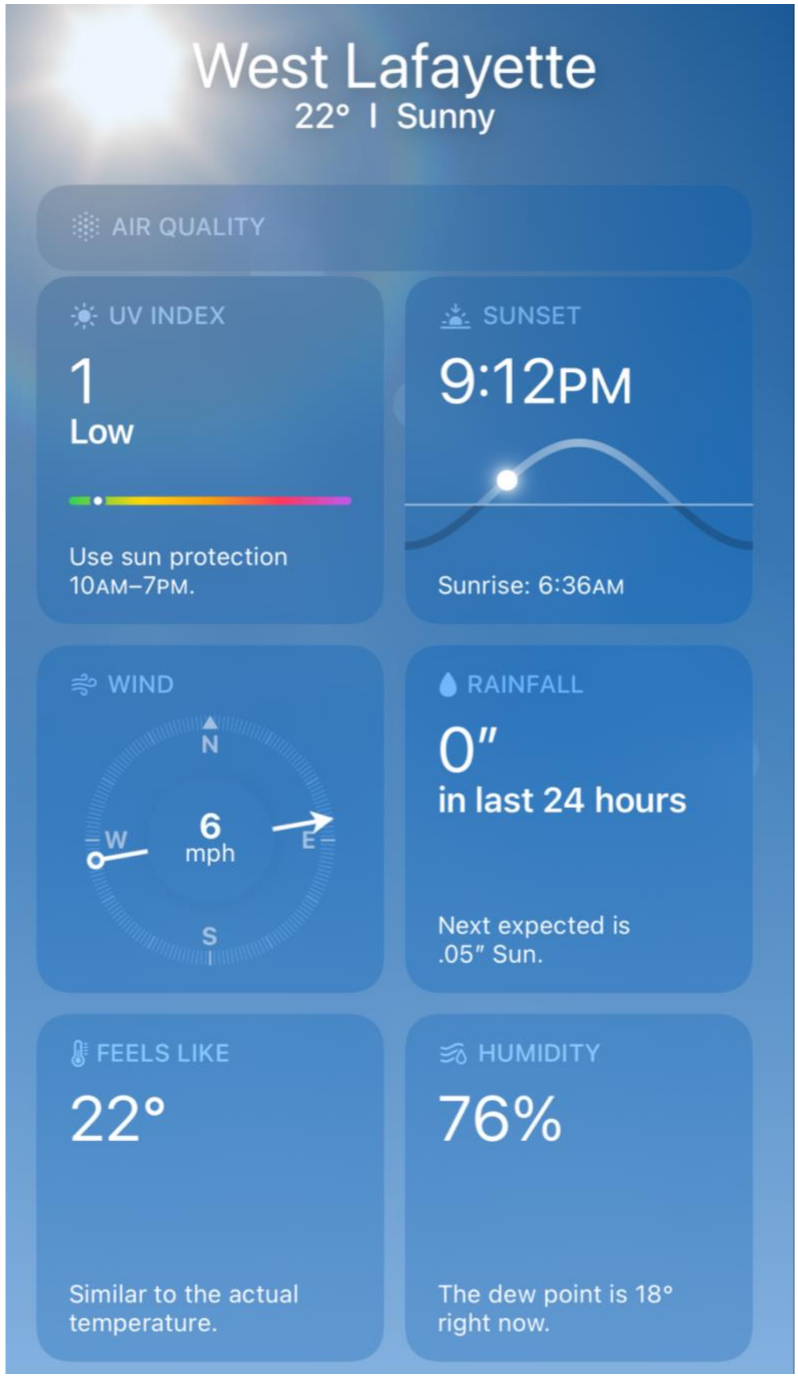

Figure 1

- **Methodology draft**

  - Data augmentation

    A large audio dataset is required to utilize DL models to solve audio classification problems. However, it is difficult to collect enough audio datasets for DL rather than text datasets or image datasets. If datasets are insufficient, overfitting and poor generalization may occur in the classification process. Data augmentation technology is one of the solutions to cover the lack of a dataset problem. In this method, a new dataset can be obtained by augmenting the original dataset. Therefore, Data augmentation methods are pivotal for the smooth and continuous improvement of audio classification performance while utilizing these DL algorithms.

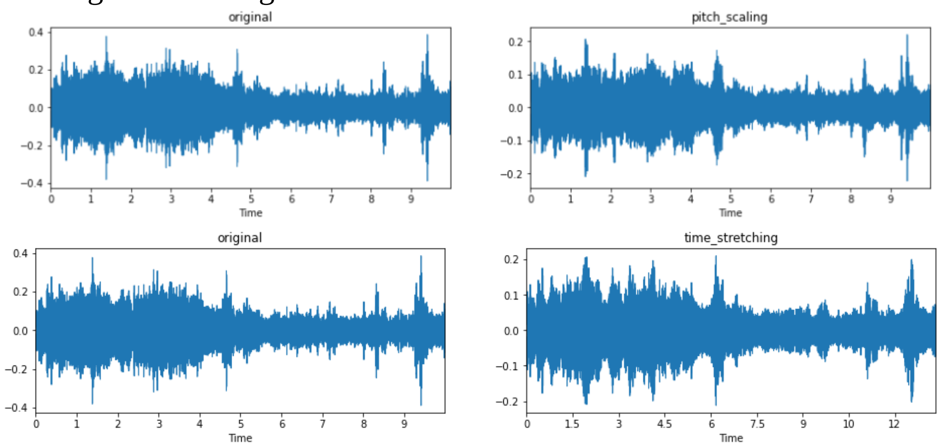

Figure 2. Raw Data Augmentation

In this study, time-stretching, pitch scaling, time masking, and frequency masking are exercised to make augmentation for audio data. Time stretching and Pitch scaling are raw audio augmentation technology whereas time masking and frequency masking is spectrogram augmentation technology that treats augmentation as a visual problem, rather than an audio problem.

Raw augmentation is a method of augmenting the audio file itself, and there is time shifting, time stretching, pitch scaling, noise addition, impulse response addition, low/high/pass-band filters, and so on. Time stretching changes the speed without changing the pitch, which can change the speed of the sound slow or fast, not impacting the frequency. It is proven to be useful in enhancing accuracy for LSTM-based RNN using raw audio data [14]. As opposed to time stretching, pitch scaling is changing the pitch without changing the speed. For example, C major is changed to D major if a signal is up to 2. It is recognized to be advantageous when exercised in advancing CNN accuracy [15].

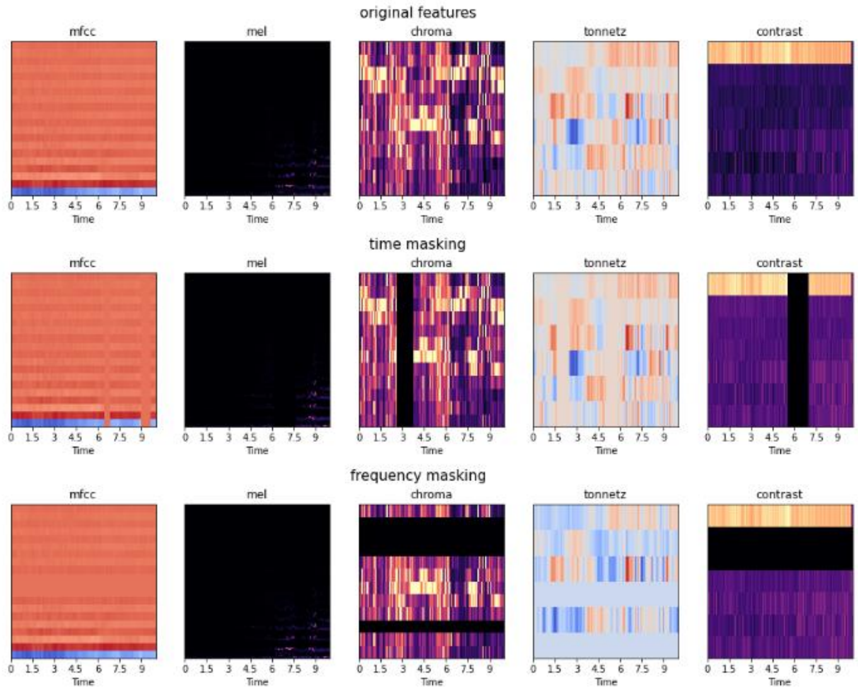

Figure 3. Spectral Data Augmentation

Spectrogram augmentation is data augmentation with a spectrum rather than a raw audio file, including time masking and frequency masking. Time masking masks a certain part of the spectrum, and those are masked with 0 or minimum value. It cuts off a portion of the time domain, which is the x-axis of the spectrum. Frequency masking is an inverse version of time masking, in which cut off a part of the frequency domain of a spectrogram by masking a certain part with 0 or minimum value. They are widely known to improve the network performance without the extra arrangement for the network or hyperparameter [16]. They help the network to be robust against deformation and prevent overfitting by presenting corrupted data on purpose.

- Augmented data to csv file

```
def save_mfcc(dataset_path, csv_path, num_segments=5):
    labelList = ["noise", "unloaded", "1_payload", "2_payloads"]
    hop_length = 512

    num_samples_per_segment = int(SAMPLES_PER_TRACK / num_segments)
    expected_num_mfcc_vectors_per_segment = math.ceil(num_samples_per_segment / hop_length )

    fieldnames = ['mfcc_timemasking', 'labels']
    csv_f = open(CSV_PATH, "w")
    writer = csv.DictWriter(csv_f, fieldnames=fieldnames)
    writer.writeheader()

    # loop through all labels
    for i, (dirpath, dirnames, filenames) in enumerate(os.walk(dataset_path)):
        if dirpath is not DATASET_PATH:
            dirpath_components = dirpath.split("/")
            semantic_label = dirpath_components[-1]
            dirpath_part = dirpath.split('/')
            label_idx = 0
            current_label = ""
            for j in labelList:
                if j in dirpath_part:
                    label_idx = labelList.index(j)
                    current_label = j
            print("₩nProcessing{}, current_label:{}" .format(semantic_label,current_label))
            print(f'dirpath: {dirpath}, dirnames:{dirnames}, label_idx:{label_idx}, current_label:{current_label}')

            for f in filenames:
                if ".csv" in f:
                    continue
                if "payload" in semantic_label:
                    if "2022_06_08" in f:
                        continue
                    if "2022_06_16" in f:
                        continue
                    if "2022_06_28" in f:
                        continue

                file_path = os.path.join(dirpath, f)
                signal, sr = librosa.load(file_path, sr = SAMPLE_RATE)
                for s in range(num_segments):
                    start_sample = num_samples_per_segment * s # s = 0 -> 0
                    finish_sample = start_sample + num_samples_per_segment # s=0 -> num_samples_per_segment

                    mfcc = librosa.feature.mfcc(signal[start_sample:finish_sample],
                                                sr= sr,
                                                hop_length = hop_length,
                                                n_mfcc=40
                                                )
                    mfcc = time_augment(mfcc)
                    mfcc = mfcc.T

                    if len(mfcc) == expected_num_mfcc_vectors_per_segment:
                        print("{}, segment:{}, dirnames:{}, label_idx:{}, current_label:{}, data[labels]:{}" .format(file_path, s+1, dirnames, label_idx, current_label, i ))
                        rowdata = {"mfcc_timemasking": mfcc.tolist(), "labels": label_idx}
                        writer.writerow(rowdata)
    csv_f.close()
```

Figure 4. Feature extraction and Data Augmentation code

After Raw data augmentation and Spectrogram augmentation, it is required to save as csv file of augmented datasets.

Things to do by next week

- complete methodology and experiment section draft.

Problems or challenges:

- Out of memory when data augmented is uploaded on RAM to feed DL models.

References

[1] M. al Kibsi, “Houthi drone targets senior yemeni officers, kills five soldiers,” Available at https://www.aljazeera.com/news/2019/1/10/houthi-drone-targets-senior-yemeni-officers-kills-five=soldiers (2022/05/18).

[2] W. Ripley, “Drone with radioactive material found on japanese prime minister’s roof,” Available at

https://www.cnn.com/2015/04/22/asia/japan-prime-minister-rooftop-drone/index.html (2022/05/18).
[3] E. Kelly, “Venezuela drone attack: Here's what happened with Nicolas Maduro,” Available at https://www.usatoday.com/story/news/politics/2018/08/06/venezuela-drone-attack-nicolas-maduro-assassination-attempt-what-happened/913096002 (2022/05/18).
[4] Y. Wang, F. E. Fagiani, K. E. Ho, and E. T. Matson, “A feature engineering focused system for acoustic uav payload detection,” the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022), pp. 470–475, 2022.
[5] S. Wei, S. Zou, F. Liao et al., “A comparison on data augmentation methods based on deep learning for audio classification,” 2019 2nd International Conference on Computer Information Science and Artificial Intelligence(CISAI 2019), pp. 1–8, 2020.
[6] S. Al-Emadi, A. Al-Ali, A. Mohammad, and A. Al-Ali, “Audio based drone detection and identification using deep learning,” pp. 459–464, 2019.
[7] S. Jamil, M. Rahman, A. Ullah, S. Badnava, M. Forsat, S. S. Mirjavadi et al., “Malicious uav detection using integrated audio and visual features for public safety applications,” Sensors, vol. 20, no. 14, p. 3923, 2020.
[8] Y. Wang, F. E. Fagian, K. E. Ho, and E. T. Matson, “A feature engineering focused system for acoustic uav detection,” pp. 125–130, 2021.
[9] L. Shi, I. Ahmad, Y. He, and K. Chang, “Hidden markov model based drone sound recognition using mfcc technique in practical noisy environments,” Journal of Communications and Networks, vol. 20, no. 5, pp. 509–518, 2018.
[10] V. Kartashov, V. Oleynikov, I. Koryttsev, S. Sheiko, O. Zubkov, S. Babkin, and I. Selieznov, “Use of acoustic signature for detection, recognition and direction finding of small unmanned aerial vehicles,” pp. 1–4, 2020.
[11] U. Seidaliyeva, M. Alduraibi, L. Ilipbayeva, and A. Almagambetov, “Detection of loaded and unloaded uav using deep neural network,” pp. 490–494, 2020.