EMOTION RECOGNITION USING RAVDESS DATABASE

AGENDA

01 WHY EMOTION DETECTION?

02 THE RAVDESS DATASET

03 APPROACH

04 EMOTION RECOGNITION ON AUDIO

05 EMOTION RECOGNITION ON VIDEO

06 DEMO

07 FUTURE SCOPE

01
WHY EMOTION DETECTION?

A MAN'S NEW BEST FRIEND

- 80% of companies in the US use chatbots
- 30% of customer service jobs will be replaced by AI by 2020
- Emotion Recognition from audio and video data can tremendously improve ML/AI performance in human-facing tasks
- Examples of applications: Intelligent chatbots, emotion recognition systems in vehicles, & virtual assistants and maybe even BAYMAX!

02
THE DATA SET

Ryerson Audio-Visual Database of
Emotional Speech and Song

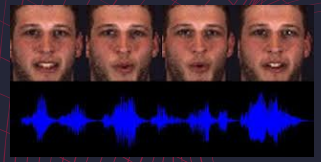
RAVDESS

- RAVDESS contains 7356 files
- 24 professional actors (12 female, 12 male)
- The filename consists of a 7-part numerical identifier (e.g., 02-01-06-01-02-01-12.mp4). These identifiers define the stimulus characteristics:
    - **Modality** (01 = full-AV, 02 = video-only, 03 = audio-only).
    - **Vocal channel** (01 = speech, 02 = song).
    - **Emotion** (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
    - **Emotional intensity** (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
    - **Statement** (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
    - **Repetition** (01 = 1st repetition, 02 = 2nd repetition).
    - **Actor** (01 to 24. Odd numbered actors are male, even numbered actors are female).

03
APPROACH

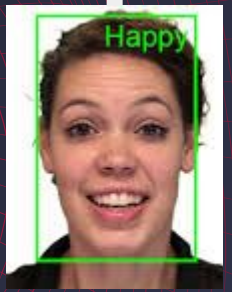
High level overview of modeling approach

Audio




Angry Speech
Angry Song
Happy Speech
Happy Song
Neutral Speech
Neutral Song
Sad Speech
Sad Song
Fearful Speech
Fearful Song
Calm Speech
Calm Song
Disgusted Speech
Surprised Speech

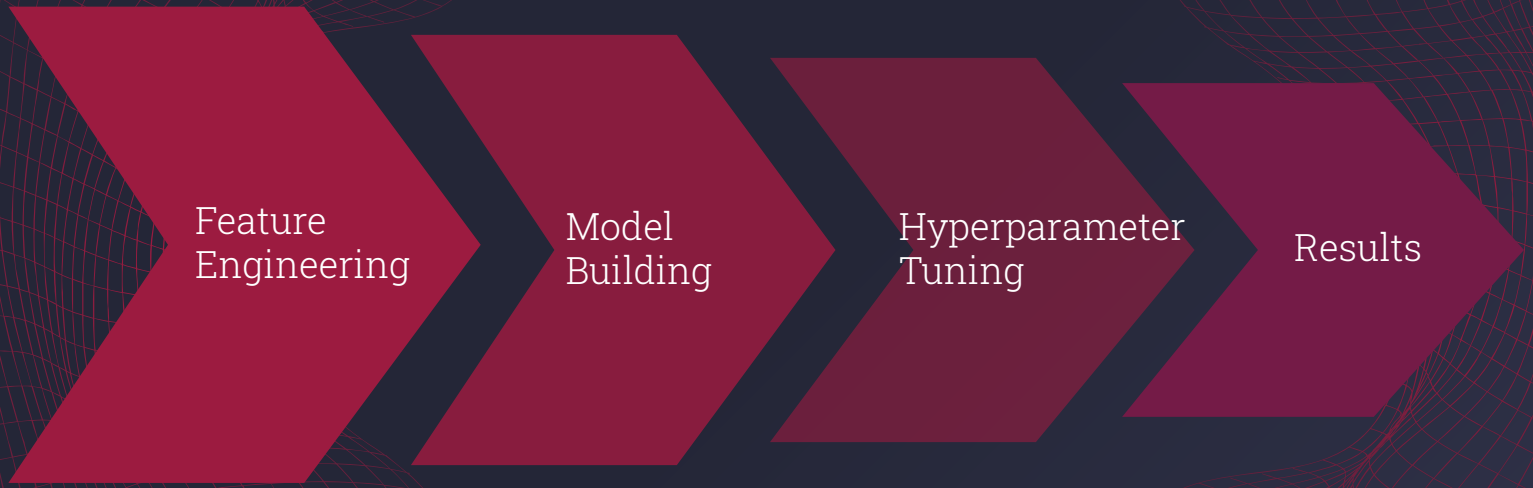
Video


Happy

SAMPLE DATA

04

DETECTING EMOTIONS FROM AUDIOS


Feature Engineering
Model Building
Hyperparameter Tuning
Results

05
FEATURE ENGINEERING

HOW DO YOU CONVERT AUDIO TO NUMBERS?

Pitch

Pixels

Spectral Centroid

Energy

????

Spectral Rolloff

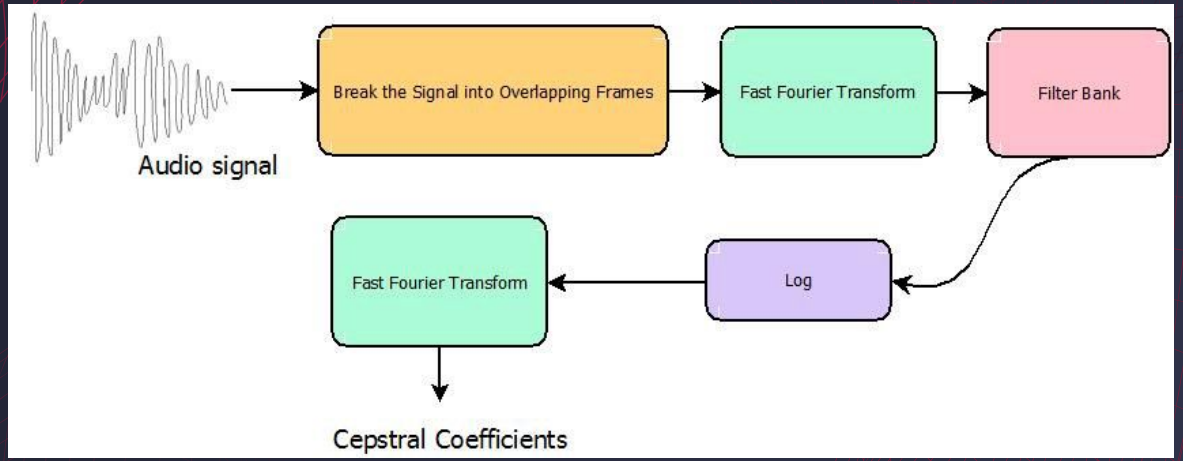
Zero Crossing Rate

MEL FREQUENCY CEPSTRAL COEFFICIENTS

But what does that mean? Something.

Any sound generated by humans is determined by the shape of their vocal tract (including tongue, teeth, etc). If this shape can be determined correctly, any sound produced can be accurately represented. The envelope of the time power spectrum of the speech signal is representative of the vocal tract and MFCC.

MFCC are widely used in speech recognition and speech emotion recognition studies



AUDIO PROCESSING

- 13 MFCCs are generated for each frame of the audio clip
- There are 156 frames in each clip
- These coefficients are aggregated by finding their mean, min and max
- We found that the 1st and 2nd order derivatives of these coefficients are also useful in detecting emotions.
- Similar aggregations were performed for the delta and delta coefficients
- Another feature extracted was the root mean squared energy

06 MODEL IMPLEMENTATION

Out of the 24 actors, we use 20 for training and isolated 4 for testing.
Randomly splitting the data would have caused a data leakage problem.

01

Neural Nets

Going for Gold

02

SVMs (SVC)

Surprisingly Effective Linear Boundaries

03

Random Forest

Bagging Features as You Go

04

XGBoost

The Kaggle Way

07

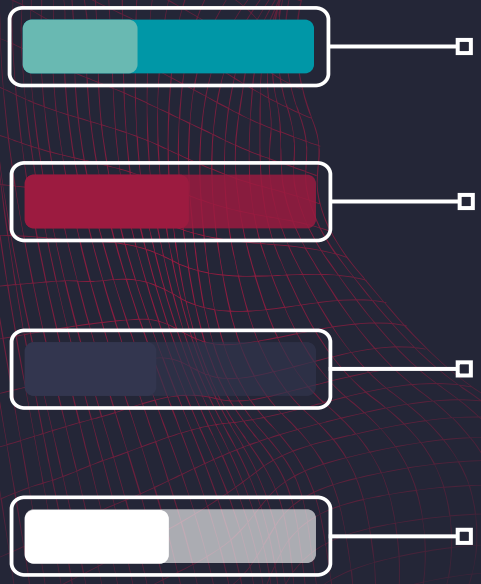
HYPER PARAMETER TUNING

Randomised Search - to narrow down options for GridSearch

Actual GridSearch to find the best parameters

Run the models on these parameters

MODEL ACCURACIES



Neural Network: 41.25%

SVM: 51.3%

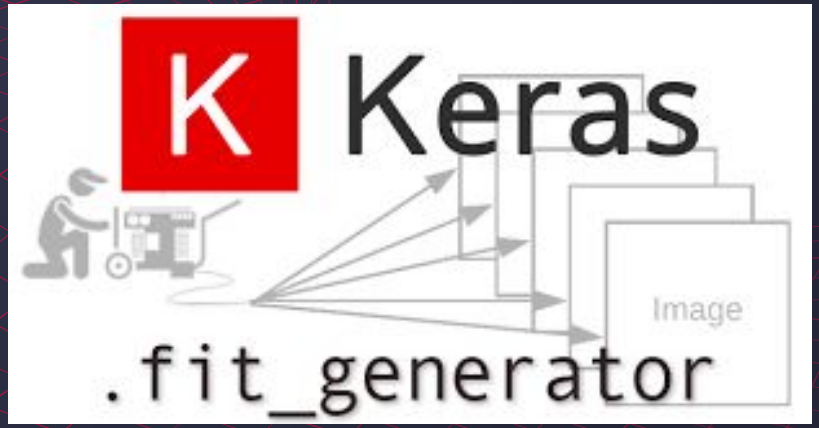
RandomForest: 48%

XGBoost: 50.0%

08

DETECTING EMOTIONS FROM VIDEOS

VIDEO FRAME PROCESSING

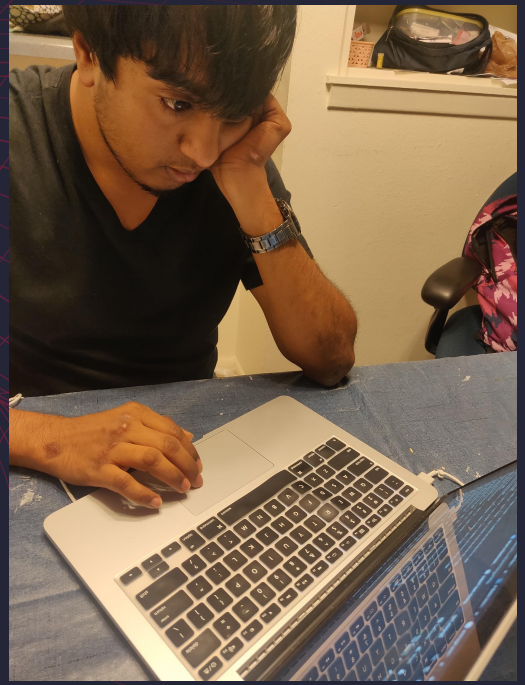


- 3 second video clips (.mp4) converted to a series of images captured frame-by-frame with openCV
- Every 5th frame is captured
- Reshaped images to a resolution 640x360 or 256x256 numpy array
- *fit_generator( )* is used to fit the frames to the model in batches due to memory limitations (>80Gb uncompressed!)

EMOTION DICTIONARY

09

MODEL IMPLEMENTATION



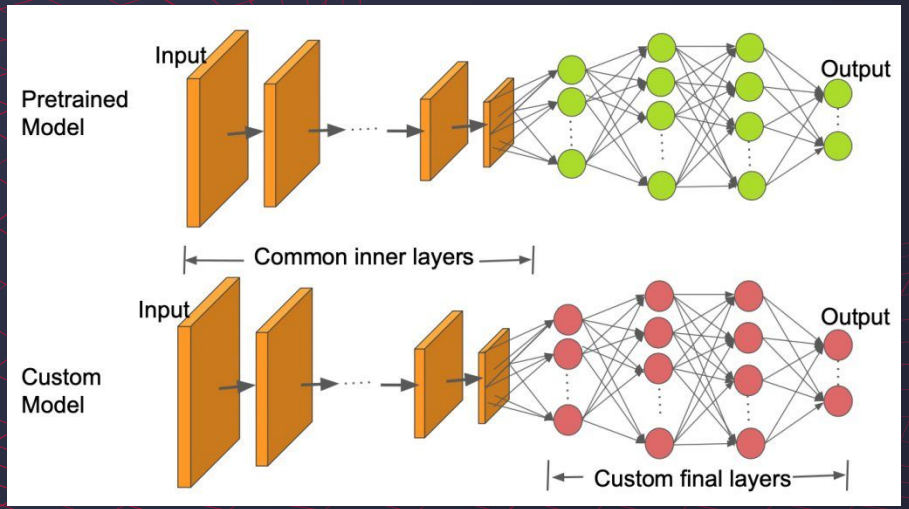
Model 1 : BUILT-FROM-SCRATCH

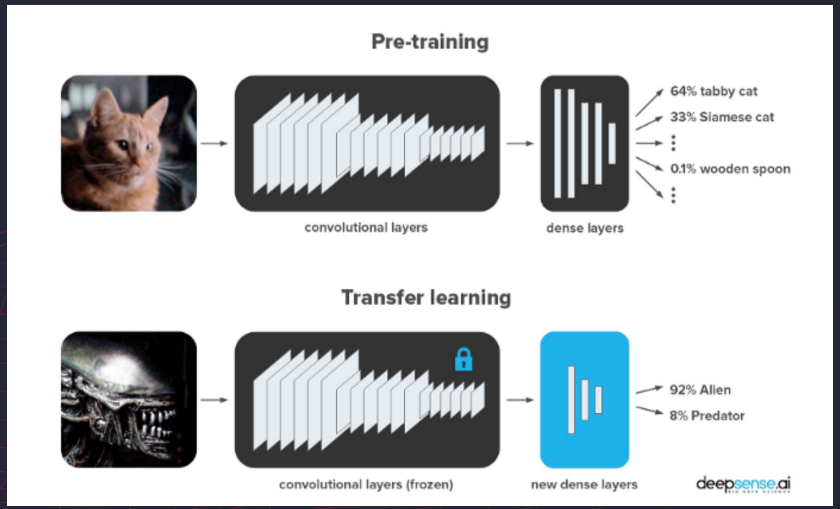
```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 359, 639, 8)       224
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 179, 319, 8)       584
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 88, 158, 8)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 88, 158, 8)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 111232)            0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                3559456
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_3 (Dense)              (None, 9)                 297
=================================================================
Total params: 3,560,561
Trainable params: 3,560,561
Non-trainable params: 0
```

- 2 Convolution Layers
- 8 nodes per layer
- Pooling in 9 pixel grids
- 2 dropouts to reduce overfitting to limited training data
- Trained on 22/24 actors

- Average Prediction Accuracy: 21%

Model II : TRANSFER LEARNING





Model II : TRANSFER LEARNING

```
Model: "sequential_11"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_54 (Conv2D)           (None, 46, 46, 32)        320
_________________________________________________________________
conv2d_55 (Conv2D)           (None, 44, 44, 64)        18496
_________________________________________________________________
max_pooling2d_27 (MaxPooling (None, 22, 22, 64)        0
_________________________________________________________________
conv2d_56 (Conv2D)           (None, 20, 20, 128)       73856
_________________________________________________________________
max_pooling2d_28 (MaxPooling (None, 10, 10, 128)       0
_________________________________________________________________
conv2d_57 (Conv2D)           (None, 8, 8, 128)         147584
_________________________________________________________________
max_pooling2d_29 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
conv2d_58 (Conv2D)           (None, 4, 4, 7)           903
_________________________________________________________________
conv2d_59 (Conv2D)           (None, 1, 1, 7)           791
_________________________________________________________________
flatten_10 (Flatten)         (None, 7)                 0
_________________________________________________________________
activation_10 (Activation)   (None, 7)                 0
=================================================================
Total params: 241,950
Trainable params: 241,950
Non-trainable params: 0
_________________________________________________________________
```

Original Model

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
model_3 (Model)              (None, 4, 4, 128)         240256
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 4, 4, 9)           1161
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 1, 1, 9)           1305
_________________________________________________________________
flatten_1 (Flatten)          (None, 9)                 0
_________________________________________________________________
activation_1 (Activation)    (None, 9)                 0
=================================================================
Total params: 242,722
Trainable params: 242,722
Non-trainable params: 0
_________________________________________________________________
```

Updated Model

Model III

1. Batch Normalization
2. Depthwise Separable Convolution
3. Global Average Pooling
4. Dropouts

Model: "model_3"

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	(None, 256, 256, 1)	0
conv2d_91 (Conv2D)	(None, 254, 254, 16)	160
conv2d_92 (Conv2D)	(None, 252, 252, 16)	2320
batch_normalization_13 (Batc	(None, 252, 252, 16)	64
zero_padding2d_5 (ZeroPaddin	(None, 254, 254, 16)	0
conv2d_93 (Conv2D)	(None, 252, 252, 32)	4640
conv2d_94 (Conv2D)	(None, 250, 250, 32)	9248
batch_normalization_14 (Batc	(None, 250, 250, 32)	128
conv2d_95 (Conv2D)	(None, 248, 248, 64)	18496
conv2d_96 (Conv2D)	(None, 246, 246, 64)	36928
batch_normalization_15 (Batc	(None, 246, 246, 64)	256
max_pooling2d_38 (MaxPooling	(None, 123, 123, 64)	0
zero_padding2d_6 (ZeroPaddin	(None, 125, 125, 64)	0
conv2d_97 (Conv2D)	(None, 123, 123, 128)	73856
conv2d_98 (Conv2D)	(None, 122, 122, 128)	65664
conv2d_100 (Conv2D)	(None, 121, 121, 128)	65664
batch_normalization_17 (Batc	(None, 121, 121, 128)	512
flatten_14 (Flatten)	(None, 1874048)	0
dense_9 (Dense)	(None, 32)	59969568
dropout_6 (Dropout)	(None, 32)	0
dense_10 (Dense)	(None, 32)	1056
dropout_7 (Dropout)	(None, 32)	0
dense_11 (Dense)	(None, 8)	264

Total params: 60,248,824
Trainable params: 60,248,344
Non-trainable params: 480

Input layer of 256*256
Epochs : 100
Steps per epoch : 25
Prediction Accuracy : 0.48

IO

DEMO

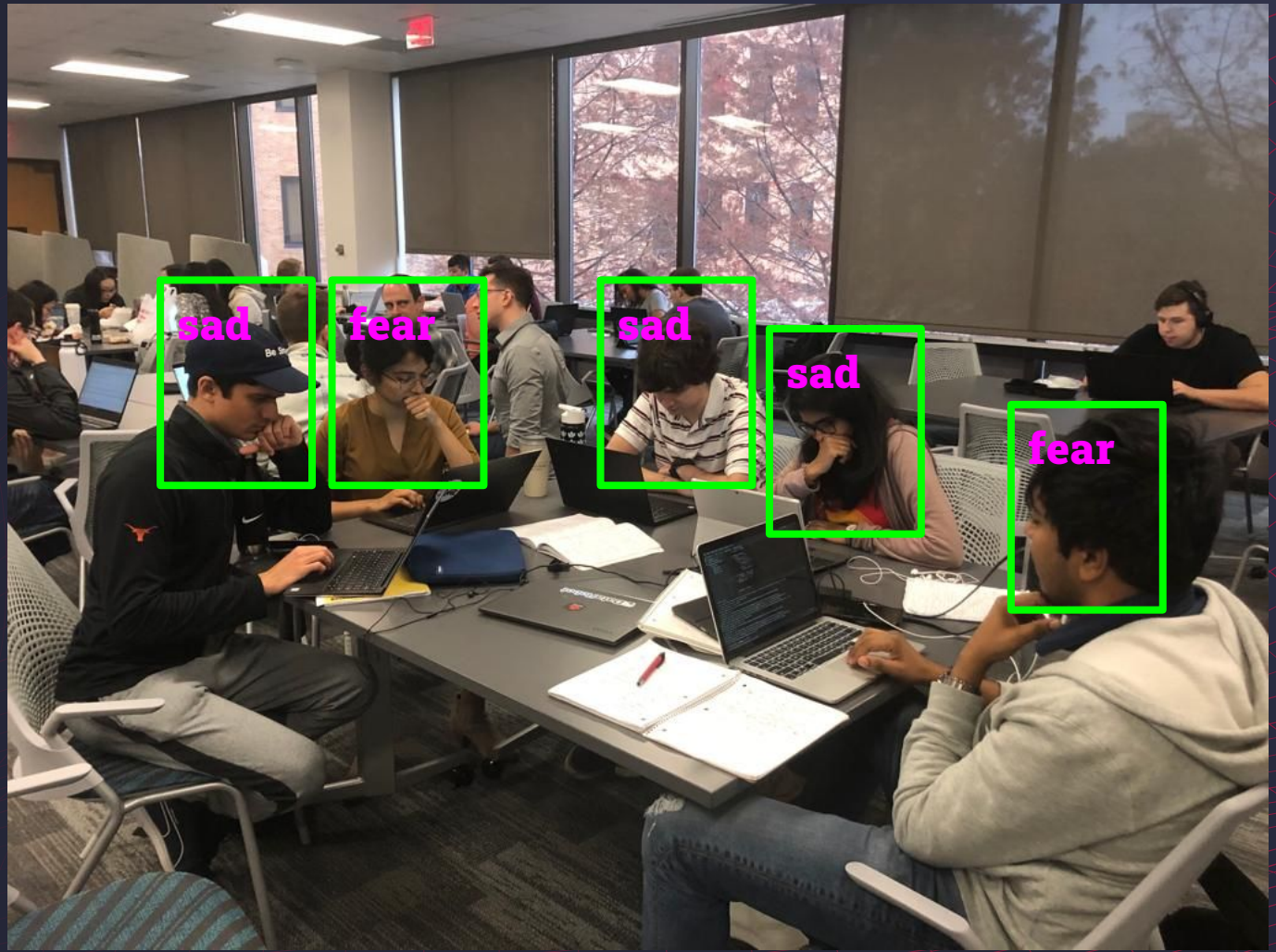
Can it read OUR emotions?

II

FUTURE SCOPE

FUTURE SCOPE

- Combine the video & audio predictive models for better accuracy & performance on test data
  - “Ensemble of two”
- Explore more models (CNN using transfer learning for audio, audio feature engineering, student teacher networks - using the video model to train audio)
- Obtain data from different geographic regions to eliminate bias towards the north american population
- Current data is collected in a highly controlled environment
  - Therefore, obtain data from real-life settings and attach class labels with the Amazon Turk service


sad
fear
sad
sad
fear