Classifying noise sounds

Presented by:
Alanoud Alosaimi, Raghad Althunayan, Shaikha bin Ateeq

TABLE OF CONTENTS

Introduction

Sounds are all around us. Whether directly or indirectly. Sounds outline the context of our daily activities, conversations, music, noise . The human brain is continuously processing and understanding this audio data, so how the machine can understand it?

Goal:

Apply Deep Learning techniques to the classification of environmental sounds:

- Assisting deaf individuals in their daily activities
- Safety and security capabilities
- Smart home use

Tools

Tensorflow

Keras

Librosa

Hdf

Pickle

Numpy

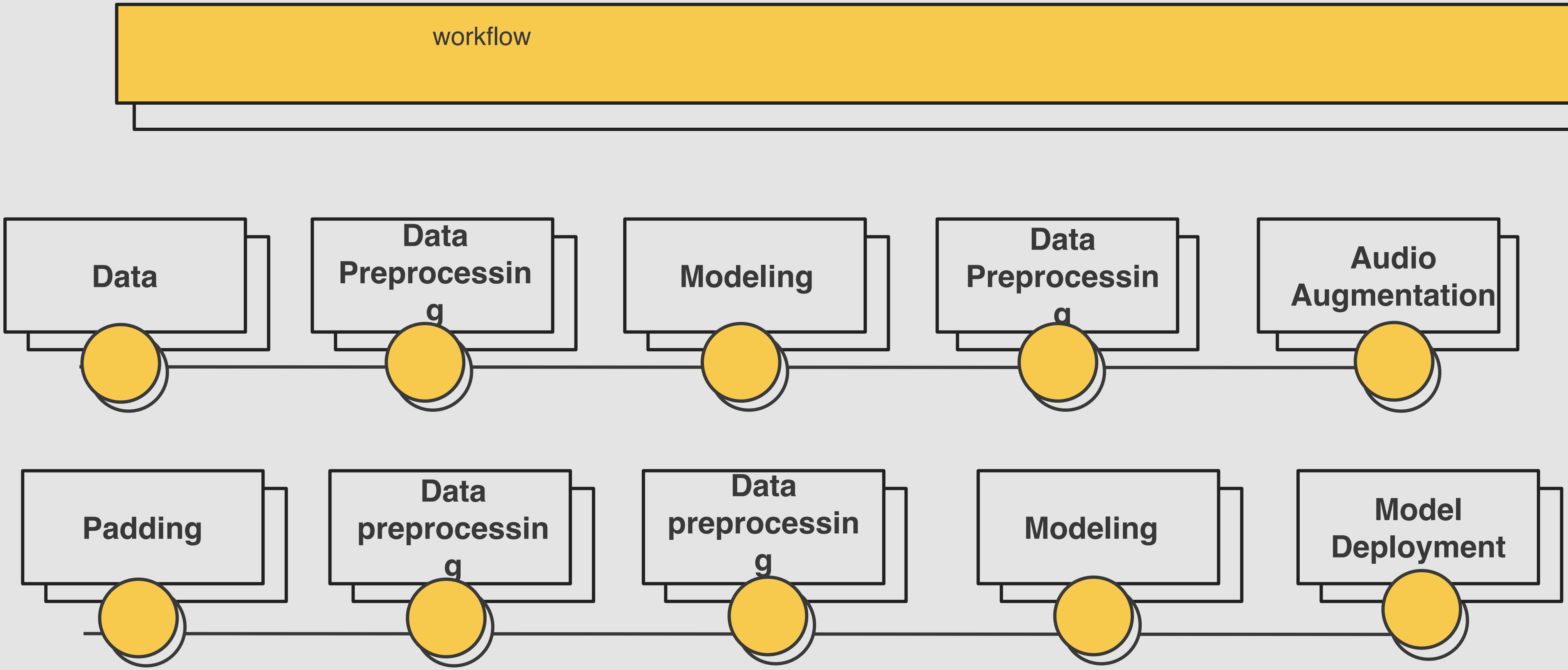
Sklearn

Pandas

Matplotlib

PRO

Flask



workflow
Data
Data Preprocessing
Modeling
Data Preprocessing
Audio Augmentation
Padding
Data preprocessing
Data preprocessing
Modeling
Model Deployment

Data Story

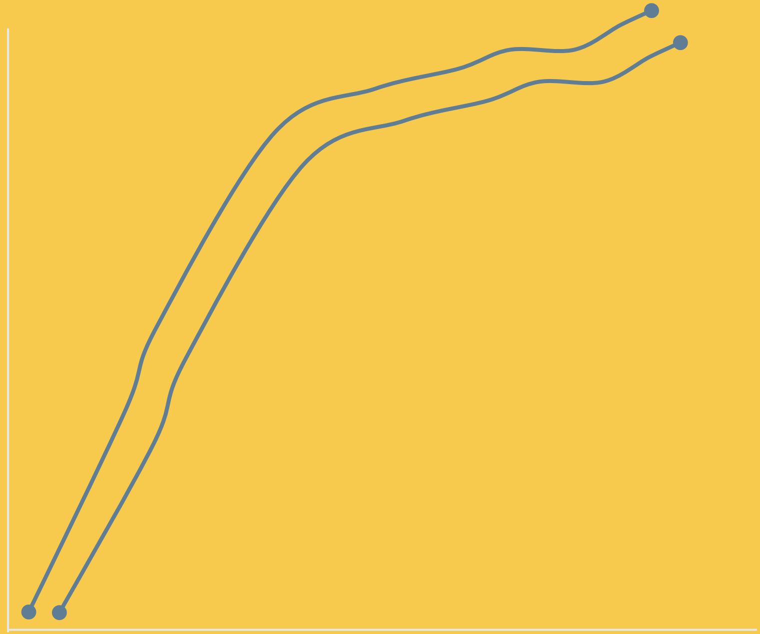
Urban sounds classification

From kaggel

Row = 8733

Class label [10]=

- Air Conditioner
- Car Horn
- Children Playing
- Dog bark
- Drilling
- Engine Idling
- Gun Shot
- Jackhammer
- Siren
- Street Music



Data Story

Urban sounds classification

From kaggel

Row = 8733

Class label [10]=

- Air Conditioner
- Car Horn
- Children Playing
- Dog bark
- Drilling
- Engine Idling
- Gun Shot
- Jackhammer
- Siren
- Street Music

+

Noise sounds classification

From kaggel

Row =2171

Class label [11]=

- Applause
- Keys_jangling
- Telephone
- Cough
- Microwave_oven
- Laughter
- Tearing
- Fireworks
- Bus
- Scissors
- Computer_keyboard'

Before = (8733 , 10)

After = (10904 , 21)

One dataset not enough
COMPLEXITY NEEDED!

Data Story

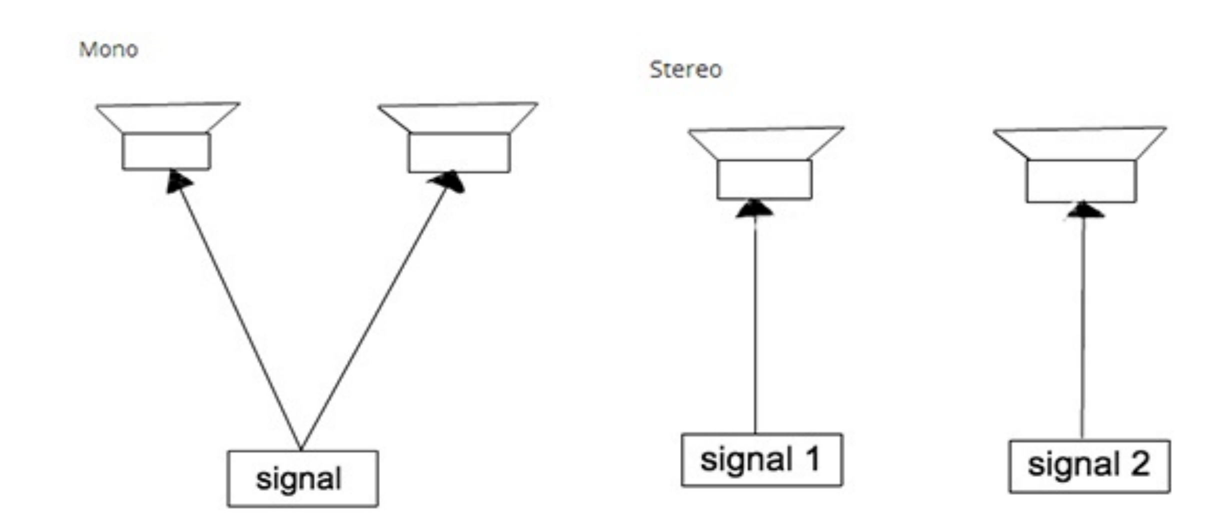
Audio Samples

Children playing	Drilling
Computer keyboard	Siren
Applause	Dog bark
Microwave oven	Telephone

We heard the audio samples, Do we know now its Properties **?**

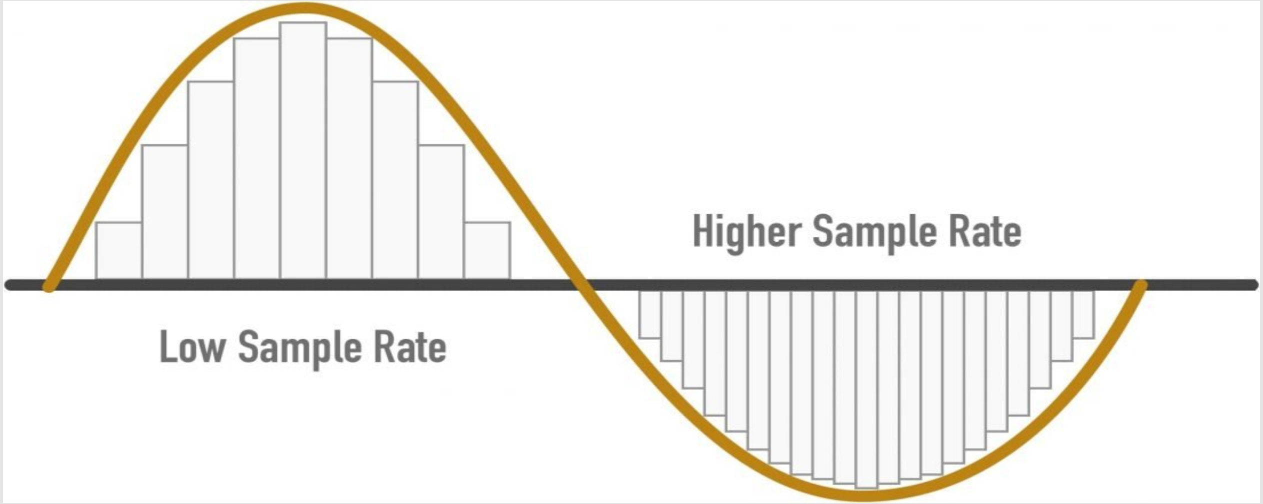
Audio Properties:

- **Audio Channels**



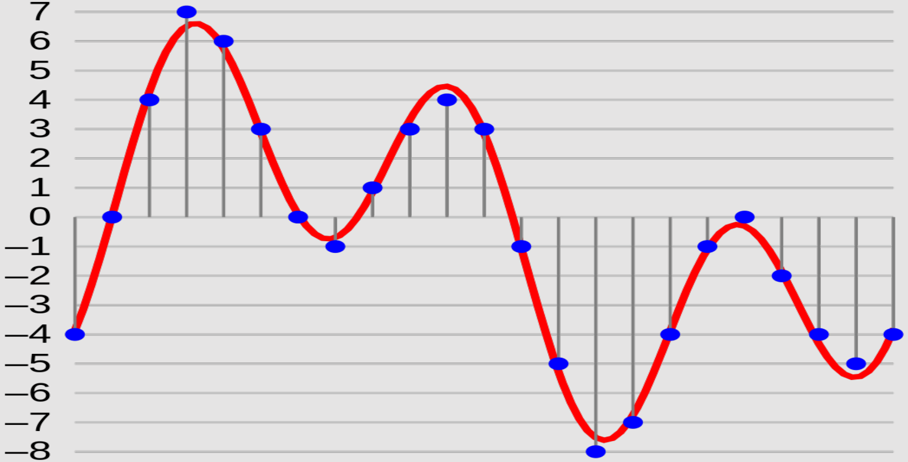
Audio Properties:

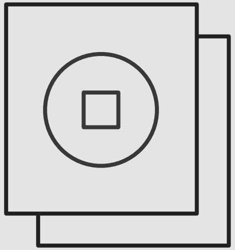
- **Sample Rate**



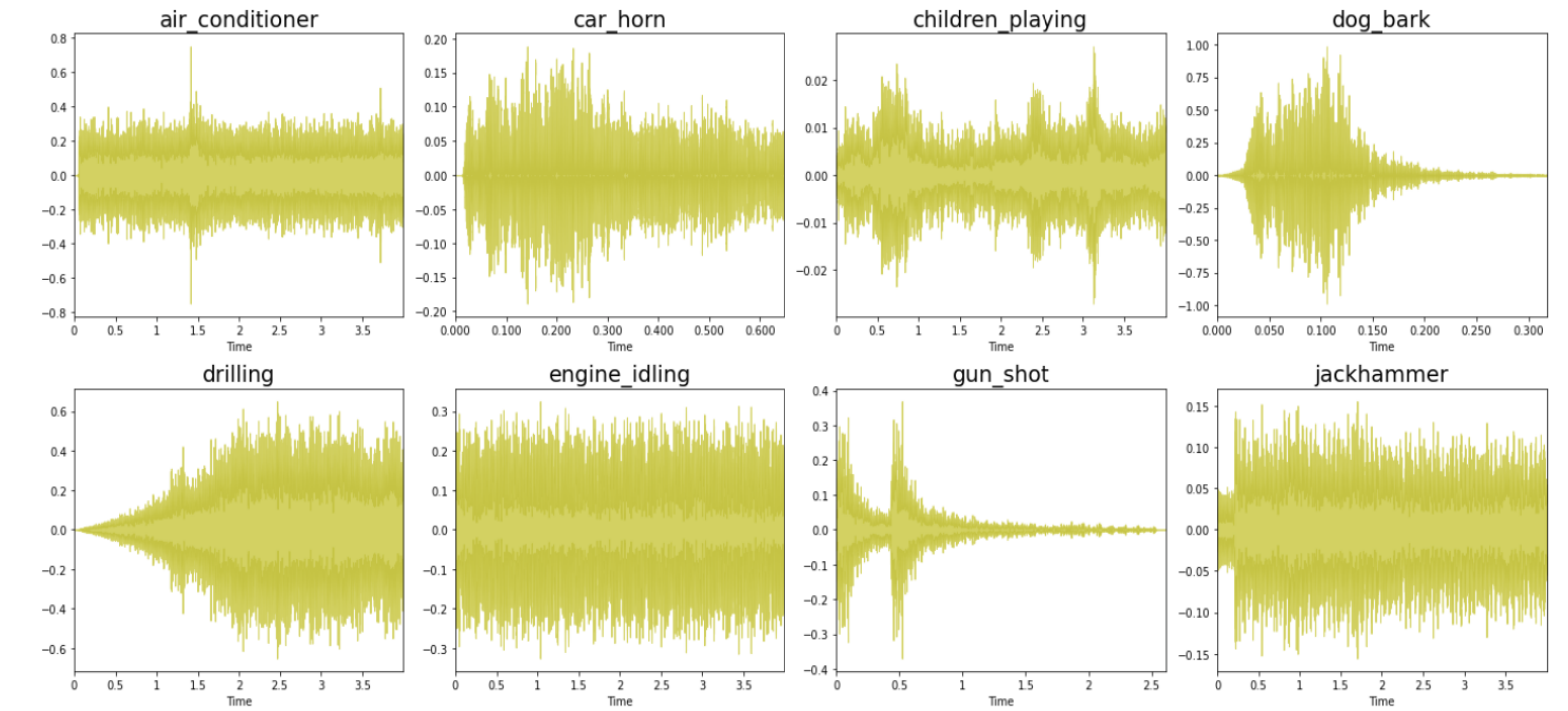
Audio Properties:

- **Bit-Depth**

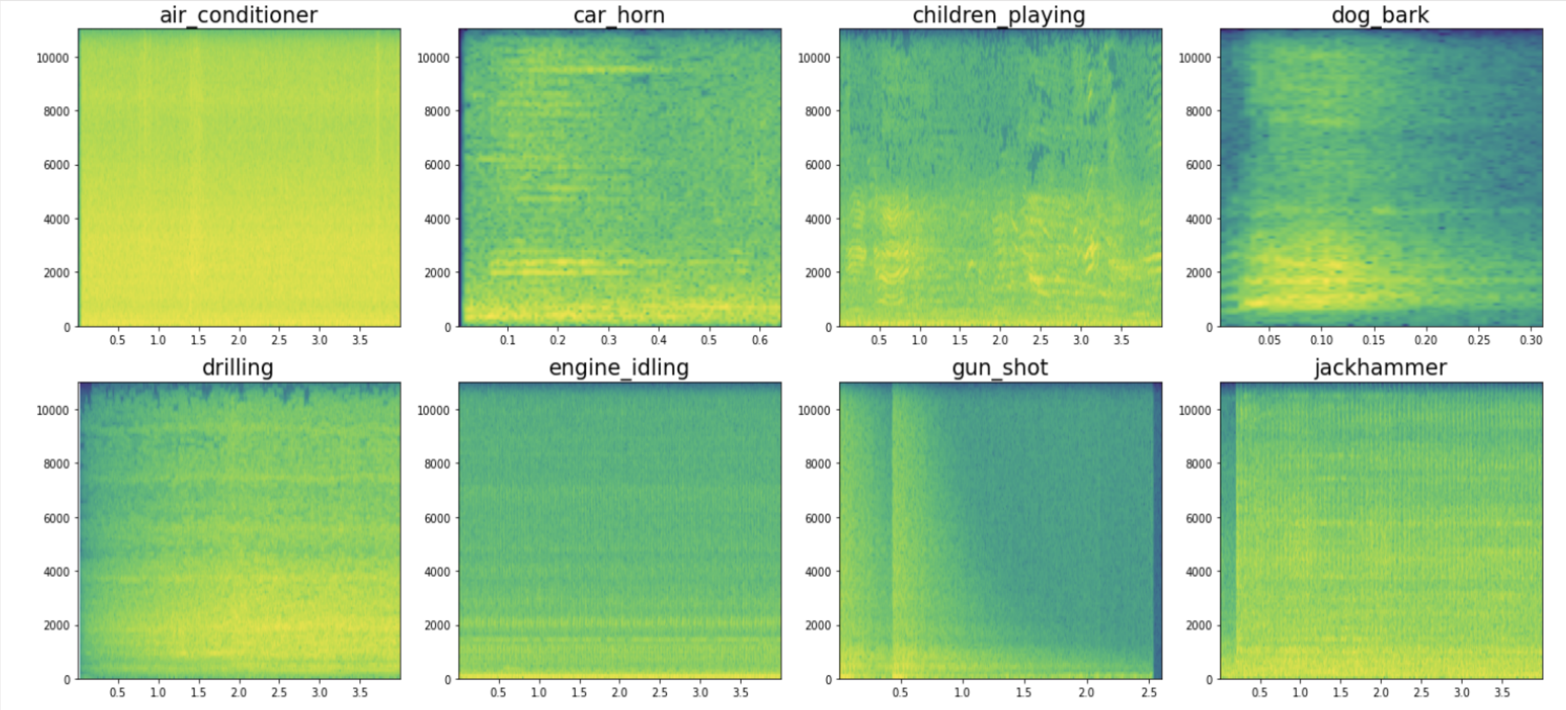




Waveplot


air_conditioner
car_horn
children_playing
dog_bark
drilling
engine_idling
gun_shot
jackhammer
Time

Specgram Plot



but!!

Can we use the spectrum images as input
for our model? or something else?

Feature Extraction Method: Mfcc

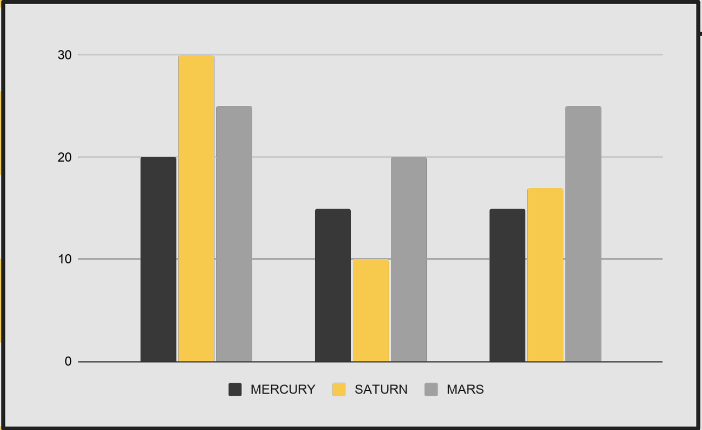
The MFCC summarises the frequency distribution across the window size, to analyse both the frequency and time characteristics of the sound. These audio representations will allow us to identify features for classification.

```
array([-0.46156558, -0.60931027, -0.7269877 , -0.65179384, -0.45346004,
       -0.35503793, -0.309869  , -0.29231593, -0.3408025 , -0.47978437,
       -0.56441855, -0.668554  , -0.77650434, -0.9077673 , -0.8961237 ,
       -0.7808625 , -0.7378508 , -0.73105305, -0.7016147 , -0.7287925 ,
       -0.7165885 , -0.6830154 , -0.6216891 , -0.604039  , -0.60051143,
       -0.5490541 , -0.5028578 , -0.46407667, -0.4999416 , -0.6309592 ,
       -0.8549022 , -0.9061938 , -0.8369524 , -0.82082754, -0.7269997 ,
       -0.57281935, -0.47586957, -0.44021174, -0.4105482 , -0.3610335 ,
       -0.3309665 , -0.29519534, -0.27117366, -0.32846212, -0.35230893,
       -0.31837335, -0.23812626, -0.21707068, -0.22060803, -0.20598894,
       -0.22767977, -0.2672934 , -0.3058963 , -0.35964936, -0.38586545,
       -0.38850698, -0.44798654, -0.550652  , -0.65550697, -0.80136925,
       -0.7675937 , -0.7436201 , -0.6249931 , -0.4383348 , -0.34919867,
       -0.33050042, -0.32238662, -0.32141203, -0.29950154, -0.26876846,
       -0.25420395, -0.24101993, -0.2581274 , -0.2699363 , -0.27291867,
       -0.24633032, -0.21710564, -0.22357889, -0.22391006, -0.22359052,
       -0.2660913 , -0.28693932, -0.2855701 , -0.27417266, -0.26417005,
       -0.22955446, -0.21966569, -0.20383127, -0.21017219, -0.23220706,
       -0.21959333, -0.19229598, -0.186843  , -0.17777492, -0.20924585,
       -0.2963816 , -0.32962874, -0.36754662, -0.3978389 , -0.3803654 ,
       -0.3560971 , -0.30606392, -0.27818236, -0.25736403, -0.27018398,
       -0.27297518, -0.26325408, -0.29688725, -0.3417926 , -0.3791692 ,
       -0.39299247, -0.39608532, -0.41692922, -0.4035706 , -0.36449316,
       -0.33859769, -0.36949402, -0.4120001 , -0.40582865, -0.38918048,
       -0.4041523 , -0.39276654, -0.3948499 , -0.41085315, -0.4472869 ,
       -0.53813744, -0.60765874, -0.6059553 , -0.50796515, -0.545878  ,
       -0.52863103, -0.4292929 , -0.3391605 , -0.30450806, -0.27651113,
       -0.23635665, -0.21272539, -0.20371285, -0.19710773, -0.2035437 ,
       -0.20224652, -0.20834276, -0.20498988, -0.26224333, -0.34169963,
       -0.360925  , -0.3994224 , -0.50174165, -0.52654827, -0.47546023,
       -0.4068555 , -0.39751032, -0.40414682, -0.44837436, -0.57356775,
       -0.6287995 , -0.56714004, -0.46860185, -0.36755478, -0.30301824,
       -0.3653177 , -0.43069097, -0.43489447, -0.44246167, -0.50372636,
       -0.46702236, -0.34863254, -0.27881584, -0.27385667, -0.3146194 ,
       -0.35711053, -0.2876631 , -0.19479881,  0.        ], dtype=float32)
```

All we think data now is PERFECT!

But, the models **didn't predict** some classes (**imbalanced data**)

Classes Imbalanced



Solving Classes Imbalanced

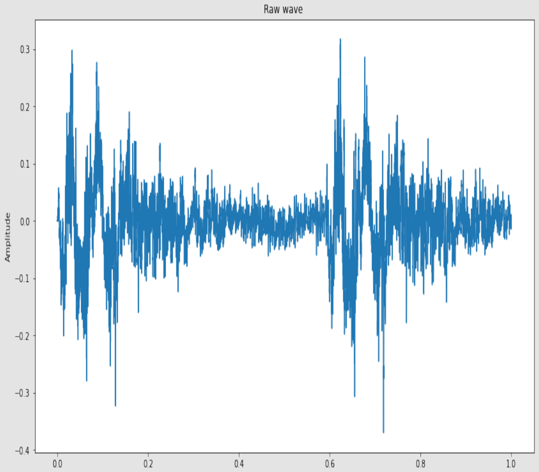
Data Augmentation

The objective is to make our model invariant to those perturbations and enhance its ability to generalize.

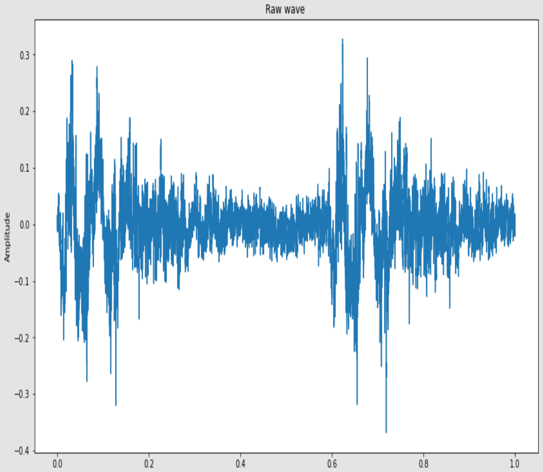
Data Augmentation:

- Noise

Orignal



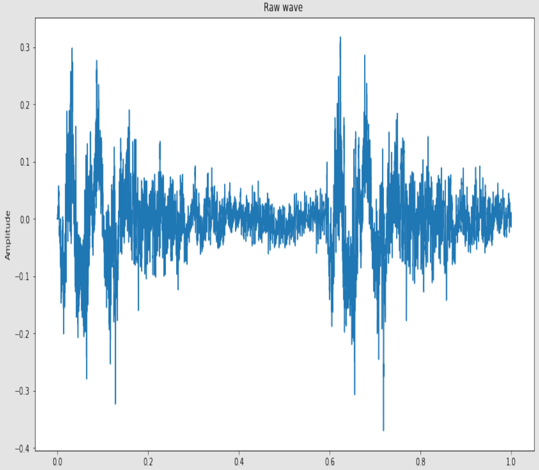
Noise



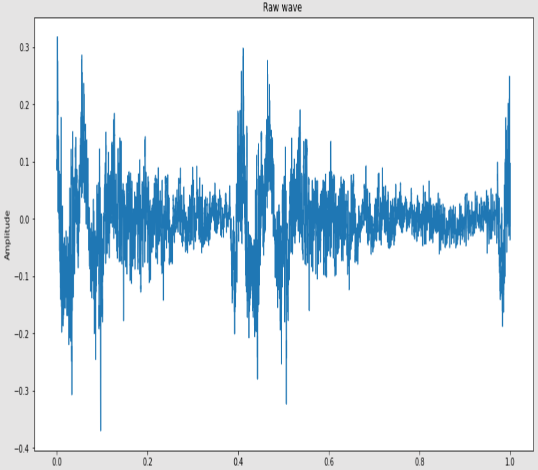
Data Augmentation:

- Shifting the Audio

Orignal



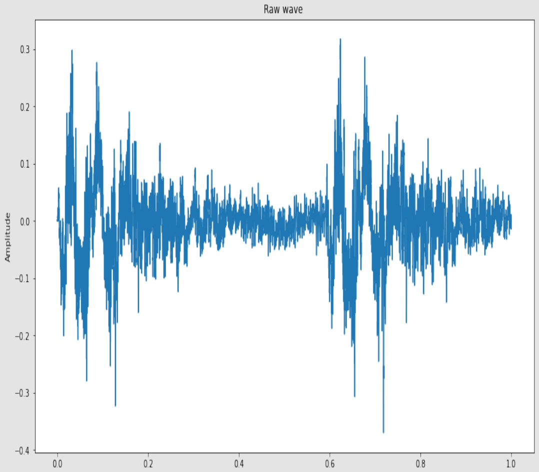
Shifting



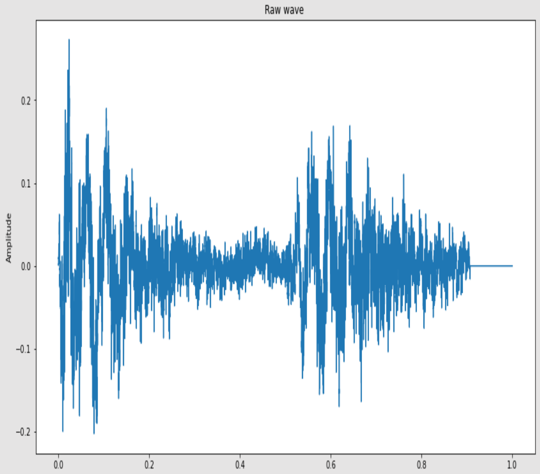
Data Augmentation:

- Time stretching (changing play time)

Orignal



Stretching



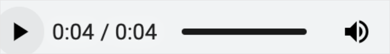
Data Augmentation

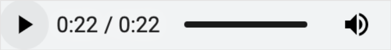
Initial data	New data	Merged data
10903	54515	65418
(10903, 21)		(65418, 21)

solving didn't enough!..WHY?

Padding

Length sounds problem





Least length (fram_num = 174)

Longest length (fram_num = 1292)

Least , why?

Split size

Train :=0.8
Test:0.2

Train :=0.75
Validations:0.25

Models result (1)

	Train acc	Val acc	Epochs	batch
Beasline	0.61	0.45	500	50
Cnn2 (1)	0.28	0.25	200	50
Cnn2 (2)	0.60	0.44	1000	50
Cnn2 (3)	0.33	0.21	300	300
Cnn2 (4)	0.48	0.44	500	50

Models result (2)

	Train acc	Val acc	Epochs	batch
Beasline	0.69	0.63	500	50
Cnn2D (1)	0.80	0.77	300	300
Cnn2d (2)	0.89	0.78	300	300
Cnn2d (3)	0.81	0.75	300	300
Cnn1d	0.27	0.3	500	50
lstm	0.3	0.34	500	50

FINAL RESULT

- Train acc= 0.81
- Val acc= 0.75
- test acc= 0.77

MODEL ARCHITECTURE

Conv2D (Filter size = 128)
MaxPooling2D

Conv2D (Filter size = 128)
MaxPooling2D
Droupout (0.8)

Flatten

Dense (512, activation = ‘relu’)
Droupout (0.8)

Dense (512, activation = ‘relu’)
Droupout (0.8)

Dense (18, activation = softmax)

Challenges

Merge two dataset

Ram crash

Long time

Future work

- build an app that helps the deaf in their daily life
- Transfer Learning

Model Deployment Demo

Thanks!