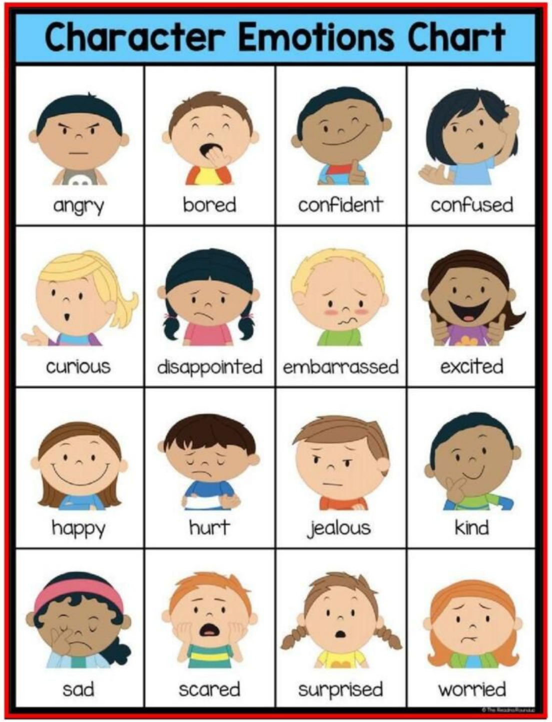
Multi-modal Emotion Detection and Classification using Audio-Visual Data

Violet Li, Roxy Rong, Weijie Yang

Berkeley
UNIVERSITY OF CALIFORNIA

Problem statement

This project aims to solve the **emotion classification problem** from both **visual** and **audio** data, without having the knowledge of the speaker information.



Dataset: SAVEE Database

- The SAVEE database was recorded from 4 native English male speakers.
- There are **7 categories of emotions**: anger, disgust, fear, happiness, sadness, surprise and neutral.
- It contains **videos** of their facial expressions and audios of them reading emotion-specific sentences.

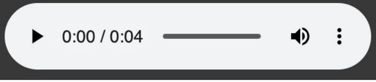
Audio Sample:
Anger

Audio Experiments

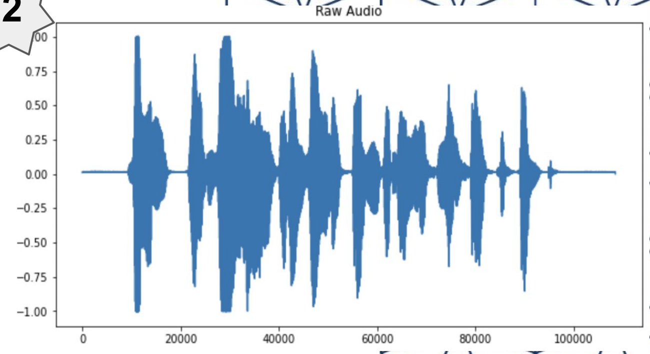
Audio Features Extraction

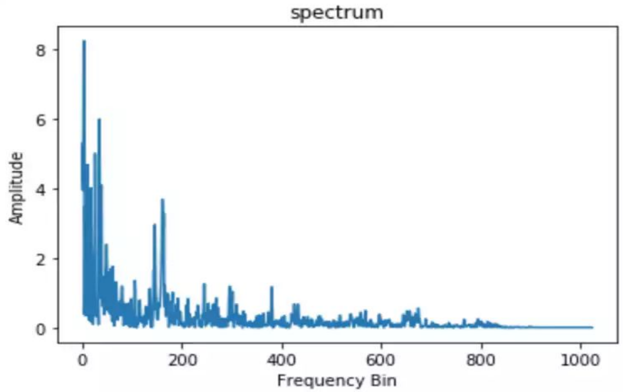
1

```
from IPython.display import Audio, display
display(Audio(list(example_data.data), rate=sr, autoplay=True))
```

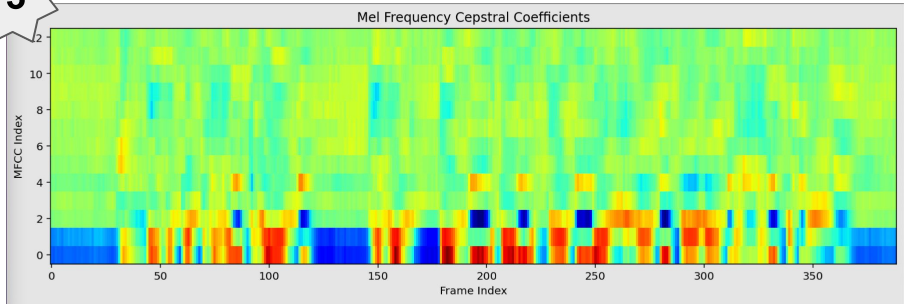


2





3

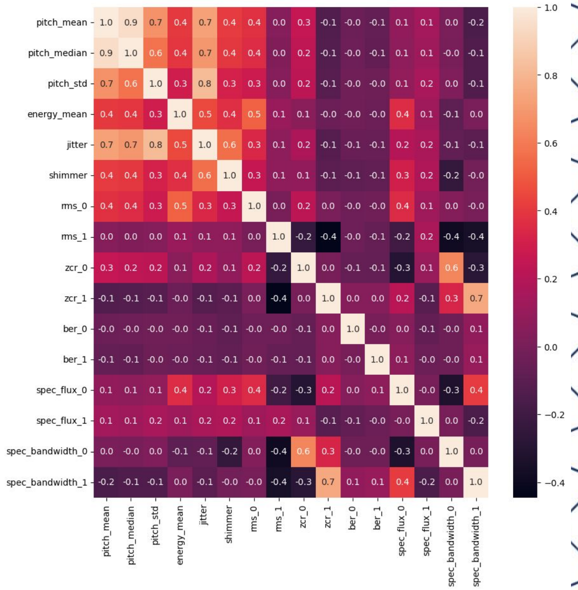


* Time period, frequency, amplitude

ML Models - Audio

Audio Feature Selection

- Basic features + PCA on 1 dimensional array features -> 40+ features
- Correlation matrix to reduce it to 16 features
- All inputs are normalized



Model Architecture

- Traditional ML model: LogReg, SVM, RF, KNN
- CNN Model (MFCC -> 2 layers Conv2D, Kernel = (5, 15), (3, 9))
- Dual CNN Model & CNN + other features Model
- Use Early Stopping, Regularization, Kernel Constraints to prevent overfitting with more complicated models

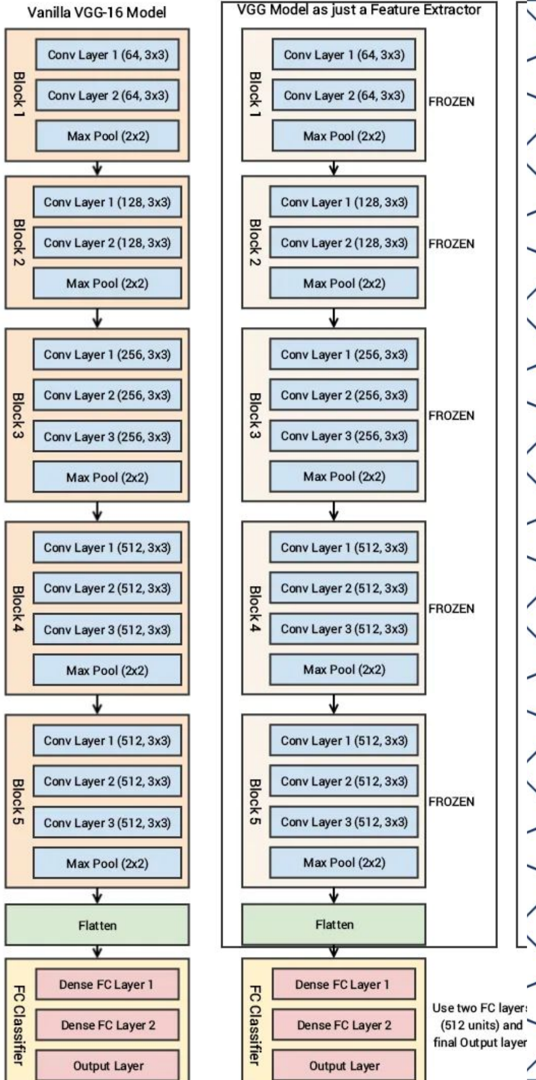
Transfer Learning

Why Transfer Learning

- As we have a **small dataset**, transfer learning allow use start from a well-trained model.
- Reduce computational time & dataset needed dramatically
- Only train the new added dense layers

Steps

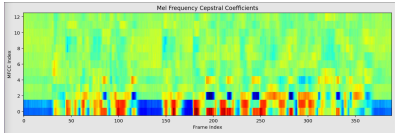
- Determine which base model to use.
- Preprocess the dataset so that it is a valid input.
- Train the model

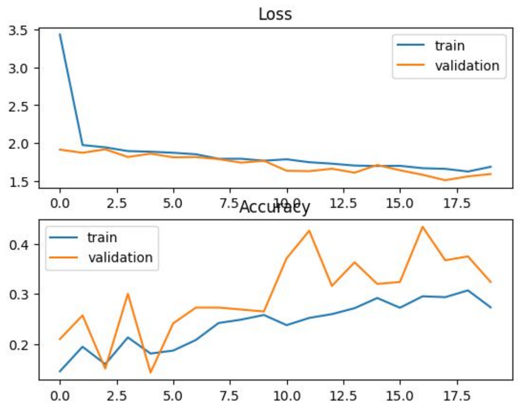


Audio: Model based on VGG19 / YAMNet

MFCC Diagram with VGG19

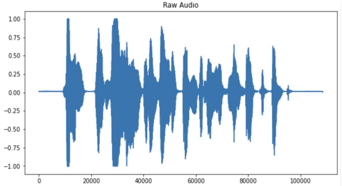
- convolutional neural network architecture consisting of 19 layers
- Take images as input.
- Test Accuracy: 0.39

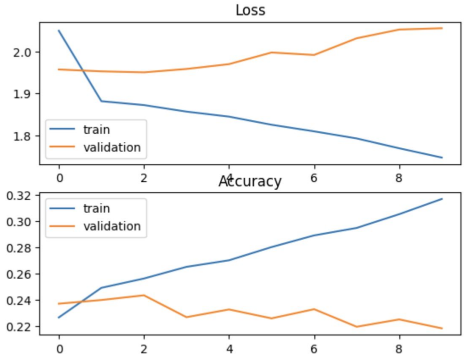




Waveform with YAMNet

- A pre-trained neural network that employs the MobileNetV1, classifying audio types.
- Use audio waveform as input.
- Over-fitting as we increase epochs.
- Test Accuracy: 0.21





Why Transfer Learning is not working so well

- Though transfer learning needs fewer samples, 480 original samples are still too few to train a good model.
- Not ideal base model selection.
    - VGG19 model is more for classifying images
    - YAMNet for classifying types of phonating objects (cars, animals, etc.).
    - Our samples will likely all get ‘speech’ embeddings without obvious differentiation.

```
print(f'The main sound is: {inferred_class}')
print(f'The embeddings shape: {embeddings.shape}')
```

```
The main sound is: Speech
The embeddings shape: (10, 1024)
```

- To improve the results
    - Extend the dataset.
    - Not only use models as embeddings extractor, but fine-tune some existing layers

Model Performance Evaluation Overview (audio only)

	Model	Accuracy	Precision	Recall	f1_score
	Log Reg	0.385	0.37	0.38	0.37
	SVM	0.447	0.43	0.45	0.44
	RF	0.464	0.43	0.46	0.42
	KNN (MFCC)	0.397	0.37	0.40	0.38
NEW	**CNN**	**0.505**	**0.49**	**0.50**	**0.49**
NEW	**Dual CNN**	0.422	0.48	0.42	0.44
NEW	**CNN + others**	**0.552**	**0.53**	**0.55**	**0.53**
NEW	**Transfer Learning (VGG19)**	0.389	0.35	0.39	0.39
NEW	**Transfer Learning (YAMNet)**	0.210	0.11	0.21	0.10

Limitations on Audio

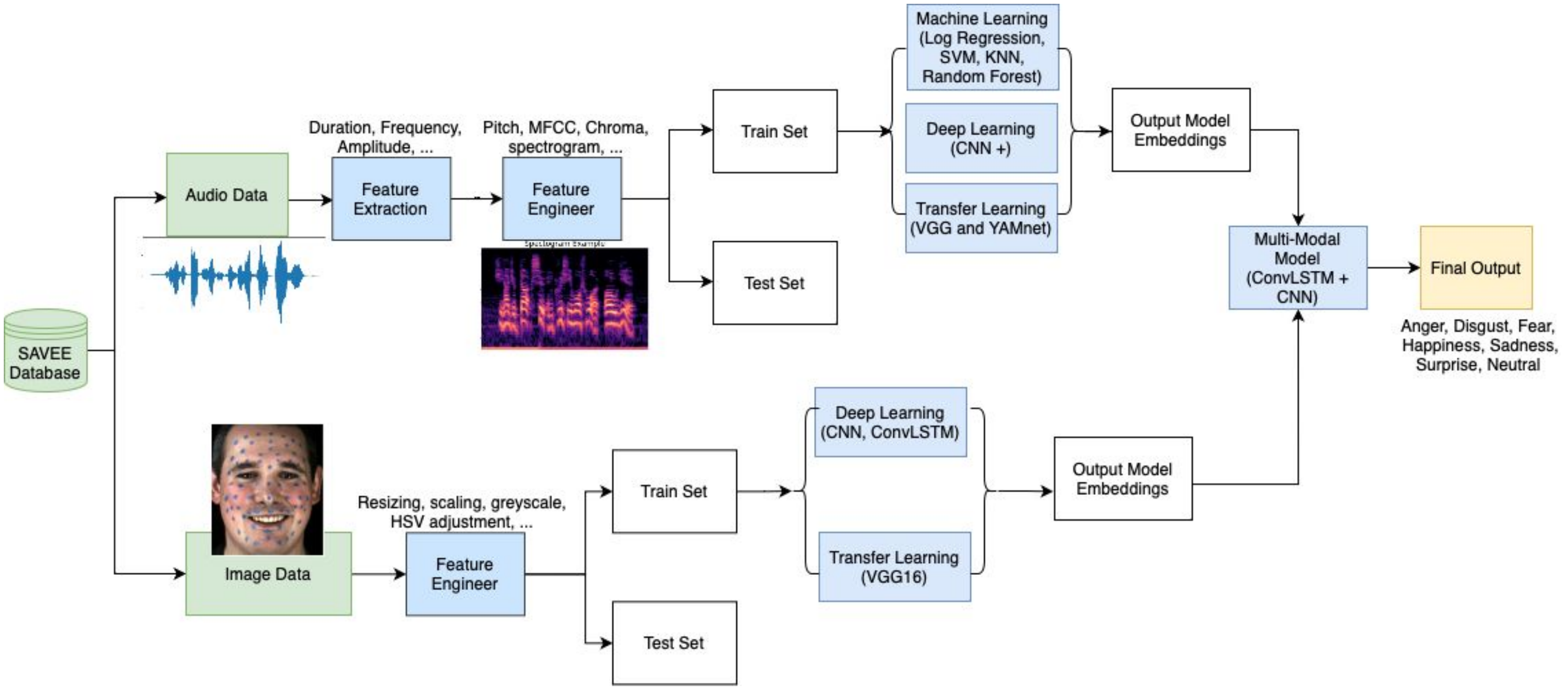
Weakness:

- Audio Model can’t separate **disgust** or **sad** from **neutral** well, also can’t distinguish between **fear** and **surprise** well
- Overall performance is still too low, not achieving 80% accuracy goal

Next Steps:

- Add visual data
- Build multimodal fusion

Flowchart



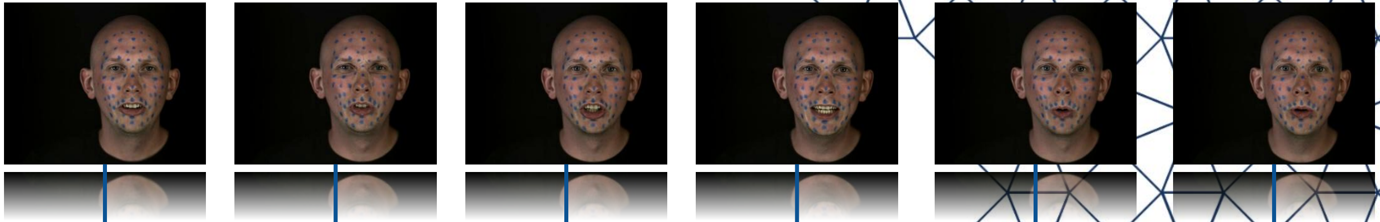
SAVEE Database
Audio Data
Duration, Frequency, Amplitude, ...
Feature Extraction
Pitch, MFCC, Chroma, spectrogram, ...
Feature Engineer
Train Set
Test Set
Machine Learning (Log Regression, SVM, KNN, Random Forest)
Deep Learning (CNN +)
Transfer Learning (VGG and YAMnet)
Output Model Embeddings
Multi-Modal Model (ConvLSTM + CNN)
Final Output
Anger, Disgust, Fear, Happiness, Sadness, Surprise, Neutral
Image Data
Resizing, scaling, greyscale, HSV adjustment, ...
Feature Engineer
Train Set
Test Set
Deep Learning (CNN, ConvLSTM)
Transfer Learning (VGG16)
Output Model Embeddings

Image Experiments

Video Feature Extraction

Video

Step 1: Extract 6 Frames for each video with sequences



Step 2: Face Detection by DNN ResNet OpenCV

Step 3: Central Cropping based on recognition result

Image: Model based on VGG16

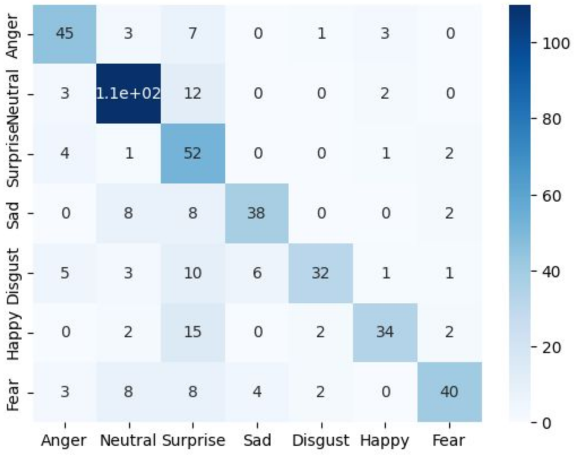
VGG16:

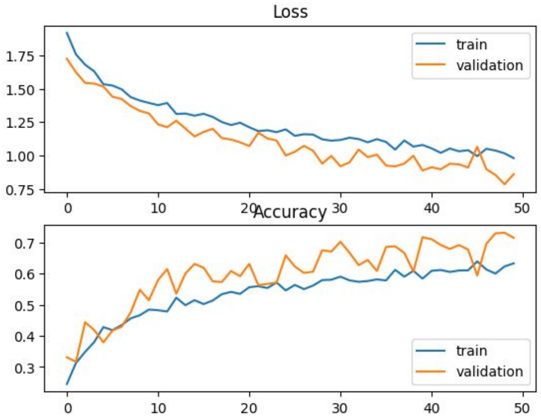
- convolutional neural network architecture consisting of 16 layers
- Trained on ImageNet, which contains more than 14 million training images across 1000 object classes.

Test Accuracy achieved after 50 epochs: **0.73**

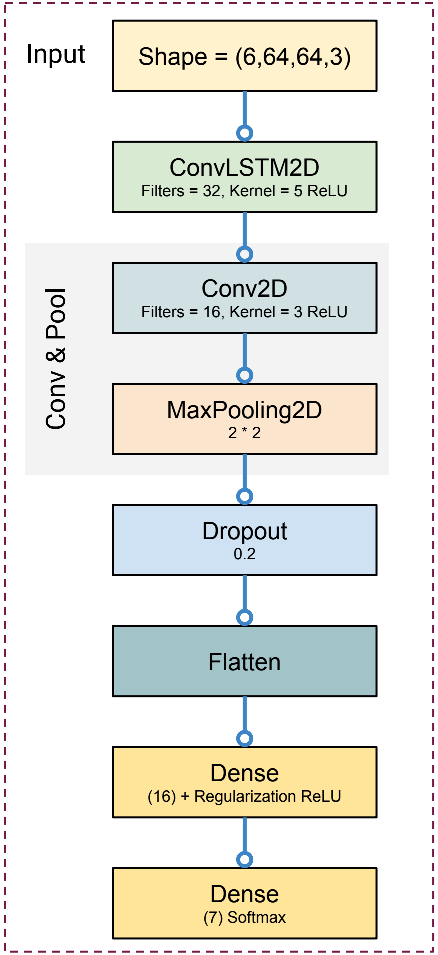
Next Steps:

- Build a model that can make use of the temporal information

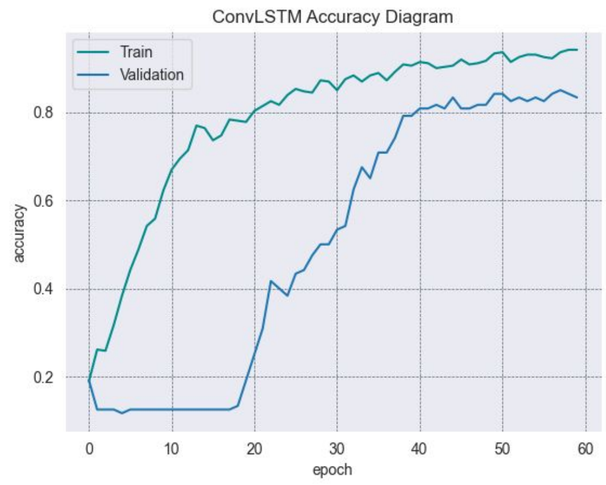




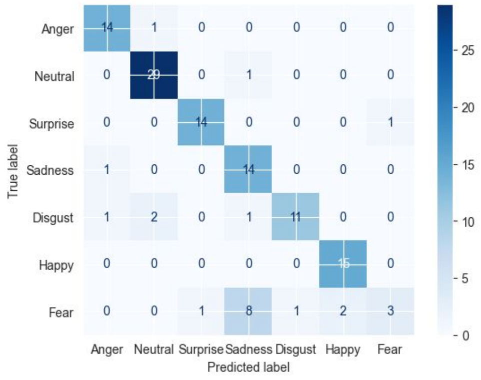
ConvLSTM on Visual Inputs



Training Result	
Training Accuracy	0.9417
Test Accuracy	0.8333
precision	0.84
recall	0.81
f1-score	0.79



Confusion Matrix



Multimodal Fusion

- ❖ Concatenate both video and audio extractors into Neural Networks.

Model Input Processing

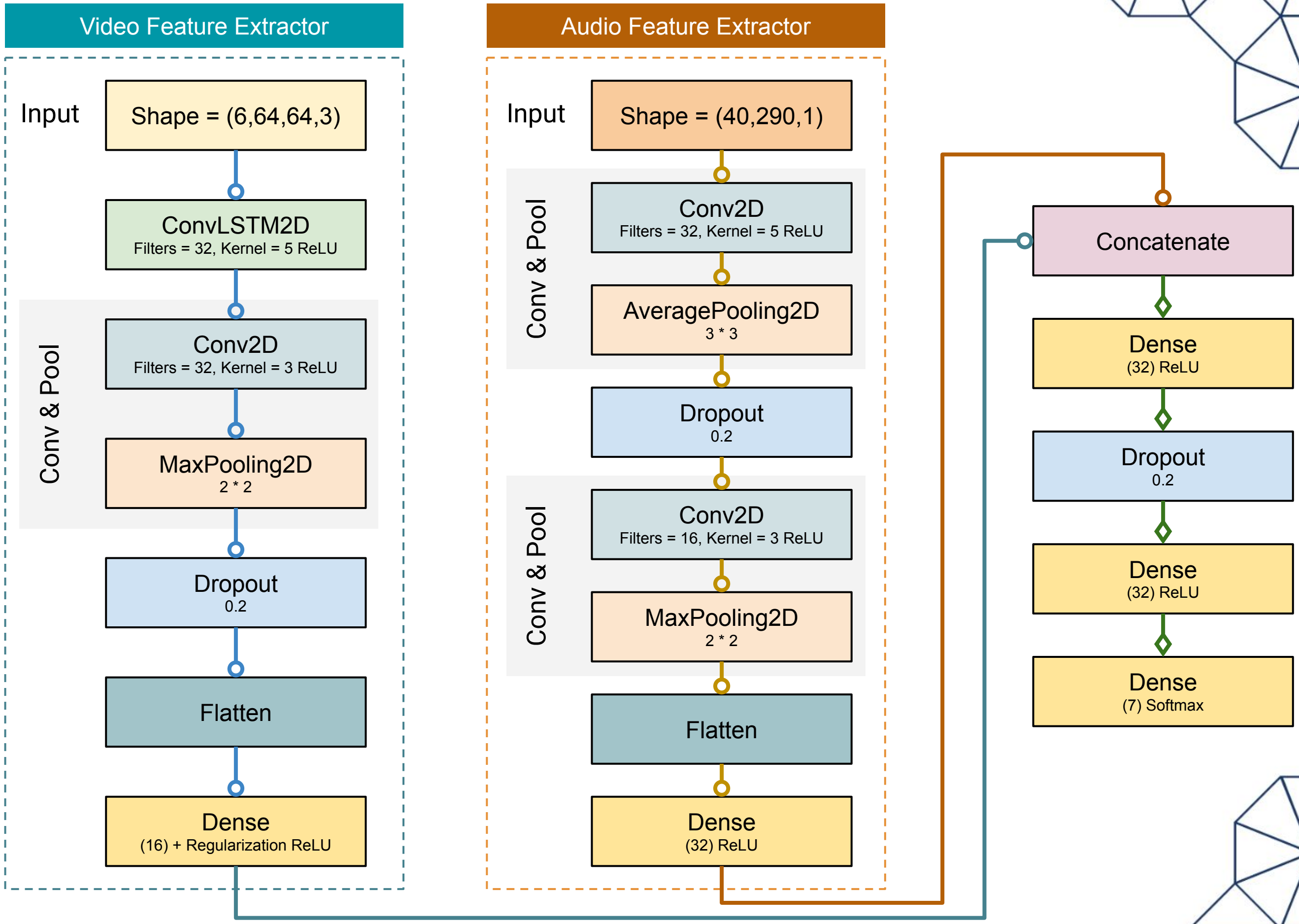
Video & Audio Feature Processing

- Video Input Size : (6, 64, 64 ,3) <6 frames with 64*64 pixels RGB images>
- Audio Input size : (40, 290, 1) <MFCC with 40*290 pixels non-RGB images>
- All inputs are normalized

Train/Test Split Technique

- Audio Dataset is augmented → Split the dataset by group
- Match video input with audio input
- Balance the portion of different emotion inputs → Weights Setting
- Cross Validation → Stratified Group K Fold (K = 4)

MultiModal Architecture



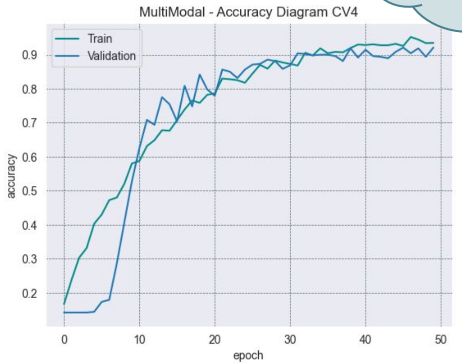
Training Result Overview - Multimodal Fusion

- Split dataset by Grouping Augmentations
- Perform Cross Validation
- Train model 4 times and get average performance

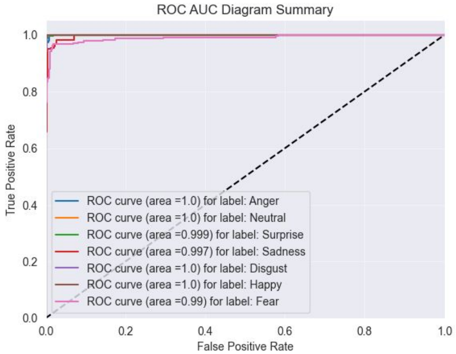
97% Accuracy

CV Training Accuracy	0.960 +/- 0.015	CV Training Loss	0.484 +/- 0.044
CV Testing Accuracy	0.966 +/- 0.028	CV Testing Loss	0.436 +/- 0.070

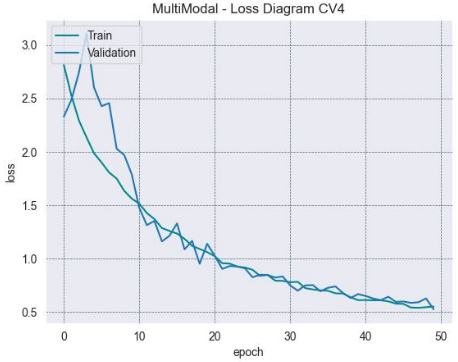
Accuracy Diagram



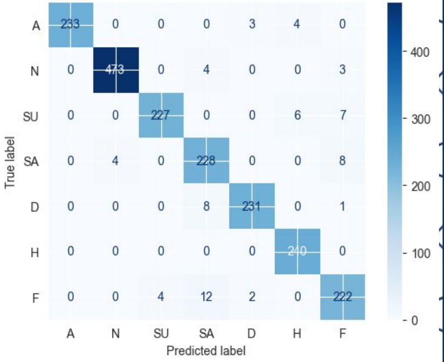
ROC AUC Diagram



Loss Diagram



Confusion Matrix



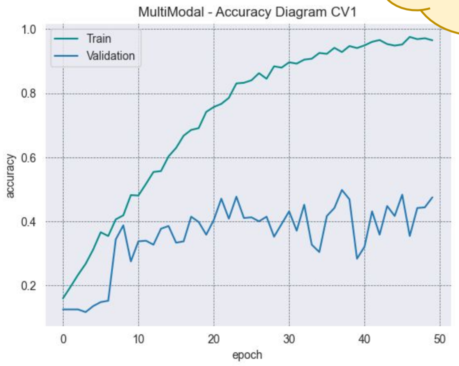
Predicting new faces - Multimodal Fusion

- Split dataset by Speakers (DC, JK, KE, KL)
- Perform Cross Validation (E.g. Train: DC, JK, KE | Test: KL)
- Model learns 3 faces and predict one new face

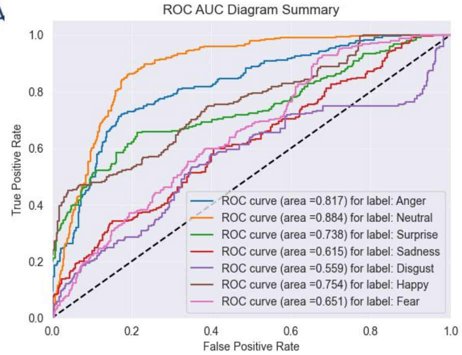
52% Accuracy

CV Training Accuracy	0.945 +/- 0.037	CV Training Loss	0.362 +/- 0.084
CV Validation Accuracy	0.517 +/- 0.123	CV Validation Loss	2.038 +/- 0.086

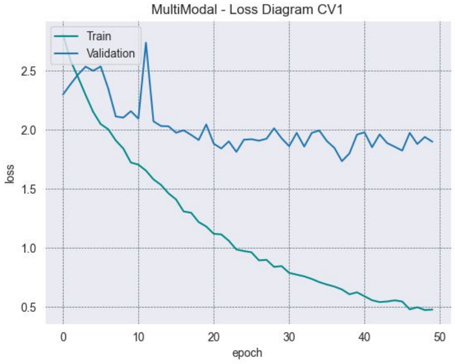
Accuracy Diagram



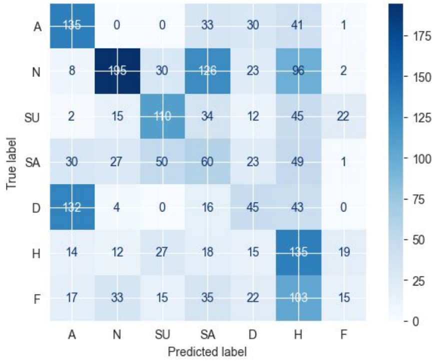
ROC AUC Diagram



Loss Diagram



Confusion Matrix



Model Performance Evaluation

Limitations (1-2mins)

- Dataset Size: Not enough video data (only 4 faces, and all white male) make it harder to generalize
- Tuning: More complicated model structure makes hyper-param tuning much harder
- AI Fairness: All speaker are white males. This means that our sample contains bias on age, race, gender, and education level.

Discussion & Conclusion

Result

- ConvLSTM Multi-modal fusion has the best performance.
- Potential reason: Emotion expression is a temporal behavior with visual and audio outputs. Multi-Modal can better capture temporal info and both features.

Reflections

- Sample selections

Future Work

- Add more samples and be more inclusive

Appendix

Berkeley
UNIVERSITY OF CALIFORNIA

A list of features:

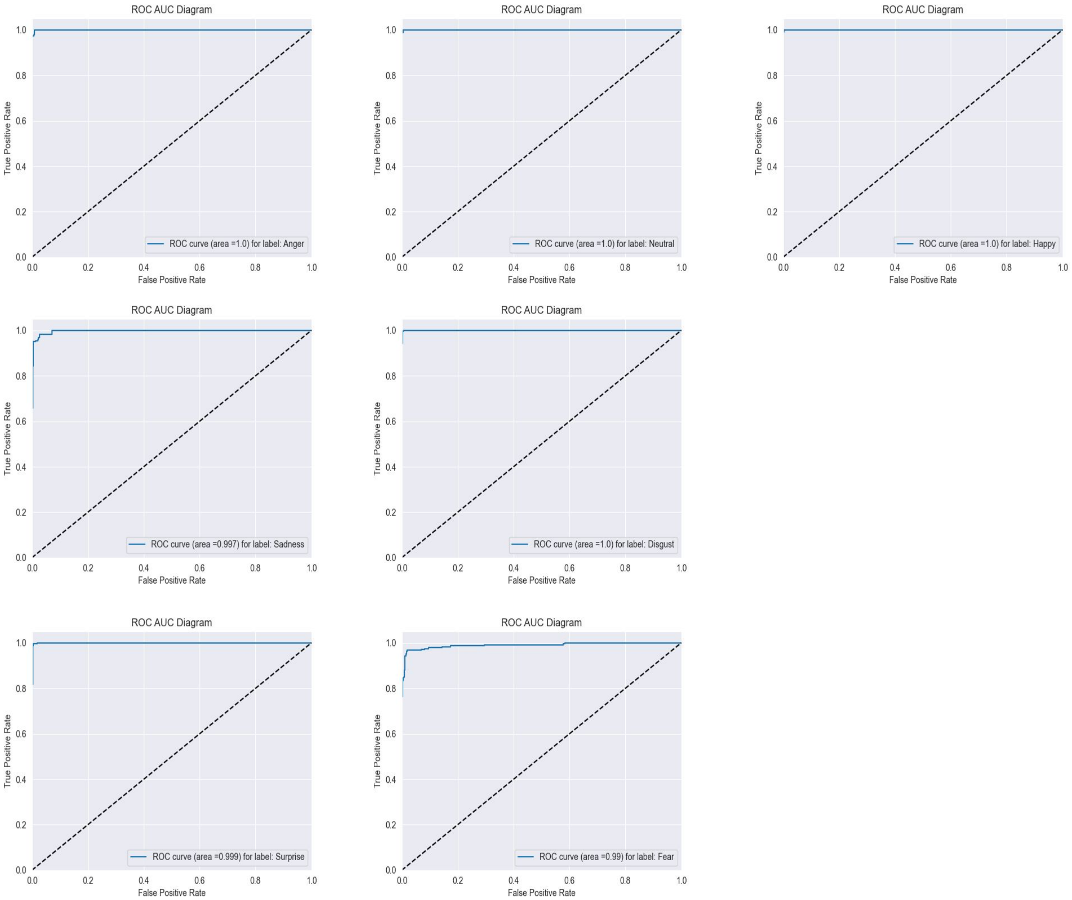
f0
pitch_mean, pitch_median, pitch_std, pitch_range, pitch_max
energy_mean, energy_median, energy_std, energy_range, energy_max
jitter, shimmer
amplitude_envelope
rms
zcr
ber
amplitude_spectrogram
dB_spectrogram
spec_contrast
spec_flux
spec_centroid
spec_bandwidth
spec_flatness
power_spectrogram
mel-spectrogram
mfcc
mfcc_delta
teo

Data Augmentation

Original file

- Only 480 samples, so we need to augment the dataset.
- Methods used:
    a. Add Gaussian noise
    b. Pitch Scaling
    c. Time Stretching
    d. Random Gain
    e. Invert Polarity
- Randomly combine these methods to obtain augmented dataset.

Performance Overview - Multimodal Fusion



Model Selection