Report Date: 07/01/2022
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)



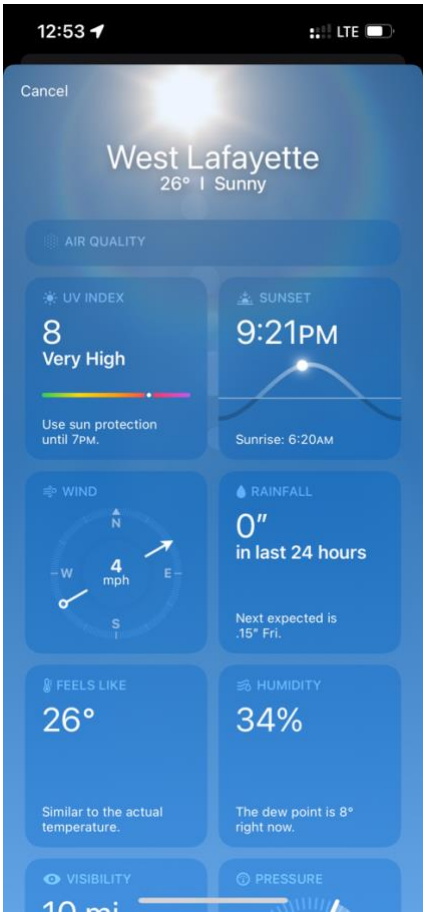
**Summary**

Data collection has been conducted this Tuesday(28th of June) and Thursday(30th of June). Feature extraction including Mel Frequency Cepstral Coefficient(MFCC), mel, chroma, contrast, tonnetz was conducted using data recently collected.

Also, Machine Learning utilizing k-fold cross-validation was carried out to inspect model overfitting. To prevent overfitting, kaiming (He) initialization is adopted to check if the model is effective. Furthermore, raw data augmentation technology has been applied the first time.


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

- Data collection trip(0628) utilizing DJI Phantom 4 and EVO 2 Pro
  - Weather




  - Picture of Unmanned Aerial Vehicle(UAV), EVO 2 Pro.

- Unloaded

- 1 payload

- 2 payloads

- **Feature extraction with new dataset**
  - MFCC, Mel, Chroma, Contrast, Tonnetz were extracted from collected data.
    - MFCC: https://drive.google.com/file/d/1-2DWAJbeydjdvZiZoG_8SiCGkumyHSn9/view?usp=sharing
    - Mel: https://drive.google.com/file/d/1-2DWAJbeydjdvZiZoG_8SiCGkumyHSn9/view?usp=sharing
    - Contrast: https://drive.google.com/file/d/1-NLNLuYxMM2TfIg5761T7XoOCKOjzW-G/view?usp=sharing
    - Tonnetz: https://drive.google.com/file/d/1-J_FW2pa6IJdY2nal4GyRcbmE1DnIQtG/view?usp=sharing
    - Chroma: https://drive.google.com/file/d/1-HG4Jg5axT9sv0ztg5c9VGB8QoNH9jSP/view?usp=sharing

- **Machine Learning with k fold cross validation**
  - MFCCs

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
import sklearn.svm as svm
from sklearn.model_selection import cross_val_score

models=[]
models.append(("KNN",KNeighborsClassifier(n_neighbors=3)))
models.append(("NB",GaussianNB()))
models.append(("SVM",svm.SVC(kernel = 'linear', random_state=100)))

for name,model in models:
    cv_result = cross_val_score(model,mfccX, mfccy, cv = 4,scoring = "accuracy")
    print(name, cv_result.mean())

```

```
KNN 0.9945163389242336
NB 0.9939668883747831
SVM 0.9989035087719298
```

  - Mel

```
for name,model in models:
    cv_result = cross_val_score(model,melX, mely, cv = 4,scoring = "accuracy")
    print(name, cv_result.mean())
```

```
KNN 0.8875722961249277
NB 0.639078465394255
SVM 0.7093008964719492
```

  - Chroma

```
for name,model in models:
    cv_result = cross_val_score(model,chromaX, chromay, cv = 4,scoring = "accuracy")
    print(name, cv_result.mean())
```

```
KNN 0.9473539618276461
NB 0.8974551764025449
SVM 0.9391242529400424
```

- Contrast

```
for name,model in models:
    cv_result = cross_val_score(model,contrastX, contrasty, cv = 4,scoring = "accuracy")
    print(name, cv_result.mean())
```

```
KNN 0.826649556583767
NB 0.7953718430692115
SVM 0.8332345286292655
```

- Tonnetz

```
for name,model in models:
    cv_result = cross_val_score(model,tonnetzX, tonnetzy, cv = 4,scoring = "accuracy")
    print(name, cv_result.mean())
```

```
KNN 0.577121891266628
NB 0.5266555812608444
SVM 0.48880253518411415
```

- Combination of individual features

```
for name,model in models:
    cv_result = cross_val_score(model,combiX, combiy, cv = 4,scoring = "accuracy")
    print(name, cv_result.mean())
```

```
KNN 0.9939668883747831
NB 0.9155605359552729
SVM 0.9994517543859649
```

K fold cross-validation on Machine Learning was conducted. Individual features (MFCCs, Mel, Chroma, Contrast, Tonnetz) and a combination of these features were used.

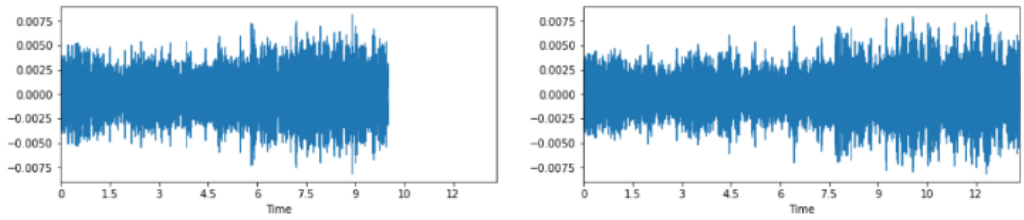
- Raw Audio Augmentation
  - To train deep learning based on acoustic datasets, raw audio augmentation and spectrogram augmentation will be utilized. Among these two methods, raw audio augmentation was tested first.

```python
def load_file(file_name):
    audio, sample_rate=librosa.load(file_name, res_type='kaiser_fast')
    return audio, sample_rate

def pitch_scaling(file_name):
    audio, sample_rate = librosa.load(file_name, res_type='kaiser_fast')
    return librosa.effects.pitch_shift(audio, sample_rate, n_steps=2)

def time_stretching(file_name):
    audio, sample_rate = librosa.load(file_name, res_type='kaiser_fast')
    return librosa.effects.time_stretch(audio, 0.75)
```

```python
pitch = []
time = []

# Iterate through each sound file
for index, row in df.iterrows():
    file_name = str(Path.cwd())+df['relative_path'][index]
    class_label = row["class2"]
    p_scaling = pitch_scaling(file_name)
    t_stretching = time_stretching(file_name)
    pitch.append([p_scaling, class_label])
    time.append([t_stretching, class_label])

# Convert into a Panda dataframe
pitch_scaling = pd.DataFrame(pitch, columns=['x','class_label'])
time_stretching = pd.DataFrame(time, columns=['x','class_label'])
```

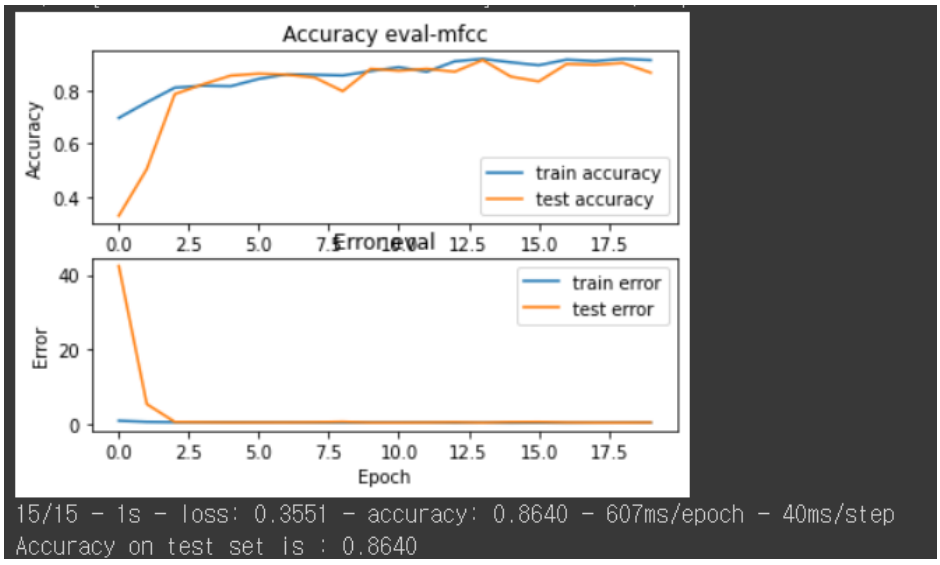
```python
fig, ax = plt.subplots(nrows=1, ncols=2, sharex=True, figsize=(15, 3))

librosa.display.waveplot(pitch_scaling.x[0], x_axis='time', ax=ax[0])
librosa.display.waveplot(time_stretching.x[0], x_axis='time', ax=ax[1])
```

```
<matplotlib.collections.PolyCollection at 0x7f8bb2e1a110>
```




■

- Weight initialization is introduced to make a stable training
  - Weight initialization can help prevent model's overfitting/underfitting problems. In this study, Kaiming (He) initialization is adopted.



**Things to do by next week**

- Going on a data collection trip with Mia.

**Problems or challenges:**

- Further Data collection would be conducted in the future.
- To check if it is possible to apply SpecAugment to raw audio data.

**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.
[12] He, K., Zhang, X., Ren, S., & Sun, J. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.” In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034, 2015.