INTOXICATED SPEECH DETECTION

Team 9

20160793 SeokJun Kim
20160811 Jeongeon Park
20170841 Sujin Han

Problem - Drunk Driving in Korea

Drunk driving is a very serious problem in Korea.

334.2 drunk driving cases per day in 2018

15,708 traffic accidents caused by drunk driving in 2019

Drunk driving cases occur **even at KAIST**

홈 > 캠퍼스

서측 쪽문에서 교통사고 발생

임성민 기자 | 입력 2016.03.29 16:51 | 댓글 0

지난 12일 0시 35분경 서측 쪽문 부근에서 오토바이를 탄 남학우가 기숙사 쪽으로 걸어가고 있던 여학우 두 명과 충돌한 사고가 발생했다. 가해 학생은 사건 당시 음주 상태였던 것으로 추정되며, 안전모를 쓰고 있지 않았다. 해당 오토바이는 번호판이 부착되지 않은 미등록 오토바이였다.

Drunk driving accident at west gate, 2016

http://times.kaist.ac.kr/news/articleView.html?idxno=3452

Problem - Limitations in Existing Methods

Breathalyzer

Very sensitive to temperature

Can register alcohol from the mouth, throat or stomach

DUI Blood Testing

Difficult to take due to physical limitation

Problem - Our Solution

Breathalyzer

DUI Blood Testing

Audio detector that can classify whether a person is drunk **in speech**

Previous Work - Intoxicated Speech Detection

- Intoxicated Speech Detection by Fusion of Speaker Normalized Hierarchical Features and GMM Supervectors
- Intoxication Detection using Phonetic Phonotactic and Prosodic Cues
- Drink and Speak: On the Automatic Classification of Alcohol Intoxication by Acoustic, Prosodic and Text-Based Features

→ All used **Alcohol Language Corpus (ALC)**, a german recording dataset.
(€1,020.00)

↓

A **binary classifier** that determines
whether the speaker is intoxicated or not, in **Korean speech**!

Approach

Data Collection

Voice Recordings,
YouTube Videos

Preprocessing

Audio to Image,
Mel spectrogram

Model

CNN,
Inception Module

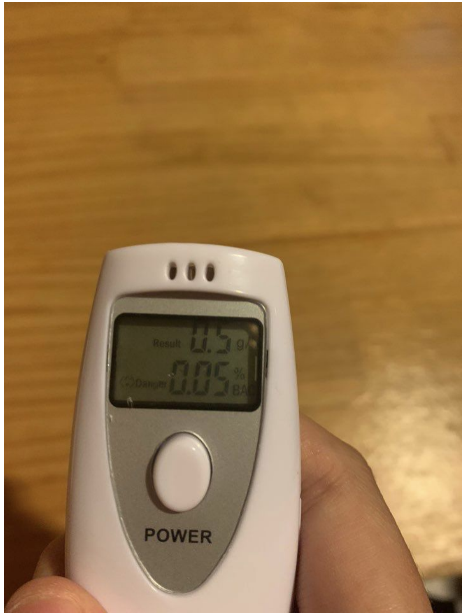
Results Analysis

Comparison of three models

Data Collection

- We collected in total **16 hours 21 minutes** of **free speech** audio files

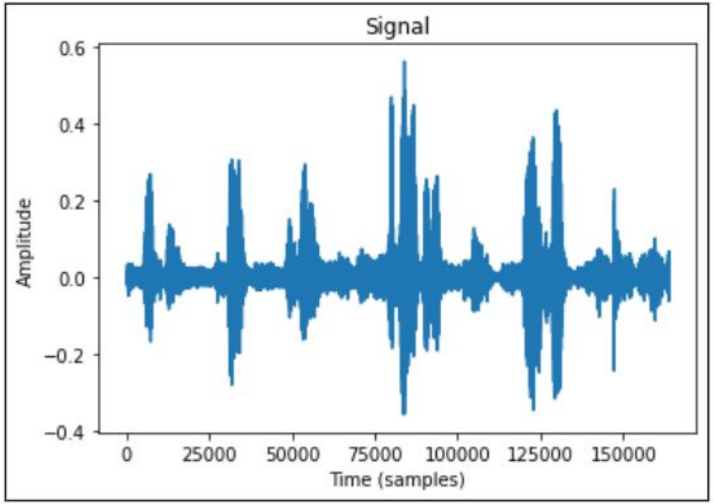
(Our own recordings + YouTube videos)



	Audio recordings	YouTube videos
Drunk BAC >= 0.08%	9 hrs 56 mins (some were removed during evaluation)	1 hr 58 mins
Sober	2 hrs 21 mins	2 hrs 5 mins

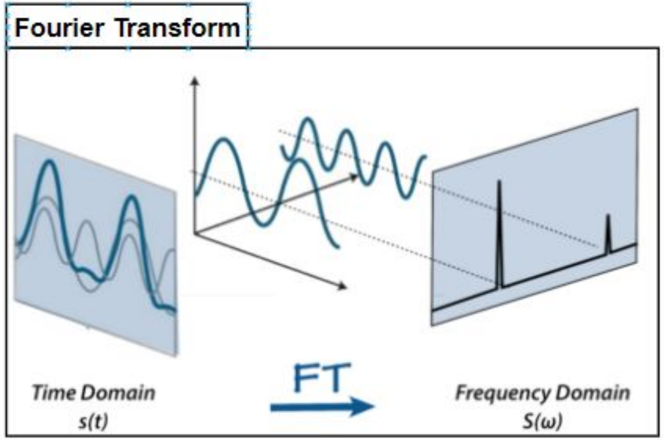
Preprocessing - Audio to Mel Spectrogram

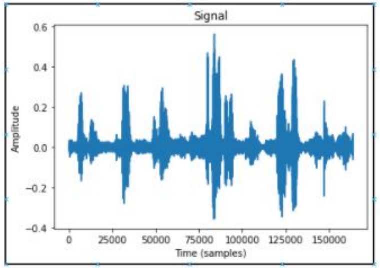
- **Represent audio signal as waveforms (8 second segment)**
- Convert signals from each time window into frequency domain using short time Fourier Transform
- Stack Fourier Transforms to get spectrogram
- Apply non-linear transformation to get mel spectrogram

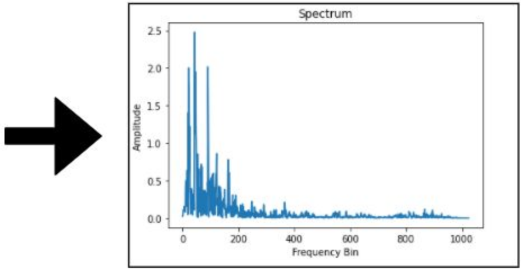


Preprocessing - Audio to Mel Spectrogram

- Represent audio signal as waveforms (8 second segment)
- **Convert signals from each time window into frequency domain using short time Fourier Transform**
- Stack Fourier Transforms to get spectrogram
- Apply non-linear transformation to get mel spectrogram

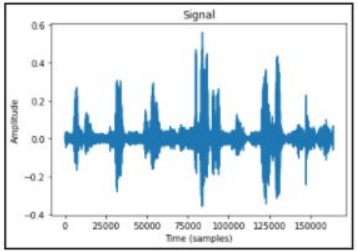


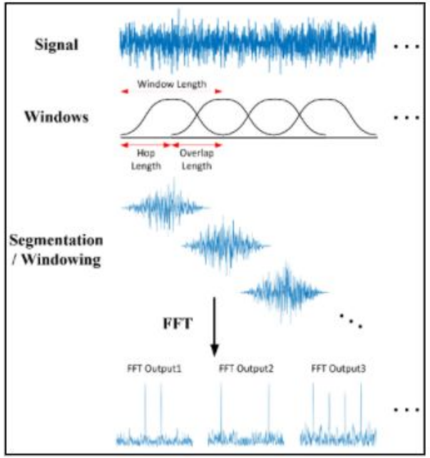


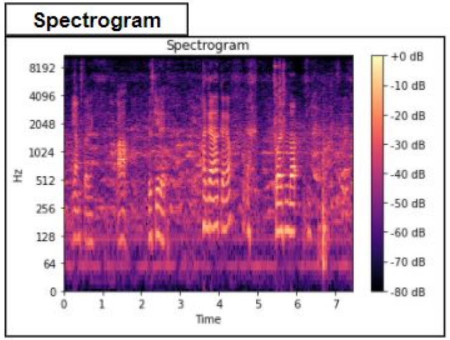


Preprocessing - Audio to Mel Spectrogram

- Represent audio signal as waveforms (8 second segment)
- Convert signals from each time window into frequency domain using short time Fourier Transform
- **Stack Fourier Transforms to get spectrogram**
- Apply non-linear transformation to get mel spectrogram



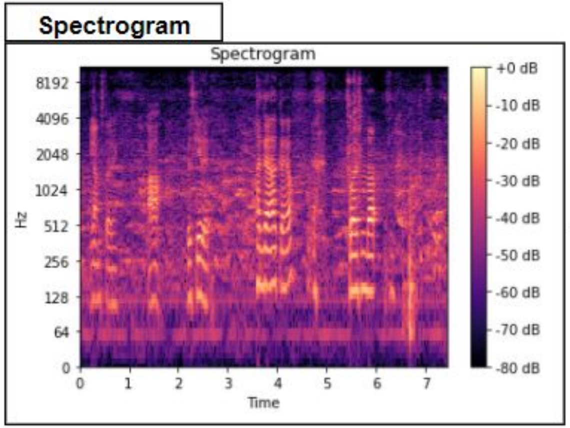




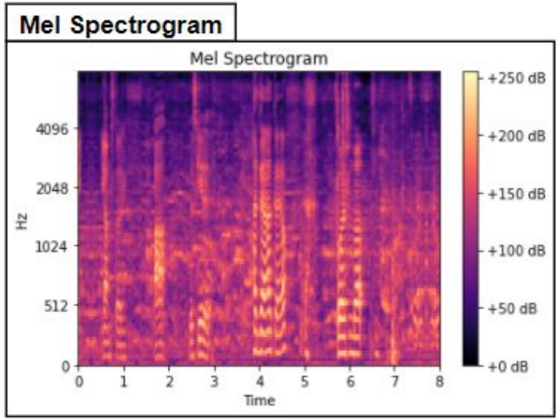
1. stack Fourier Transforms
2. wave amplitude to human perceivable **decibels**

Preprocessing - Audio to Mel Spectrogram

- Represent audio signal as waveforms (8 second segment)
- Convert signals from each time window into frequency domain using short time Fourier Transform
- Stack Fourier Transforms to get spectrogram
- **Apply non-linear transformation to get Mel spectrogram**



frequency scale to **mel** scale



Equal distances in pitch sound are *perceived differently*

500 ~ 1000 Hz ==> noticeable difference
10,000 ~ 10,500 Hz ==> unnoticeable difference

Equal distances in pitch **sound equally distant !!!**

Model - KAISD (Korean language Alcohol Intoxicated Speech Detector)

01

KAISD-naive

4 CNN layers with 3 FC layers

02

KAISD-pretrained

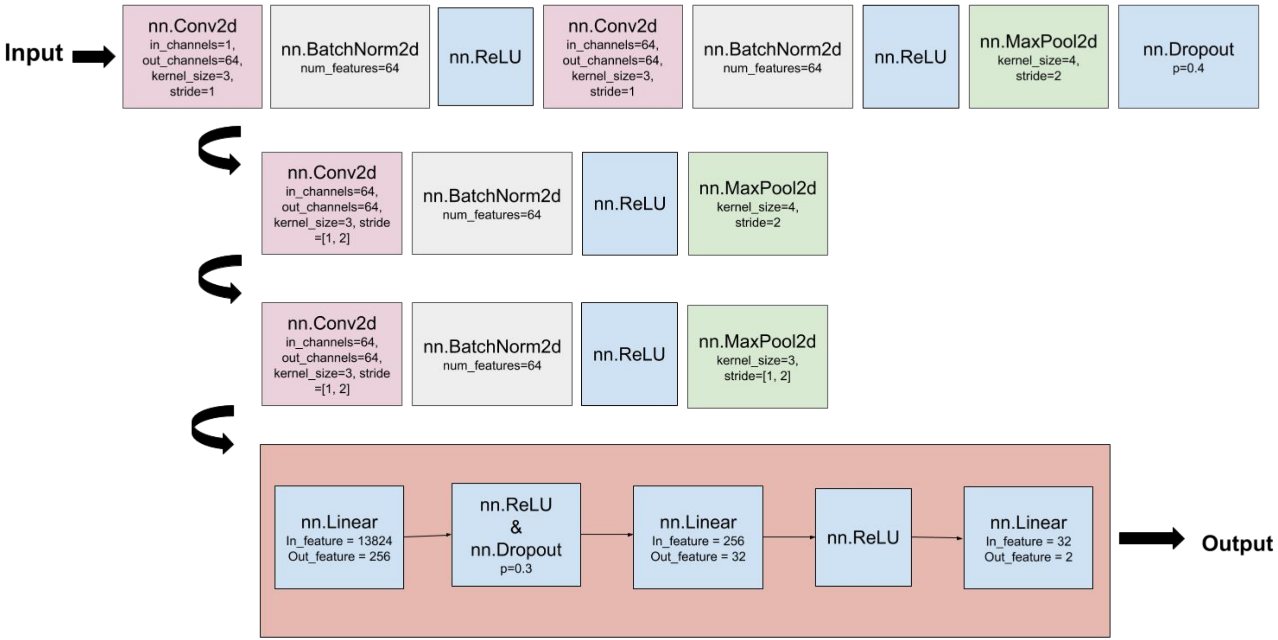
KAISD-native with some weights preloaded from **AlcoAudio**

03

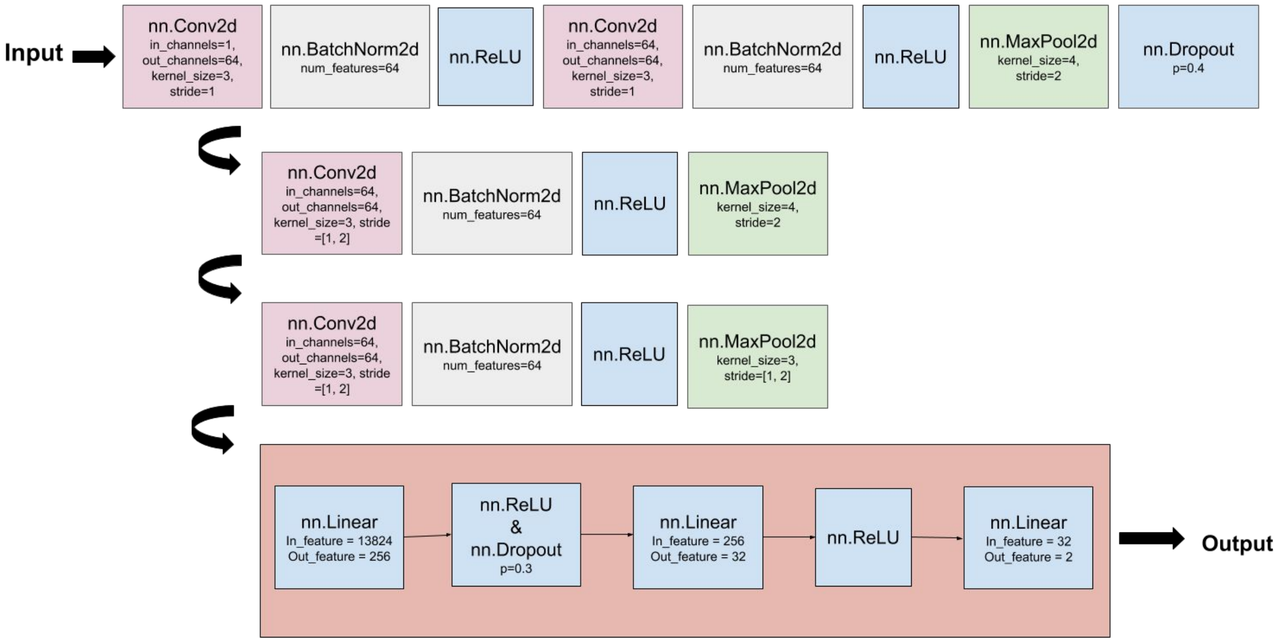
KAISD-inception

Model with 15 inception modules, 3 CNN and 1 FC layers

Model Architecture - KAISD-naive

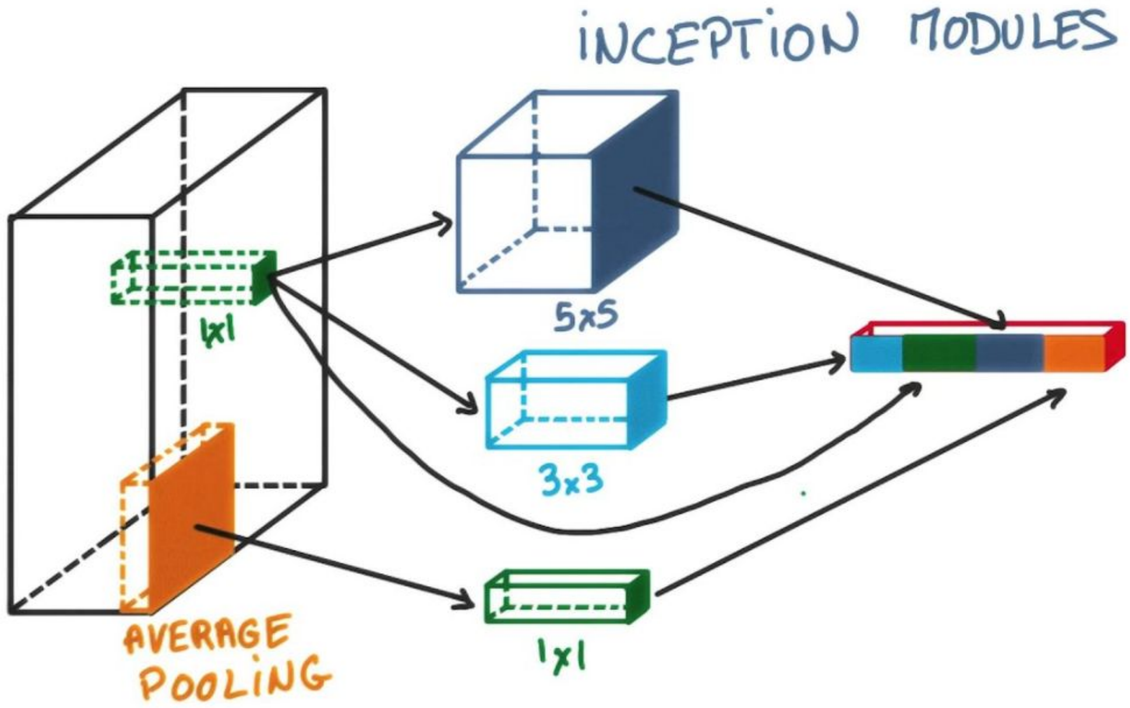


Model Architecture - KAISD-pretrained



Weights of CNN1, CNN2, CNN3, CNN4 and FC2 were loaded from AlcoAudio

Model Architecture - KAISD-inception

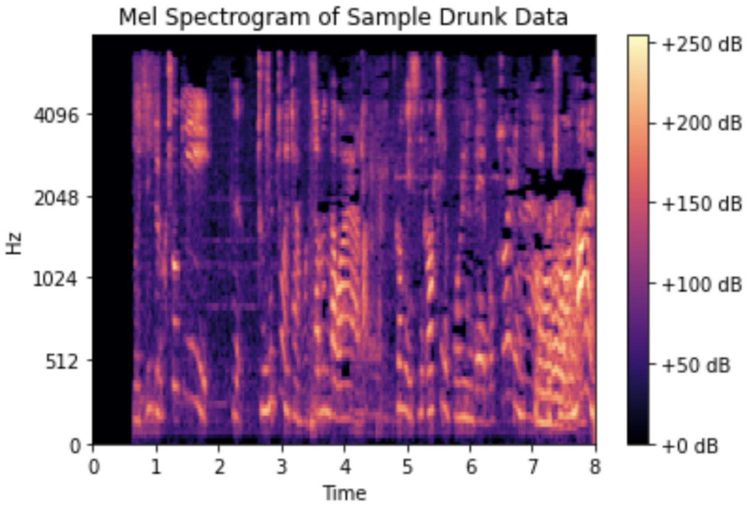


Experiment Setup

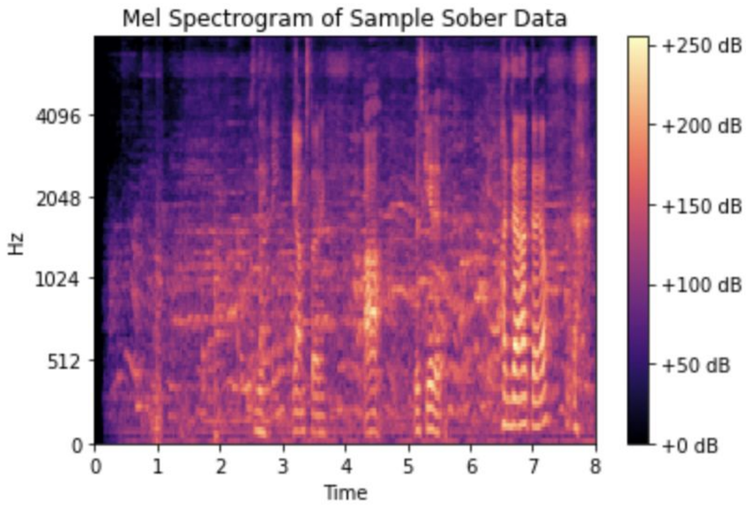
- We used in total **5,107** audio segments
  - 71.8% drunk audio segments
  - 28.2% sober audio segments
- The segments were divided into Training-80%, and Test-10%, Validation-10%
- We compared **three models** and their performances
  - KAISD-naive, KAISD-pretrained, KAISD-inception

Results - Qualitative

Mel Spectrogram of Sample Drunk Data

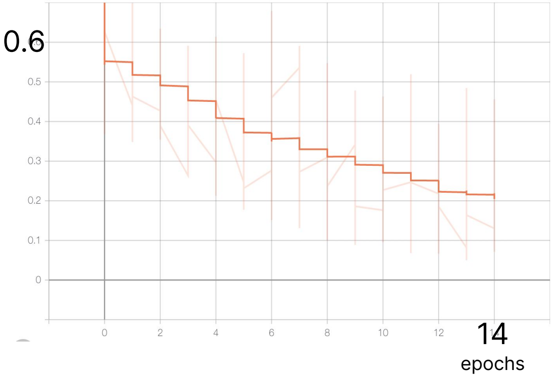


Mel Spectrogram of Sample Sober Data

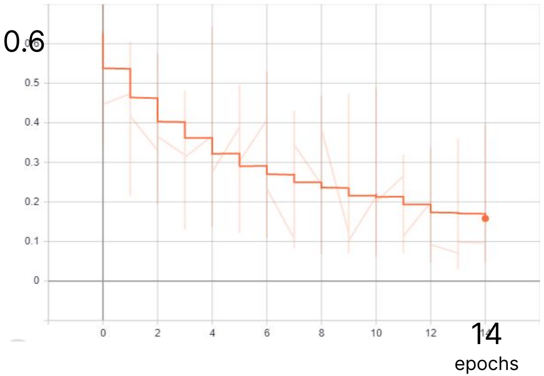


Results - Quantitative (1/2)

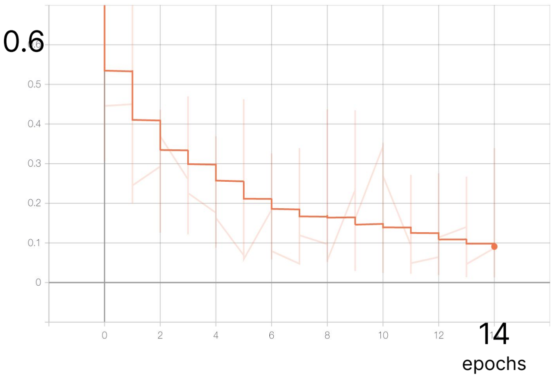
Train Loss



KAISD-naive



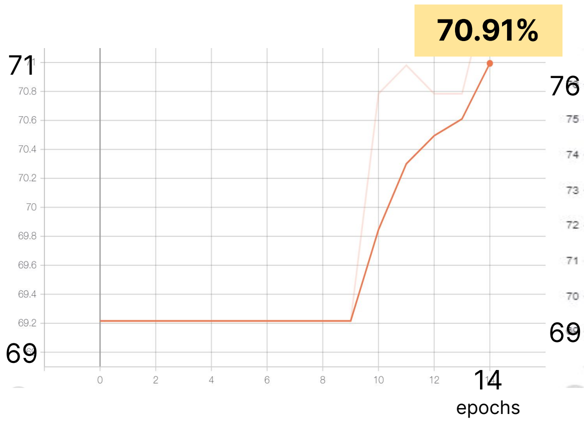
KAISD-pretrained



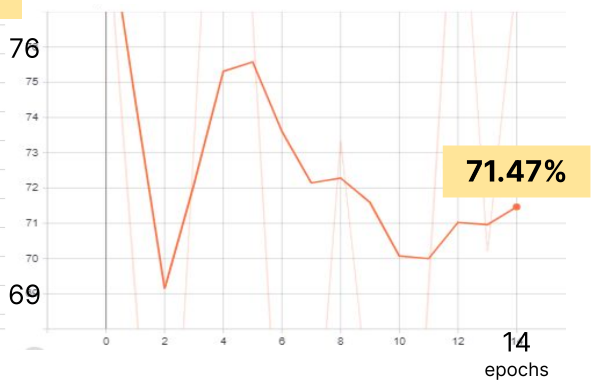
KAISD-inception

Results - Quantitative (2/2)

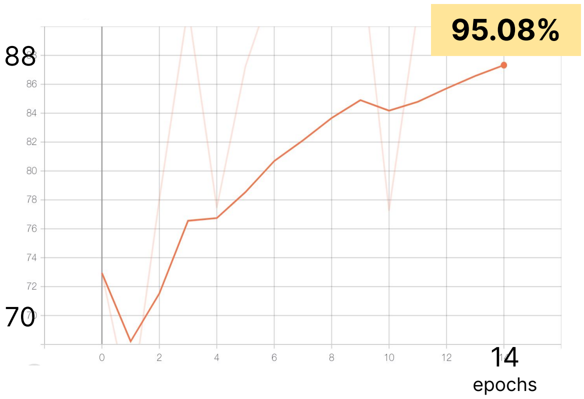
Validation Accuracy (with Test set)



KAISD-naive



KAISD-pretrained



KAISD-inception

Evaluation

- **F1 Scores** (with Validation set)
  - KAISD-naive: **85.39%**, KAISD-pretrained: **81.67%**, KAISD-inception: **98.58%**

***Predicts drunk every time**

	Predicted Drunk	**Predicted Sober**
Actually Drunk	374	0
Actually Sober	128	0

KAISD-naive

	Predicted Drunk	**Predicted Sober**
Actually Drunk	254	107
Actually Sober	7	144

KAISD-pretrained

	Predicted Drunk	**Predicted Sober**
Actually Drunk	347	3
Actually Sober	7	155

KAISD-inception

Conclusion - Discussion & Future work (1/2)

- Since our data was collected through various methods, recordings had varying degree of background noise.

  → Collect data in a more consistent environment.

- Our model is a binary classifier - can only distinguish between drunk and sober.

  → Detect BAC (Blood Alcohol Content) level in speech **- more practical!**

Conclusion - Discussion & Future work (2/2)

- Although mel spectrogram is widely used in the field of audio classification, some other representation might've suited this problem better.

  → Test with other audio representations. Ex. MFCC

- Due to limited time and GPU usage in colab, we could not run the model many times for hyperparameter tuning :(

  → Get a computer with cuda compatible GPU and spend more time for hyperparameter tuning.

Thank you

And please don’t drink and drive!

(even bicycles or scooters 😉)