Report Date: 06/24/2022
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

Feature extraction using Librosa was conducted to utilize Machine Learning models to classify payload. The new datasets collected in the data collection trip were used to extract features. Also, Machine Learning with prior dataset was conducted. NN(Neural Network), KNN(K Nearest Neighborhood), GNB(Gaussian Naïve Bayes), and SVM(Support Vector Machine) were utilized.

Deep Learning(DL) including Convolutional Neural Network(CNN) and RNN(Recurrent Neural Network) was employed as well. Furthermore, the combination of five features was applied in CNN and RNN models using data from a prior study.

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

- **Feature extraction with new dataset**
  - Feature extraction (MFCCs, Mel, Chroma, Tonnetz, Contrast)

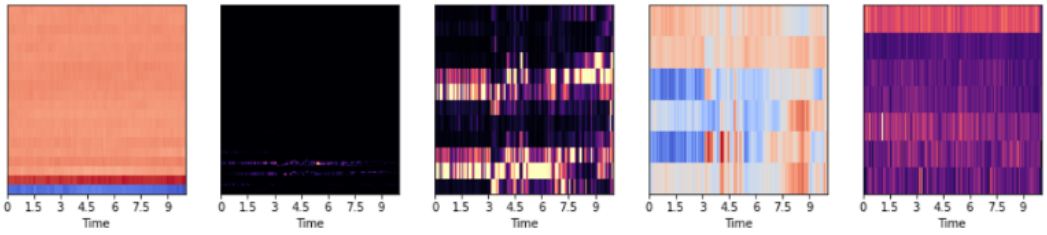


Figure 1. Feature extraction (2 payload)

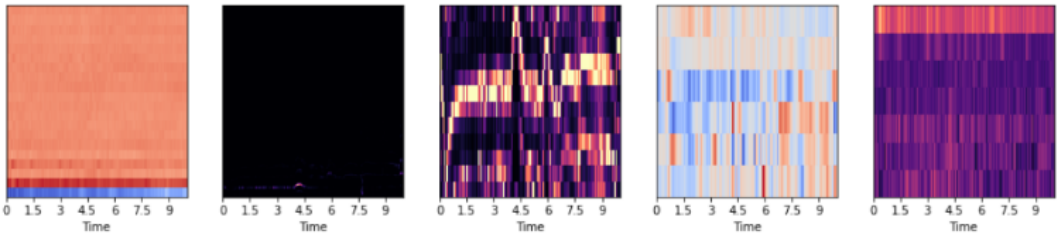


Figure 2. Feature extraction (1 payload)

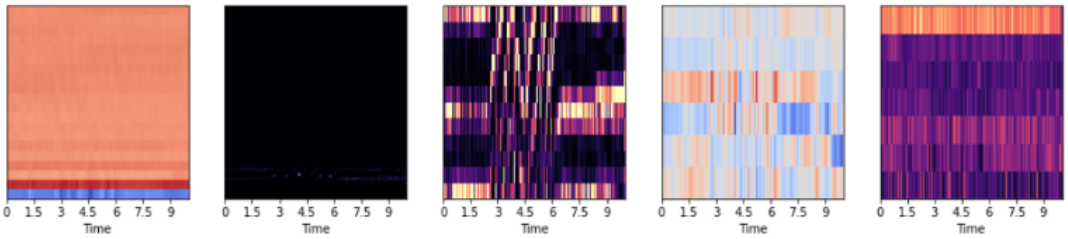


Figure 3. Feature extraction (0 payload)

Feature extraction with new datasets (2payload, 1payload, without payload) was conducted. MFCCs, Mel, Chroma, Tonnetz, and Contrast will be utilized to train and test Machine Learning models.

- Machine Learning with prior dataset
  - NN, KNN, GNB, SVM with MFCC
    - NN

```
# split the dataset
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, yy, test_size=0.2, random_state = 127)
```

```
# neural network architecture
num_labels = yy.shape[1]
filter_size = 2

model = Sequential()
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(num_labels))
model.add(Activation('softmax'))
# Compile the model
model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer='adam')

model.build(input_shape=(None, 40))
# Display model architecture summary
model.summary()

```

```
from keras.callbacks import ModelCheckpoint
from datetime import datetime
num_epochs = 100
num_batch_size = 32

history=model.fit(x_train, y_train, batch_size=num_batch_size, epochs=num_epochs, validation_data=(x_test, y_test), verbose=1)
```

```
# Evaluating the model on the training and testing set
score = model.evaluate(x_train, y_train, verbose=0)
print("Training Accuracy: {0:.2%}".format(score[1]))
score = model.evaluate(x_test, y_test, verbose=0)
print("Testing Accuracy: {0:.2%}".format(score[1]))
```

```
Training Accuracy: 100.00%
Testing Accuracy: 100.00%
```

◆ KNN (99%)

```
from sklearn.model_selection import train_test_split
training_data, validation_data , training_labels, validation_labels = train_test_split(X, yy, test_size = 0.2, random_state = 100)
```

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 3)
```

```
classifier.fit(training_data, training_labels)
```

```
KNeighborsClassifier(n_neighbors=3)
```

```
print(classifier.score(validation_data, validation_labels))
```

```
0.9972602739726028
```

◆ GNB (100%)

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train, y_train)
```

```
GaussianNB()
```

```
print("accuracy: ",nb.score(x_test, y_test))
```

```
accuracy:  1.0
```

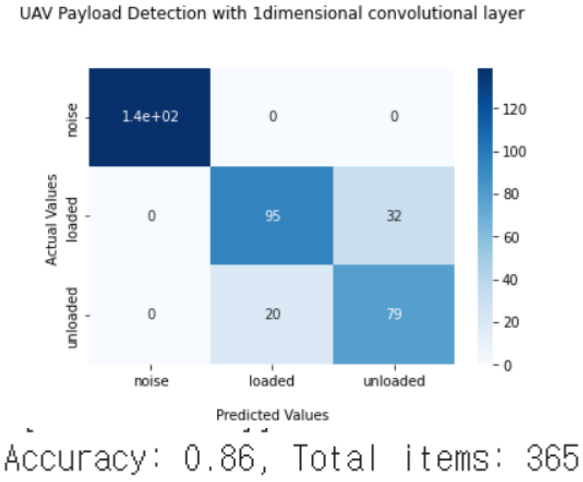
◆ SVM (linear kernel: 99%, rbf kernel : 98%)

```
# Data split

X_train, X_test, y_train, y_test = train_test_split(X, yy, test_size = 0.3, random_state = 100)
import sklearn.svm as svm
import sklearn.metrics as mt
from sklearn.model_selection import cross_val_score, cross_validate

# SVM, kernel = 'linear'
svm_clf =svm.SVC(kernel = 'linear', random_state=100)

# cross validation
scores = cross_val_score(svm_clf, X, y, cv = 5)
scores

pd.DataFrame(cross_validate(svm_clf, X, y, cv =5))

print('cross validation average(linear kernel): ', scores.mean())

# SVM, kernel = 'rbf'
svm_clf =svm.SVC(kernel = 'rbf')

# cross validation
scores = cross_val_score(svm_clf, X, y, cv = 5)
scores

pd.DataFrame(cross_validate(svm_clf, X, y, cv =5))

print('cross validation average(rbf kernel): ', scores.mean())
```

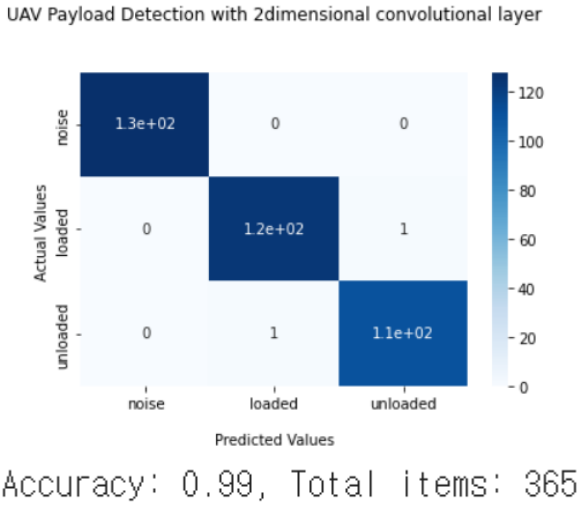
```
cross validation average(linear kernel):  0.9928767123287671
cross validation average(rbf kernel):  0.9873957549300016
```

- Deep Learning with prior dataset

- CNN using 1dimensional convolutional layer and MFCCs (86%)
  https://colab.research.google.com/drive/1uWFH6BrzBMzT3XElr84_p37rV4lXCq2e?usp=sharing



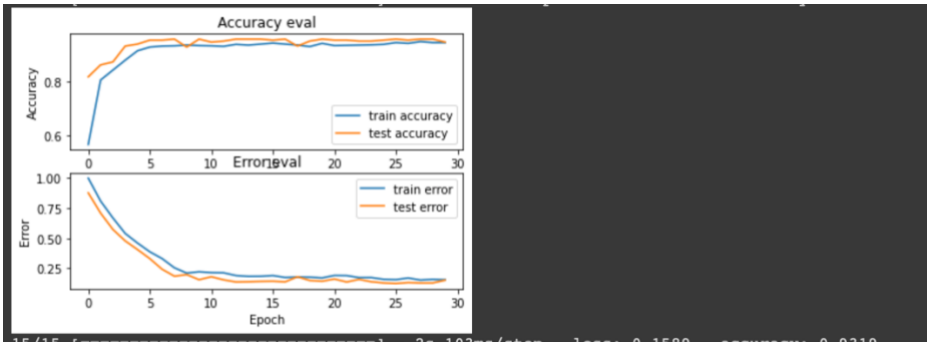
- CNN using 2dimensional convolutional layer and MFCCs (99%)
  https://colab.research.google.com/drive/1uWFH6BrzBMzT3XElr84_p37rV4lXCq2e?usp=sharing



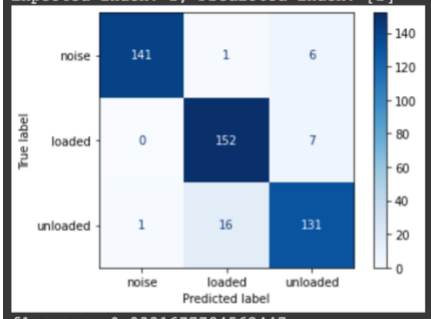
- RNN with a combination of features. (93.1%)

Each feature was extracted with the same length and different parts from raw data.

```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_16 (LSTM)              (None, 431, 64)           18176

 lstm_17 (LSTM)              (None, 64)                33024

 dense_16 (Dense)            (None, 64)                4160

 dropout_8 (Dropout)         (None, 64)                0

 dense_17 (Dense)            (None, 3)                 195

=================================================================
Total params: 55,555
Trainable params: 55,555
Non-trainable params: 0
_________________________________________________________________
```




```
15/15 [==============================] - 2s 103ms/step - loss: 0.1589 - accuracy: 0.9319
Accuracy on test set is : 0.9318681359291077
Expected index: 2, Predicted index: [2]
```




```
f1 score: 0.9321677784562447
precision_score: 0.9340294175749214
recall_score: 0.9312708935350446
```

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.

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.