Intoxicated Speech Detection for Korean Language

SeokJun Kim 20160793 | **Jeongeon Park** 20160811 | **Sujin Han** 20170841

Github Repository: `https://github.com/vilotgit/kaisd`

1 Introduction

Drunk driving is a serious problem in Korea. In 2019, 15,708 traffic accidents were caused by drunk driving [7], and in 2018, police reported that the number of drunk driving cases averaged 334.2 per day [6]. Similar problem also prevails in KAIST, where a student on their motorcycle was drunk driving and bumped into other students in 2016 [9].

Some methods such as Breathalyzer and Driving Under the Influence (DUI) blood testing are used to measure the drivers' Blood Alcohol Content (BAC) and prevent people from driving under the influence. However, there exists limitations to these widely used methods. Breathalyzer are found to be very sensitive to temperature, and can register alcohol from the mouth, throat, or stomach, possibly leading to a false BAC level detection. DUI blood testing is often difficult to be performed at the right moment due to physical limitation. Moreover, both these methods are not always accessible in our daily lives. Thus, we thought that audio detection of the DUI would be a more accessible and accurate solution.

Several work has been done as a result of the INTERSPEECH 2011 Speaker State Challenge Intoxication Sub-Challenge [11], to detect whether the person speaking is intoxicated or not. However, the challenge uses a German corpora, which has different grammatical and phonetic structure from Korean language.

In this report, we present **KAISD** (**K**orean language **A**lcohol **I**ntoxicated **S**peech **D**etector), a set of binary classifiers that determines whether a speaker is intoxicated or not in Korean speech. `KAISD` consists of three models, `KAISD_naive`, `KAISD_pretrained`, and `KAISD_inception`, and those models were compared in terms of their F1 scores. With the validation set, `KAISD_inception` performed the best with F1 score of 0.96, and with the held out data, `KAISD_pretrained` was the best with F1 score of 0.83.

With the model we created, we wish to decrease the rate of drink driving incidents in the Korean society. We believe that a more accessible method for intoxicated detection, through voice detection, can increase the awareness of drink driving.

2 Related Work

Many work on intoxicated speech detection [1, 3, 2] was carried out as a part of the INTERSPEECH 2011 Speaker State Challenge Intoxication Sub-Challenge [11]. The challenge corpora used was the Alcohol Language Corpus (ALC) [5] which is in German, and most of them used support vector machine (SVM) or Gaussian mixture model (GMM). To apply the concept we learned in class to the problem we are trying to tackle, we decided to use neural network for intoxicated speech detection.

AlcoAudio [10] also tried to detect alcohol speech using Neural Networks. However, ALC, a German language corpus, was used in both training and test set. We extend this work by using a self-collected Korean language corpus, and by classifying intoxicated speech in Korean language with various modifications such as adding of an inception module and hyper-parameter tuning.

Preprint. Under review.

3 Method

3.1 Data Collection

We collected **9 hours 34 minutes** worth of speech data from ourselves (3 authors) and 13 YouTube channels. The following two paragraphs explain how we collected data.

For our own data, we recorded our own speech in sober state and in intoxicated state. For sober speech, we recorded ourselves talking at a restaurant. For intoxicated speech, we recorded ourselves talking at a pub after having some drinks and confirming that our Blood Alcohol Concentration was above 0.08% with a Breathalyzer. We used our own phones to record. To make our own voice be the dominant sound in those recordings, we wore earphones and placed the microphones close to ourselves.

For YouTube data, we considered various factors and carefully selected 13 YouTube channels. The factors we considered were: (1) whether the YouTube channel contains video logs (Vlogs) of people in both sober state and in intoxicate state, (2) whether those Vlog audio is primarily comprised of speech and have minimal sound effect, (3) variety in background setting within collected data (e.g. drinking alcohol at a large pub and that in a small room). From each YouTube channel, we selected one video for sober recording and one video for intoxicated recording. Then, we extracted audio files from selected videos and edited out sound effects. Also, for intoxicated speech data, we only used recordings from parts where we thought the people appearing in the video are sufficiently drunk.

We left out and shortened some recordings to make our ratio of sober to intoxicated data one to one. At the end, we used around 9 hours of speech recordings. The collected data was shuffled and split into three sets - 80% training set, 10% test set, and 10% validation set. In addition, recordings from one YouTube channel were entirely left out from the mixture to be used in evaluation. Summary of data used for training and evaluation is shown in Figure 1.

	Length of Recording (H:MM:SS)		
Source	Intoxicated	Sober	Total
SeokJun	0:50:00	1:02:49	1:52:49
Jeongeon	0:46:12	0:41:27	1:27:39
Sujin	1:05:20	0:37:24	1:42:44
YouTube_1	0:06:22	0:18:00	0:24:22
YouTube_2	0:07:21	0:03:04	0:10:25
YouTube_3	0:03:28	0:08:40	0:12:08
YouTube_4	0:11:50	0:11:03	0:22:53
YouTube_5	0:14:16	0:11:16	0:25:32
YouTube_6	0:17:11	0:24:55	0:42:06
YouTube_7	0:18:29	0:11:33	0:30:02
YouTube_8	0:12:32	0:18:47	0:31:19
YouTube_9	0:06:38	0:13:56	0:20:34
YouTube_10	0:08:05	0:04:07	0:12:12
Overall except YouTube_11	4:27:44	4:27:01	8:54:45
YouTube_11	0:06:43	0:06:50	0:13:33
Overall	4:34:27	4:33:51	9:08:18

Figure 1: Summary of recordings used for training and evaluation. YouTube_11 was intentionally left out from the initial mix to perform evaluation on held out subject.

3.2 Preprocessing

Instead of feeding the naive audio file into the network, we converted the input audio files into **mel spectrogram** images. This was because our proposed models are highly based on convolutional (CNN) layers and thus, we thought that 2D image data will be the most suitable for training.

Split Audio Data Majority of our collected data were very long (> 1 hour). Feeding each of these data into the model would have resulted in a restricted receptive field and thus, inefficient training.

Therefore, we cut each of the audio input into 8 second segments. Then, we obtained waveform representations for each of the segments using the python's librosa[1] package.

Fourier Transform The resulting waveform captures only the resulting amplitude of several frequencies without a time range. This means that the current waveform contains frequencies not only from voices but also from other background noises. In order to separately see the individual frequencies within the waveform, we used **Fourier transform** to decompose the waveform into its individual frequencies.[8]

Short-time Fourier Transform (STFT) Just like most audio signals, content of single frequency sound waves inside each recording varies over time. Hence, it is ideal to represent the full spectrum of all frequencies within each audio segment. Thus, we applied **short-time Fourier transform** to overlapping time windows. Each time window size was 2048 Teslas per second (T/s).[8]

Stack STFT and Obtain Mel Spectrogram To obtain the mel spectrogram, We first stacked the results from short-time Fourier transform. Then, we converted the y-axis (frequency) of the result into a log scale, and also converted the wave amplitudes to decibels.[8] Consequently, we applied nonlinear transformation to change the frequency scale to mel scale so distances in pitch in any ranges sound equally distant. These were done to make the range of frequencies and amplitudes much more perceivable to humans.

3.3 Models

We designed **three binary classification models** that take preprocessed speech data as input and predict whether the person speaking in the audio file is intoxicated or sober. We used the model architecture in AlcoAudio [10] as our starting point and made adjustments to maximize the performance. Specifically, we experimented with (1) adjusting CNN layer depth in AlcoAudio architecture (`KAISD_naive`), (2) adjusting CNN layer depth in AlcoAudio architecture after preloading weights from model trained with German dataset (`KAISD_pretrained`), and (3) adding various numbers of inception modules (`KAISD_inception`). All models were trained and evaluated with our Korean intoxicated speech dataset for 15 epoches. For each epoch, accuracy (correct/total) was calculated using test set and parameters that resulted in the highest test set accuracy was saved.

KAISD_naive The original AlcoAudio model has four 2D convolutional (CNN) layers and three fully connected (FC) linear layers. We got the model architecture from AlcoAudio (without pretrained weights) and trained the model with our own data. We attempted to improve on this model by adjusting the number of CNN layers. We tried removing two CNN layers (2 CNN layers in total) and adding two CNN layers (6 CNN layers in total). Our results indicated that the network with 2 CNN layers perform the best for Korean language intoxicated speech detection. Thus, our final `KAISD_naive` consists of two 2D CNN layers and three FC layers. Detailed architecture is shown in Figure 2.

KAISD_pretrained We also tried training the model with Korean dataset after preloading weights from the AlcoAudio model, which was trained with German dataset. Thus, the weights of all the CNN layers and the second FC layer were preloaded from the AlcoAudio baseline model before training. The weights of first and third FC layers could not be preloaded due to matrix shape differences resulting from different input size. Similar to `KAISD_naive`, we experimented with CNN layer depths: two, four, and six CNN layers. The last two CNN layers of the six CNN layer network could not be preloaded because the original AlcoAudio model consists of only 4 CNN layers. Similar to `KAISD_naive`, the shallow architecture with two CNN layers performed the best.

KAISD_inception We attempted to improve the performance of `KAISD` by adding inception modules from GoogLeNet. Inception modules concatenate outputs from 3 CNN branches, each with different filter size, and one maxpooling branch [4]. We adopted the code from assignment 1 to implement `KAISD_inception`. We hypothesized that adding inception modules will improve performance because it allows the model to examine the input in a wider context while reducing the computation cost. After some hyper-parameter tuning, we found out that adding 15 inception blocks results in

[1] https://github.com/librosa/librosa

the best performance. Thus, our `KAISD_inception` consists of 3 CNN layers, 1 FC layer and 15 inception modules.

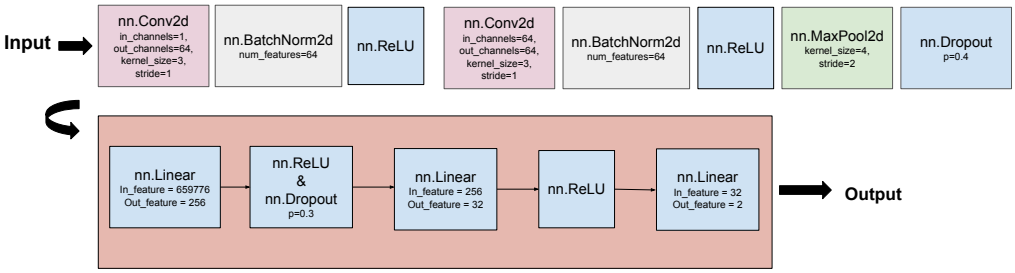


Figure 2: Detailed architecture of `KAISD_naive` and `KAISD_pretrained`.

4 Results

4.1 Quantitative Result

Since we developed binary classifiers, we used F1 scores as our primary evaluation metric. We carried out evaluation with validation set, 10% of Korean speech data collected by ourselves, and also with speech data of held out subject (YouTube_11), whose speech was not fed into any of the models during training. The baseline model we are comparing to is AlcoAudio (referred to as `AlcoAudio_baseline`). To compare our results, we carried out the same set of evaluations for AlcoAudio model. However, we could not load the weights of two FC layers due to input size differences. Thus, two FC layers of `AlcoAudio_baseline` were filled with random weights. The original AlcoAudio model evaluated with German dataset has unweighted average recall of 66.28 [10].

KAISD_naive The evaluation results for `KAISD_naive` candidate models are shown in Figure 3. `conv_shallow`, `conv_original` and `conv_deep` each have two, four and six CNN layers respectively. Since `conv_shallow` resulted in the highest F1 score, we selected `conv_shallow` as our `KAISD_naive` model.

KAISD_pretrained The evaluation results for `KAISD_pretrained` candidate models are shown in Figure 4. `pretrained_shallow`, `pretrained_original` and `pretrained_deep` each have two, four and six CNN layers respectively. We selected `pretrained_shallow` as our `KAISD_pretrained` model because, when compared to the 3rd decimal place F1 score of `pretrained_shallow` (0.869) was higher than F1 score of `pretrained_original` (0.867).

KAISD_inception The evaluation results for `KAISD_inception` candidate models are shown in Figure 5. We tested 7 different combination of inception block numbers. Since `inception_5` resulted in the highest F1 score, we selected `inception_5` as our `KAISD_inception` model.

Held Out Subject We carried out the same set of evaluations with speech data of YouTube_11, which was not included in our mixture of training, test, and validation sets. The results are in Figure 6. `KAISD_pretrained` performed the best with F1 score of 0.83.

Model	Precision	Recall	F1 Score
conv_shallow*	0.86	0.86	0.86
conv_original	0.73	0.97	0.83
conv_deep	0.84	0.32	0.47
AlcoAudio_baseline	0.51	1.00	0.67

Figure 3: Evaluation result of `KAISD_naive` candidate models.

Model	Precision	Recall	F1 Score
pretrained_shallow*	0.86	0.88	0.87
pretrained_original	0.89	0.85	0.87
pretrained_deep	0.74	0.81	0.77
AlcoAudio_baseline	0.51	1.00	0.67

Figure 4: Evaluation result of `KAISD_pretrained` candidate models.

Model	Precision	Recall	F1 Score
inception_1	0.89	0.98	0.94
inception_2	0.89	0.95	0.92
inception_3	0.98	0.91	0.95
inception_4	0.88	0.93	0.91
inception_5*	0.96	0.95	0.96
inception_6	0.91	0.94	0.92
inception_7	0.98	0.93	0.95
AlcoAudio_baseline	0.51	1.00	0.67

Figure 5: Evaluation result of `KAISD_inception` candidate models.

Model	Precision	Recall	F1 Score
KAISD_naive	0.70	1.00	0.82
KAISD_pretrained	0.71	1.00	0.83
KAISD_inception	0.63	0.86	0.73
AlcoAudio_baseline	0.50	1.00	0.67

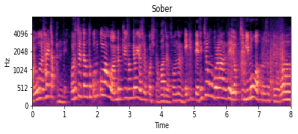
Figure 6: Evaluation result with YouTube_11 speech data, which was entirely held out during training.

4.2 Qualitative Result

Mel Spectrogram Here, we display spectrograms for drunk and sober speech (See Fig. 7) that our model successfully classified by their labels. The figure displays a clear difference between the two images - sober and drunk. The spectrogram for drunk speech displays a great amplitude (intensity) across wide range of frequencies. On the other hand, that for sober speech shows big amplitude over lower frequencies. From this, we conclude that drunk people tend to have less control over how loud they can speak. Moreover, drunk people tend to speak loud regardless of the pitch. Our accuracy suggests that our model is able to capture these differences.

Output from Intermediate Layer We display the intermediate output (Fig. 7) from convolution layers of `KAISD_pretrained`. Both figures (8 9) indicate that the model focuses on features from the lower frequencies. However, lower frequency regions are much more highlighted in the sober output, which implies that the model tends to examine the pitch and intensity of lower frequencies when classifying a speech as either sober or drunk.

Filter Visualization We show some filters from our best model, `KAISD_pretrained`. The figure (Fig. 10) displays that every filter learns a different set of features, allowing the model to view a larger context of the input and thus, make accurate generalisations.

(1) 

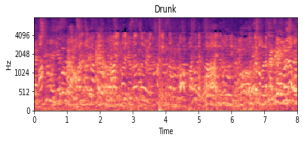
(2) 

Figure 7: Mel spectrogram of sober and drunk speech.

Figure 8: Intermediate output from the last filter `KAISD_pretrained` for sober mel spectogram. (1) indicates output from conv1, and (2) indicates output from conv2.

Figure 9: Intermediate output from the last filter `KAISD_pretrained` for drunk mel spectogram. (1) indicates output from conv1, (2) indicates output from conv2 respectively.

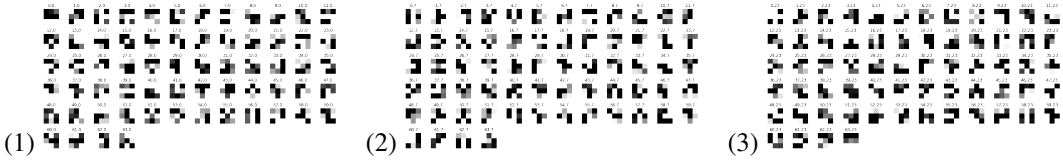


Figure 10: Illustration of C64 CNN filters from `KAISD_pretrained`. (1) indicates conv1, (2) indicates conv2[7], and (3) indicates conv2[23] filters respectively.

5 Discussion

5.1 Result Analysis

All of our models, `KAISD_naive`, `KAISD_pretrained`, and `KAISD_inception`, outperformed `AlcoAudio_baseline` in evaluations with both validation set and held out data. This may be because there exist fundamental difference in sober/drunk speech of German and Korean language. However, we replaced two FC layers with random weights, and our data was not collected in a controlled environment like the German dataset used to train AlcoAudio. These factors most likely also contributed to poor performance of `AlcoAudio_baseline`. Furthermore, when tested with held out data, all of the models showed much higher recall than precision. In other words, all models rarely classified drunk speech as sober speech while often classifying sober speech as drunk speech. This may be because we used audio file from Youtube as held out data. Most Youtubers use high-tension voice in their videos, which is often similar to their drunk voice.

5.2 Limitation & Future Work

Data Collection As we collected the dataset through various methods (our own recordings and YouTube channels), there was a varying degree of sound quality and background noises. We would like to collect data in a more consistent environment and method for a higher quality dataset. Also, since we labeled our data in a binary format (sober and drunk), our resulting models were binary classifiers. We want to make this more practical with BAC labeled inputs, so that the model can detect BAC level in speech.

Data Processing We used mel spectogram, a widely used representation in the field of audio classification. However, we believe that there could be other representations that suits the problem space better, such as Mel-frequency cepstral coefficients (MFCC). Comparing the result of various audio representations may lead to progress of the model.

6 Contributions

All members collected both the sober and drunk data with their own recordings, prepared for the final presentation and wrote parts of the final report. SeokJun implemented functions for data-preprocessing and visualizing filters and intermediate layers, modified AlcoAudio architecture to fit our input data, and helped with evaluation with validation set. Jeongeon was in charge of project management (overall direction and schedule) and collected YouTube data. Sujin collected YouTube data, implemented `KAISD_pretrained` and `KAISD_inception` models, performed hyper-parameter tuning on models, and conducted evaluation with validation set and held out data.

References

[1] Fadi Biadsy, William Yang Wang, Andrew Rosenberg, and Julia Bell Hirschberg. Intoxication detection using phonetic, phonotactic and prosodic cues. In *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, pages 3209–3212. ISCA, 2011.

[2] Tobias Bocklet, Korbinian Riedhammer, and Elmar Nöth. Drink and speak: On the automatic classification of alcohol intoxication by acoustic, prosodic and text-based features. In *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, pages 3213–3216. ISCA, 2011.

[3] Daniel Bone, Matthew P. Black, Ming Li, Angeliki Metallinou, Sungbok Lee, and Shrikanth Narayanan. Intoxicated speech detection by fusion of speaker normalized hierarchical features and gmm supervectors. In *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, pages 3217–3220. ISCA, 2011.

[4] Yangqing Jia Pierre Sermanet Scott Reed Dragomir Anguelov Dumitru Erhan Vincent Vanhoucke Christian Szegedy, Wei Liu and Andrew Rabinovich. Going deeper with convolutions. 2015.

[5] Bavarian Archive for Speech Signals. Alcohol language corpus - alc, 2011-09-30.

[6] Arin Kim. 153 caught drunk driving nationwide on first day of toughened dui law, 2019-06-25.

[7] KOSIS. Drunk driving traffic accident rate (cities and provinces), 2020-07-24.

[8] Roberts Leland. Understanding the mel spectrogram. `https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53`, 2020.

[9] Sungmin Lim. Traffic accident on the west gate, 2016-03-29.

[10] Shreesha Murthy. Alcoaudio. `https://github.com/ShreeshaN/AlcoAudio`, 2020.

[11] Bjorn Schuller, Stefan Steidl, Anton Batliner, Florian Schiel, and Jarek Krajewsk. The interspeech 2011 speaker state challenge. 2011.