SPEECH EMOTION RECOGNITION FOR سنڌي LANGUAGE

Present by: Maaz-Ullah, Ali Raza, Anas Bin Yousuf
Group 17

TODAY'S AGENDA

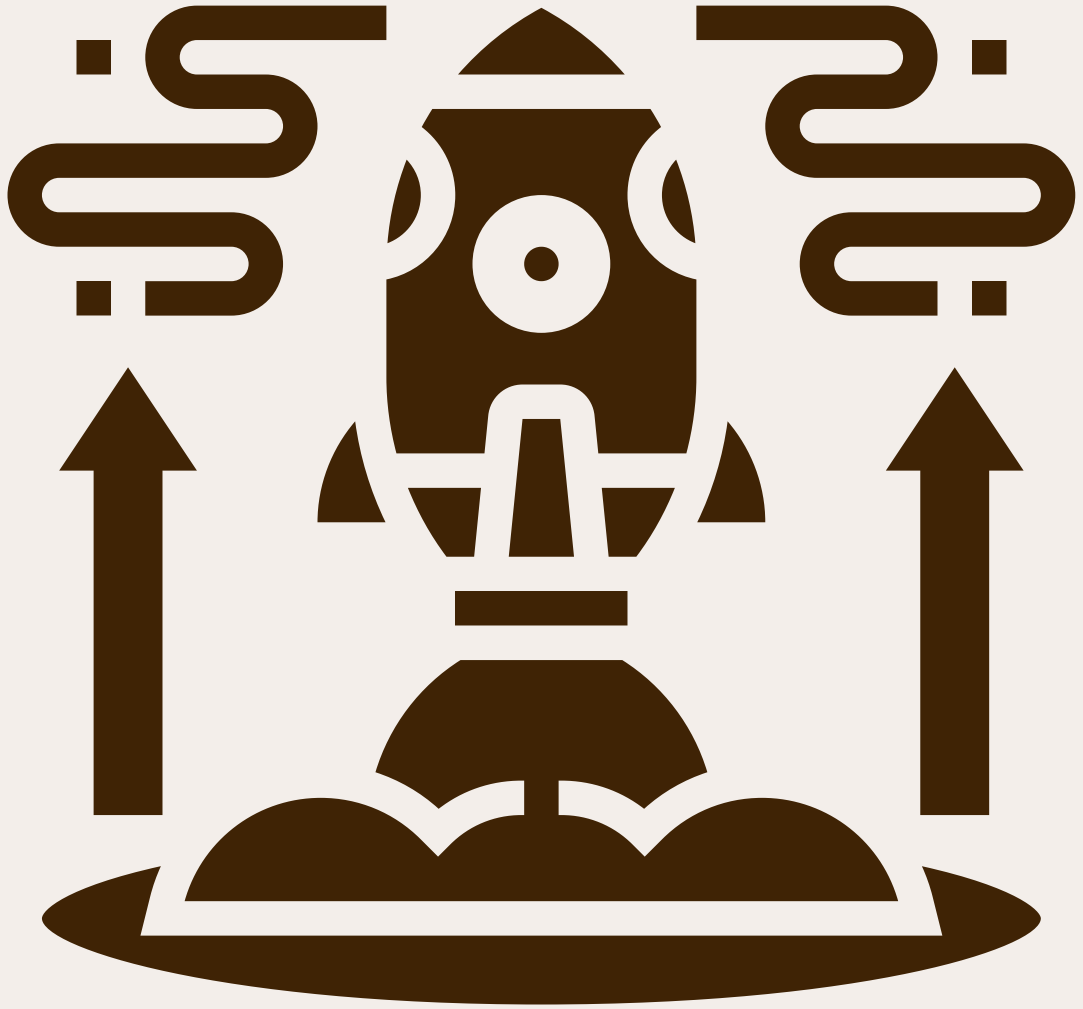
INTRODUCTION

WHAT ARE WE DOING?

- Develop a speech emotion recognition system for Sindhi language using deep learning techniques.
- Emotion Classes: **Happy**, **Sad**, **Angry**, **Neutral**.
- Data Collection: Audio data collected through WhatsApp from native Sindhi speakers and an Existing Corpus.
- Techniques Used: Deep learning models– Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs).

- How can machine learning models be optimized for accurate emotion recognition in low-resource languages like Sindhi using extracted acoustic features?
- How well can emotion recognition models trained on Urdu, English, or German Speech Emotion Corpus generalize to low-resource languages like Sindhi?

OUR MOTIVATION



Low Resource Language

Existing emotion recognition models often focus on widely spoken languages like English, leaving a gap for languages like Sindhi.

Potential Beneficiaries

The Sindhi population in Pakistan, specially in Sindh is very high. Contribution towards the Sindhi language in the field of computer science has the potential to positively affect a large number of people.

Scarcity of Labeled Data

Emotion recognition in speech requires large datasets with labeled emotional categories (e.g., happy, sad, angry). For many languages, including Sindhi, these labeled datasets are scarce, making it challenging to train accurate models.

DATA-SET ACQUISITION

Reaching Out via Social Media

We reached out to native Sindhi speakers via WatsApp. The speakers include friends, family and myself.

Existing FeatureSets and Scraping.

The Urdu-Sindhi Speech Emotion Corpus is a dataset collected at Mehran University of Engineering & Technology. The dataset contains per-processed feature sets of the audio samples collected.[1]. We also scraped audio from Sindhi Dramas from YouTube.

Data Augmentation

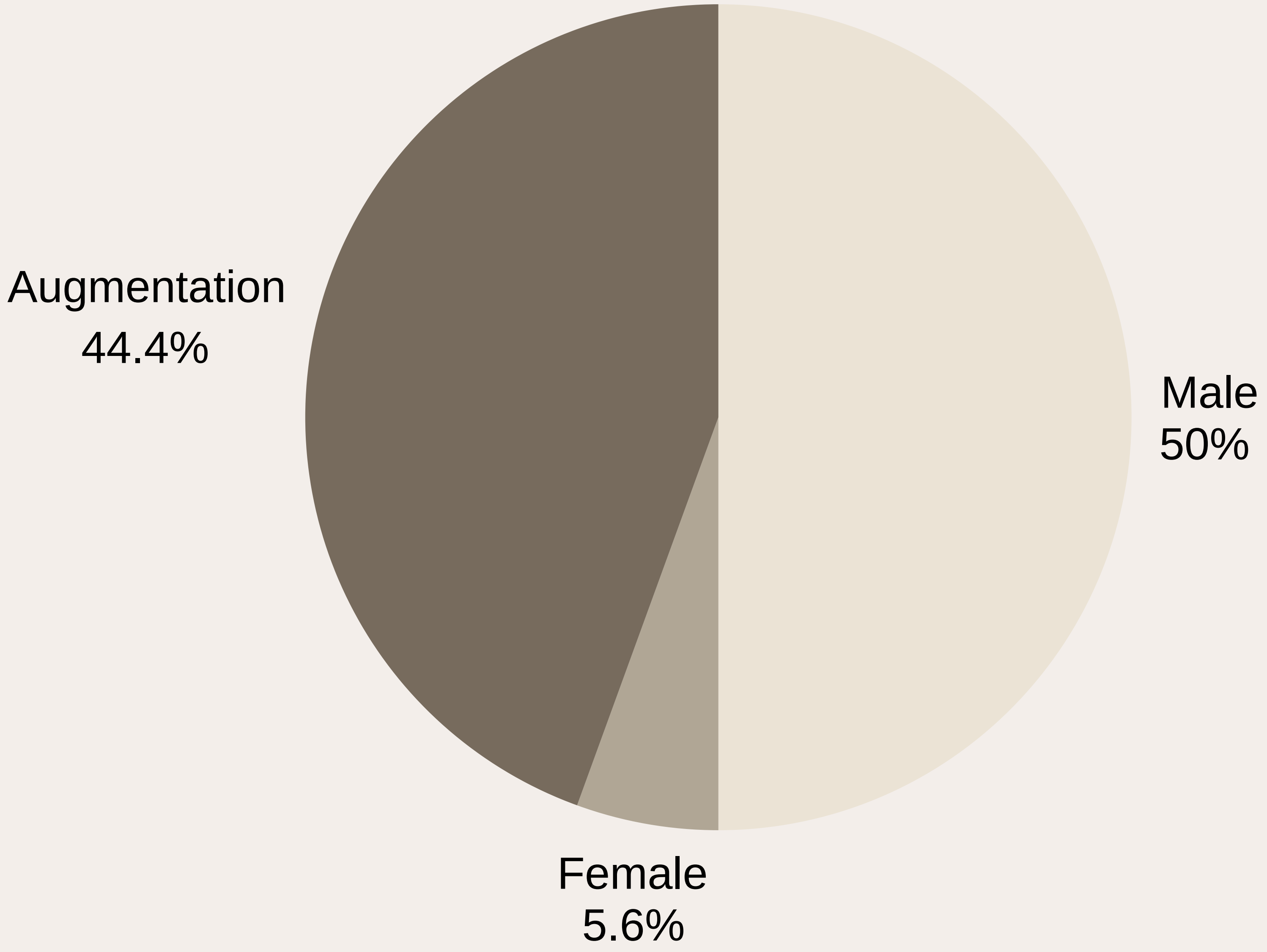
We applied data augmentation techniques on the collected voice samples to introduce variety and increase the number of samples.

- Time Stretching
- Pitch modulation
- White-noise

FINAL DATASET

Our final Data-set contains a wide variety of data samples:
The Breakdown is given below:

13 males && 250 samples	45%
5 Females && 50 samples	5%
Augmented Data: ~400 samples	40%



GETTING THE DATA READY

Prepocessing

Feature Extraction: Extracted key features using librosa:

- Zero Crossing Rate (ZCR): Measures signal fluctuations.
- Root Mean Square Energy (RMSE): Indicates signal energy.
- MFCCs: Encodes spectral properties of audio

Class Balancing

Oversampling to address class imbalance (e.g., more examples of "Angry" and "Sad").

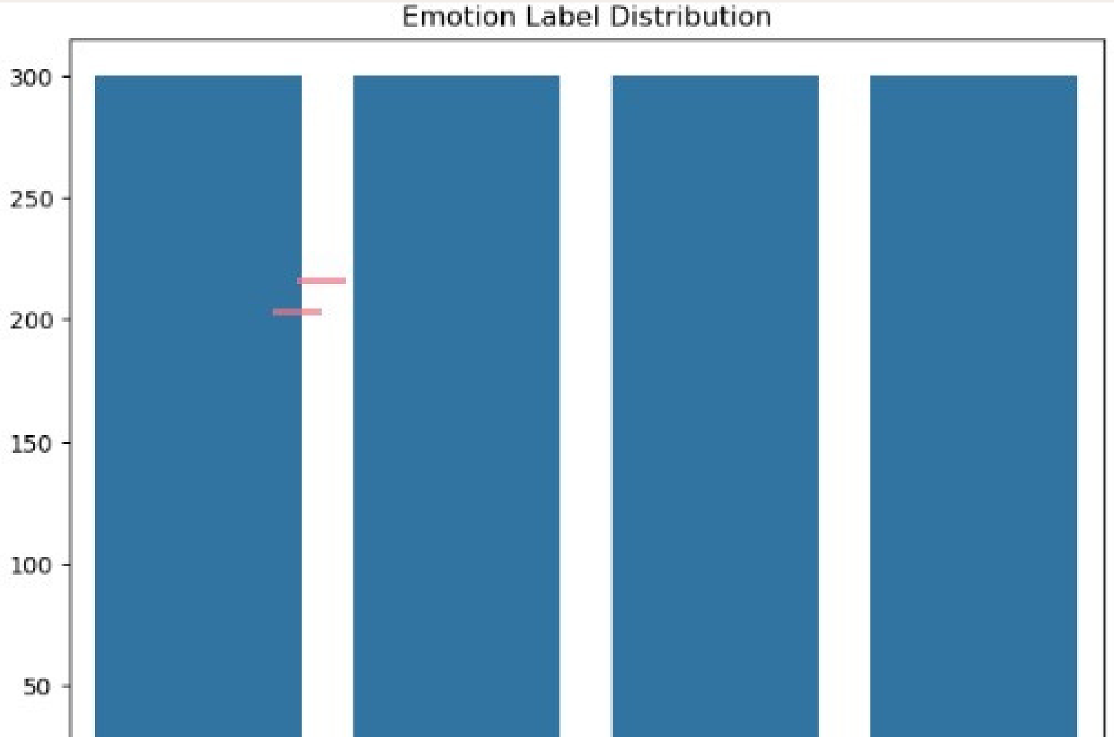
Inconsistent Formats

- Collecting Data Online Led To Inconsistent Formats eg. (.ogg,.wav,.mp3,.oppus etc)
- Files stored in emotion-specific directories.

CNNS

```
Loaded features with shape: (1200, 30)
Loaded labels with shape: (1200,)
Label encoding:
Angry: 0
Happy: 1
Neutral: 2
Sad: 3
Feature mean (after scaling): [-8.93729535e-17  1.96787031e-16 -8.45619870e-17  1.94659104e-16
 -1.52348042e-15] (truncated)
Feature std (after scaling): [1. 1. 1. 1. 1.] (truncated)
Training set size: 960 samples
Validation set size: 120 samples
Testing set size: 120 samples

Processed data saved as .npy files.
```

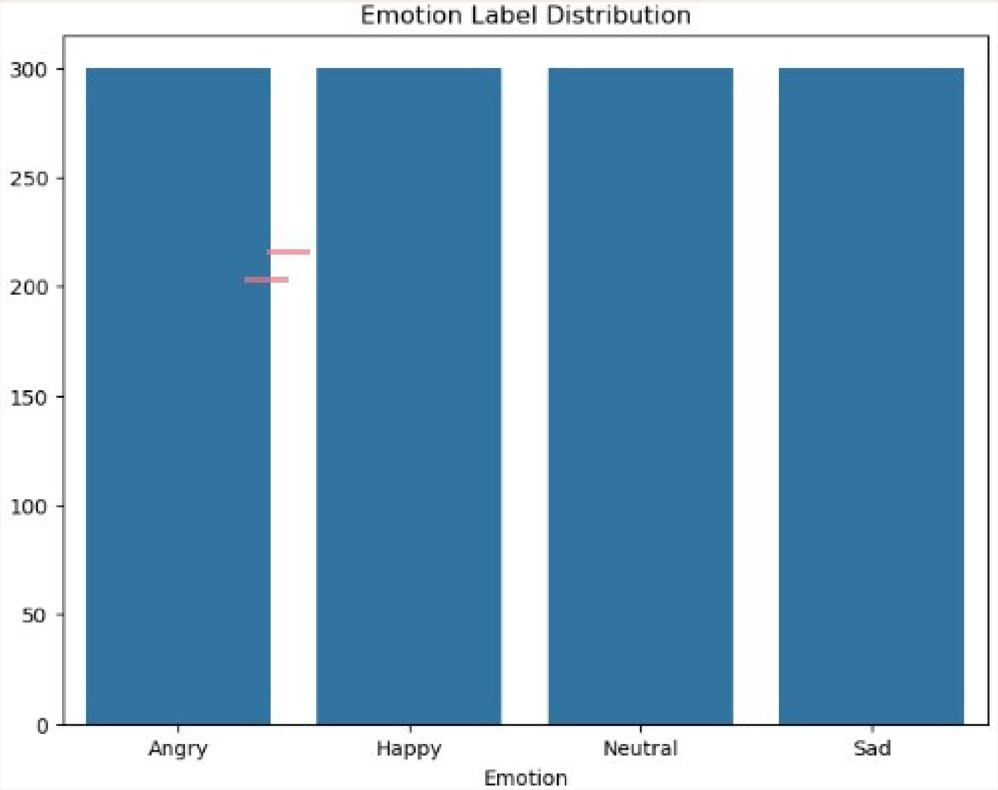


Training Samples:	960
Test Samples	120
Feature Size	30/sample
No. of Classes	4

DNNS

```
Loaded features with shape: (1200, 30)
Loaded labels with shape: (1200,)
Label encoding:
Angry: 0
Happy: 1
Neutral: 2
Sad: 3
Feature mean (after scaling): [-8.93729535e-17  1.96787031e-16 -8.45619870e-17  1.94659104e-16
 -1.52348042e-15] (truncated)
Feature std (after scaling): [1. 1. 1. 1. 1.] (truncated)
Training set size: 960 samples
Validation set size: 120 samples
Testing set size: 120 samples

Processed data saved as .npy files.
```

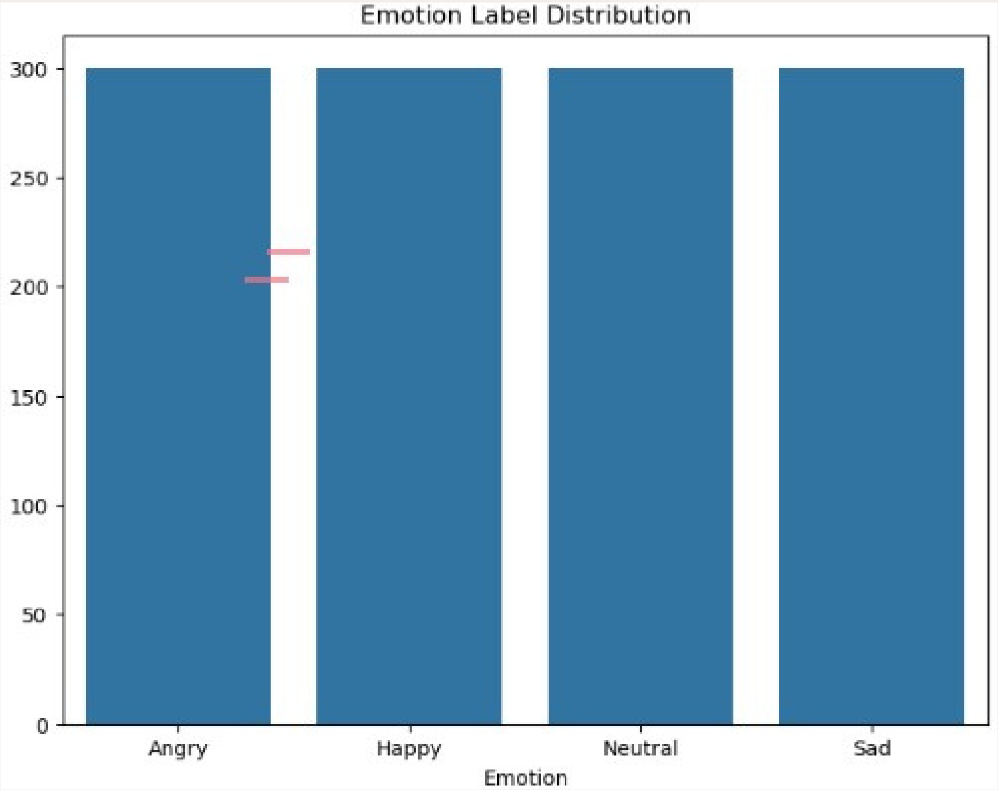


Training Samples:	960
Test Samples	120
Feature Size	30/sample
No. of Classes	4

TCN

```
Loaded features with shape: (1200, 30)
Loaded labels with shape: (1200,)
Label encoding:
Angry: 0
Happy: 1
Neutral: 2
Sad: 3
Feature mean (after scaling): [-8.93729535e-17  1.96787031e-16 -8.45619870e-17  1.94659104e-16
 -1.52348042e-15] (truncated)
Feature std (after scaling): [1. 1. 1. 1. 1.] (truncated)
Training set size: 960 samples
Validation set size: 120 samples
Testing set size: 120 samples

Processed data saved as .npy files.
```

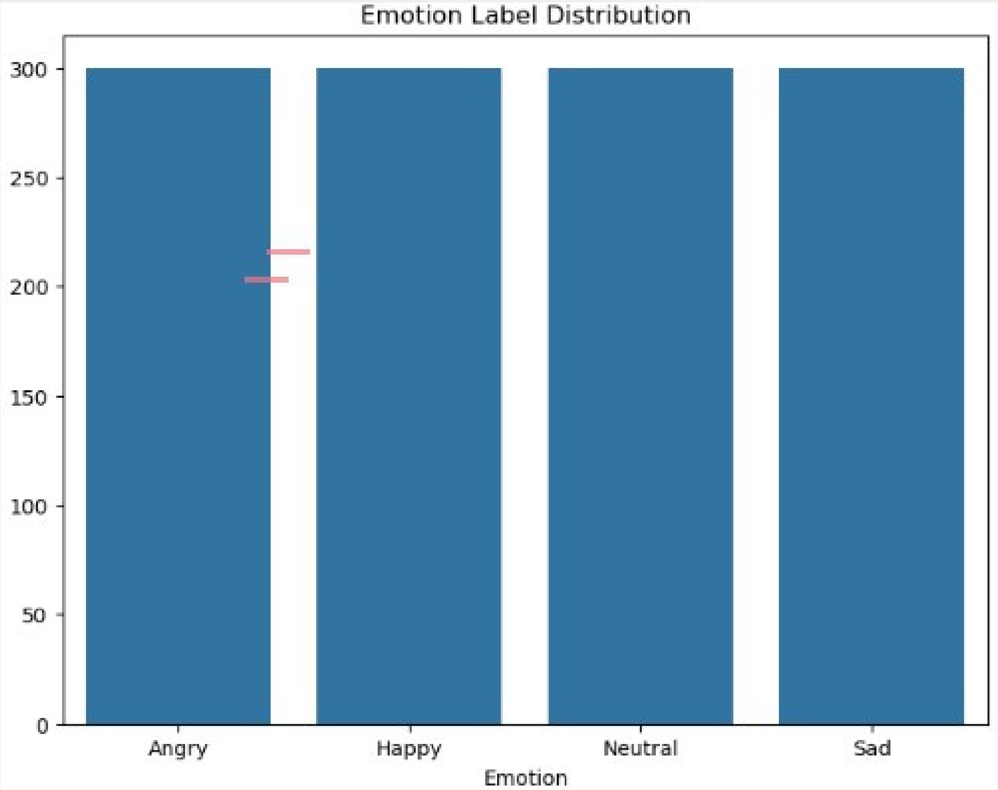


Training Samples:	960
Test Samples	120
Feature Size	30/sample
No. of Classes	4

LSTM

```
Loaded features with shape: (1200, 30)
Loaded labels with shape: (1200,)
Label encoding:
Angry: 0
Happy: 1
Neutral: 2
Sad: 3
Feature mean (after scaling): [-8.93729535e-17  1.96787031e-16 -8.45619870e-17  1.94659104e-16
 -1.52348042e-15] (truncated)
Feature std (after scaling): [1. 1. 1. 1. 1.] (truncated)
Training set size: 960 samples
Validation set size: 120 samples
Testing set size: 120 samples

Processed data saved as .npy files.
```

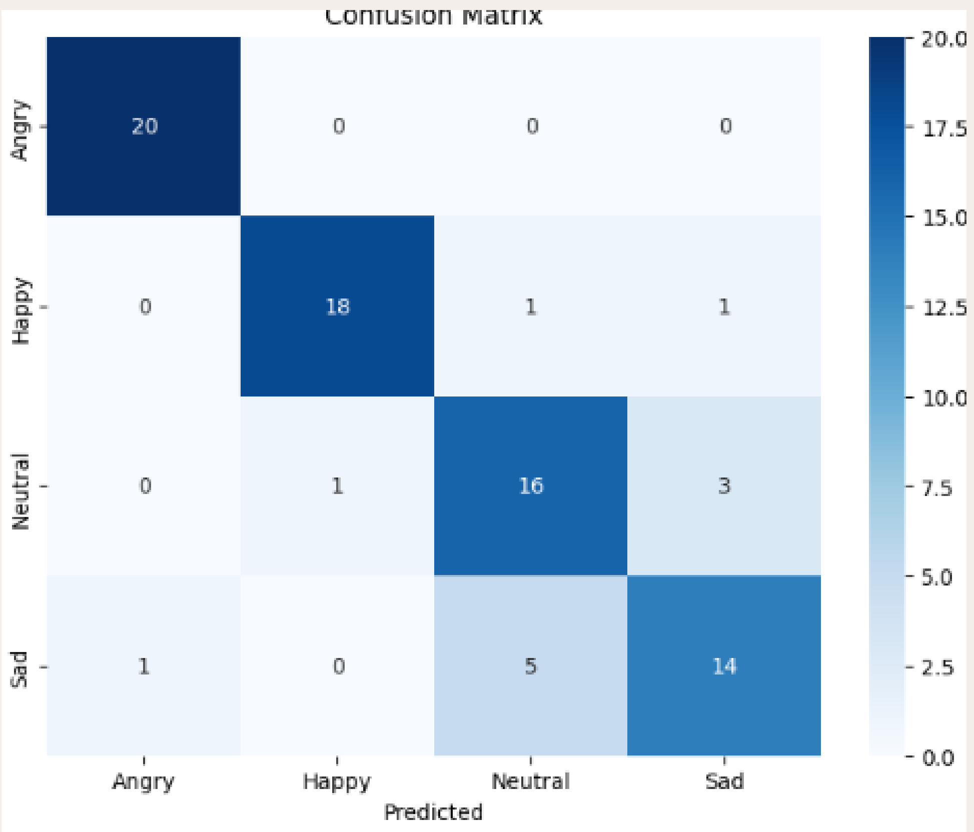


Training Samples:	960
Test Samples	120
Feature Size	30/sample
No. of Classes	4

CNNs

Classification Report:

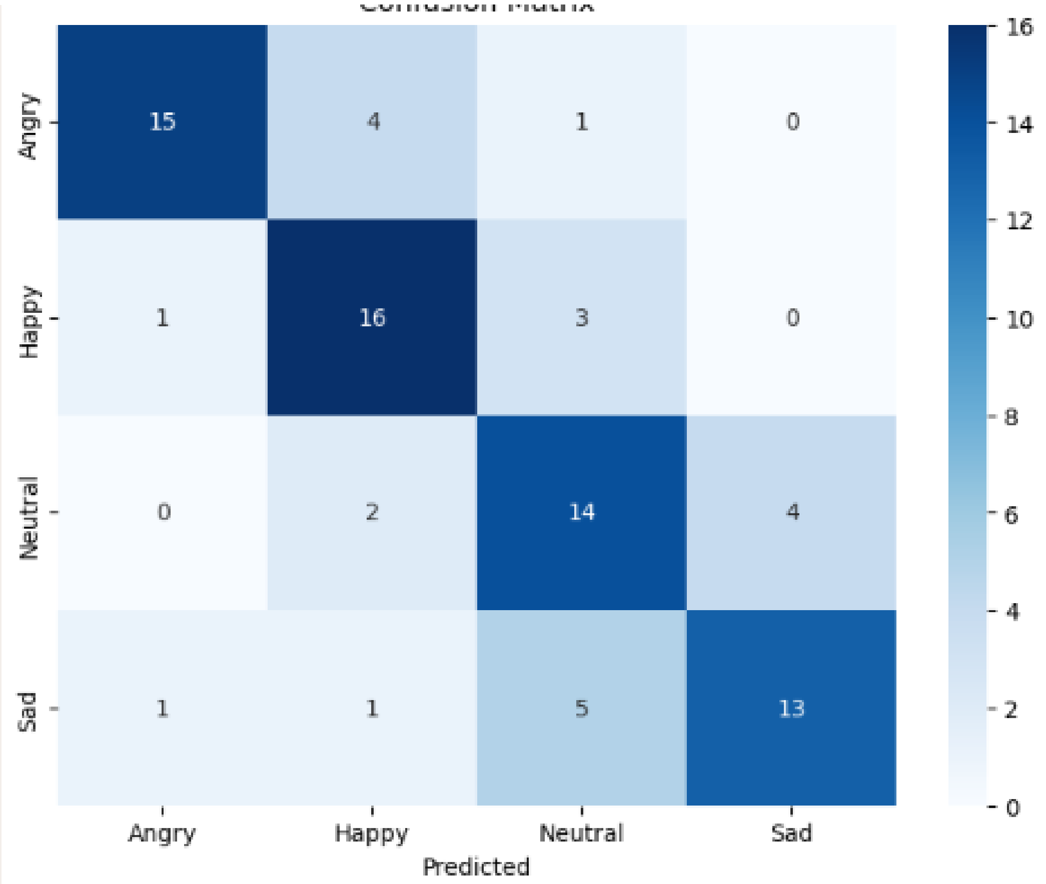
	precision	recall	f1-score	support
Angry	0.95	1.00	0.98	20
Happy	0.95	0.90	0.92	20
Neutral	0.73	0.80	0.76	20
Sad	0.78	0.70	0.74	20
accuracy			0.85	80
macro avg	0.85	0.85	0.85	80
weighted avg	0.85	0.85	0.85	80



DNNs

Classification Report:

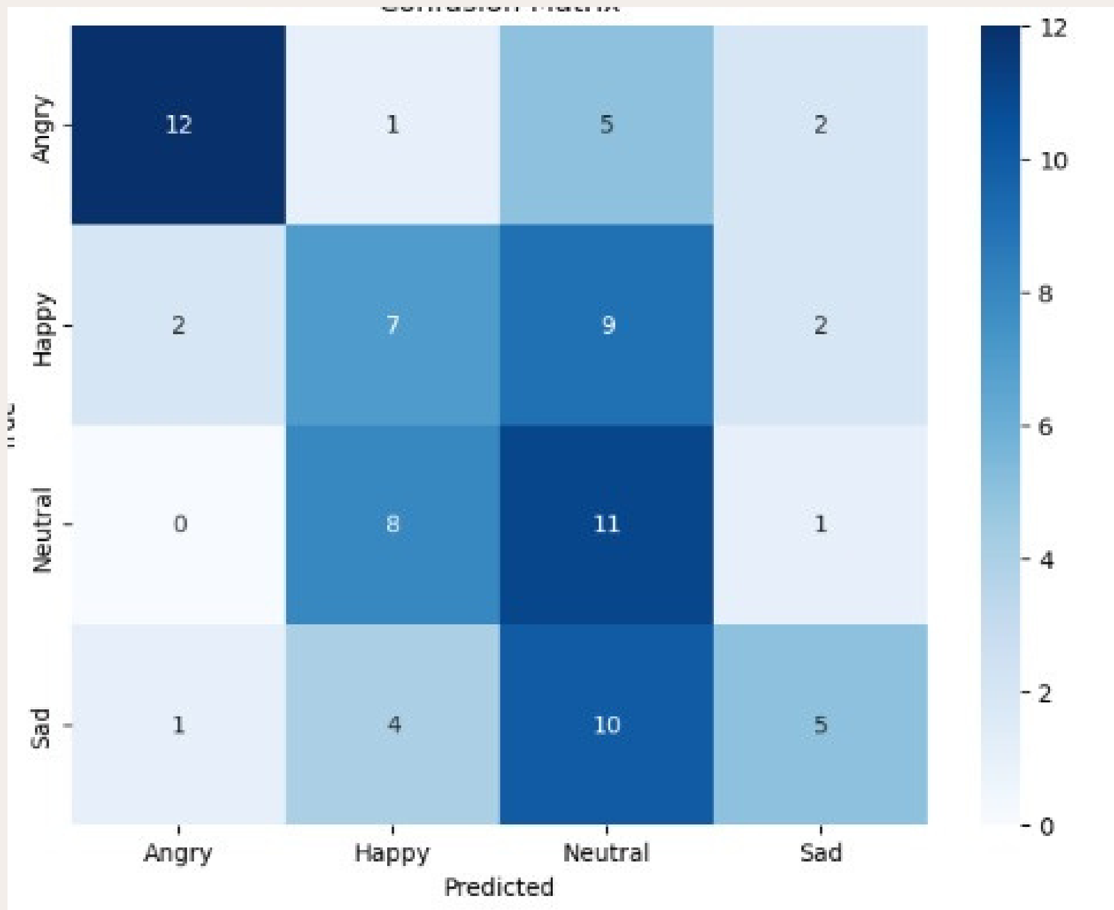
	precision	recall	f1-score	support
Angry	0.88	0.75	0.81	20
Happy	0.70	0.80	0.74	20
Neutral	0.61	0.70	0.65	20
Sad	0.76	0.65	0.70	20
accuracy			0.72	80
macro avg	0.74	0.72	0.73	80
weighted avg	0.74	0.72	0.73	80



LSTM

Classification Report:

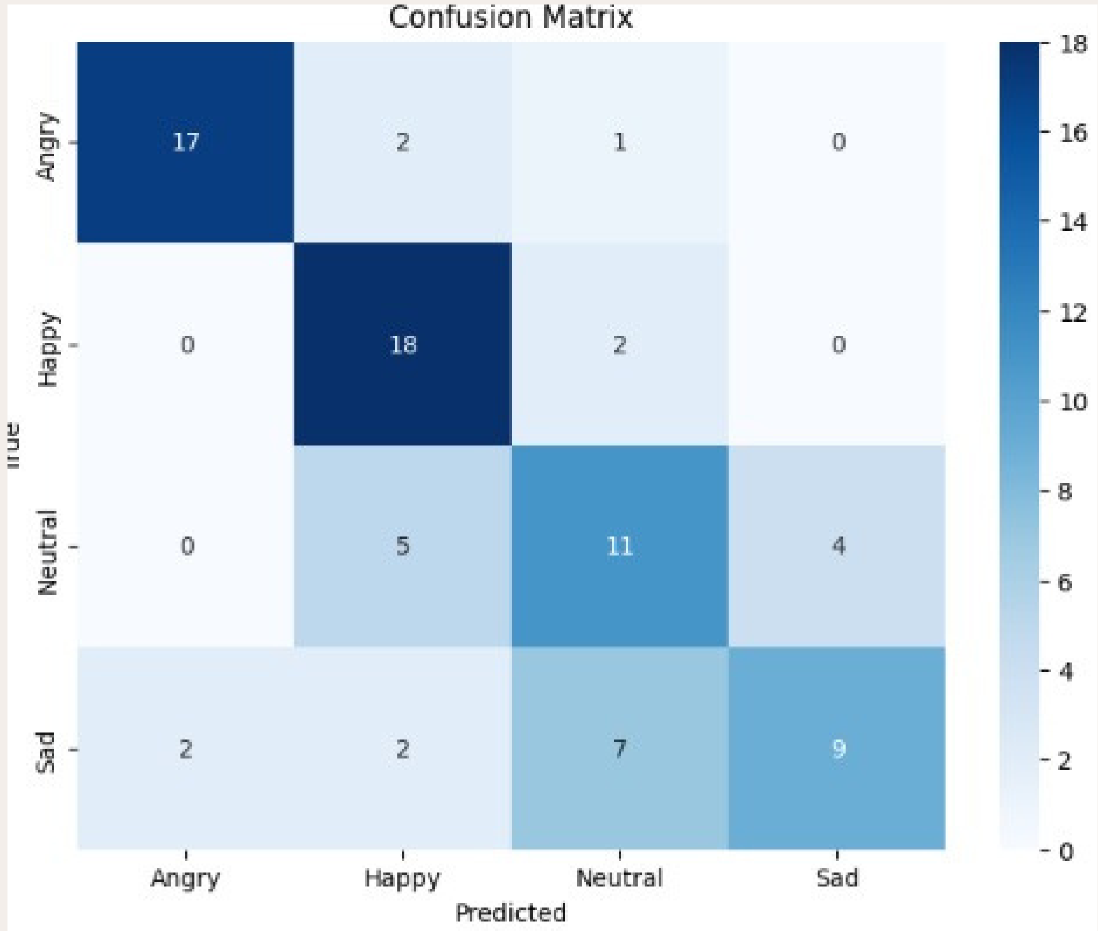
	precision	recall	f1-score	support
Angry	0.80	0.60	0.69	20
Happy	0.35	0.35	0.35	20
Neutral	0.31	0.55	0.40	20
Sad	0.50	0.25	0.33	20
accuracy			0.44	80
macro avg	0.49	0.44	0.44	80
weighted avg	0.49	0.44	0.44	80



RNN

Classification Report:

	precision	recall	f1-score	support
Angry	0.89	0.85	0.87	20
Happy	0.67	0.90	0.77	20
Neutral	0.52	0.55	0.54	20
Sad	0.69	0.45	0.55	20
accuracy			0.69	80
macro avg	0.69	0.69	0.68	80
weighted avg	0.69	0.69	0.68	80



CNNs	DNNs
Efficiently captures local patterns and hierarchical features, in audio signals, by leveraging convolutional layers	Relies solely on fully connected layers, which are less effective at capturing temporal dependencies present in audio features.
Works well with high-dimensional inputs, by reducing dimensionality through convolution and pooling layers	Struggles with high-dimensional inputs as it lacks a mechanism to focus on local feature region
Sindhi is a phonetic and tonal language where emotion can be expressed through pitch and energy changes	May fail to distinguish tonal variations effectively as it does not explicitly focus on localized patterns in the feature space

WHAT'S DIFFERENT

CNNs	Laghari	**DNNs**
Test Accuracy: 85%: weighted avg: 85% Chance Level: 25%	Test Accuracy: 66.50% weighted avg: 66.23% Chance Level: 16.67%	Test Accuracy: 72%: weighted avg: 73% Chance Level: 25%
01	02	03

CHALLENGES

Data Scarcity

A lack of publicly available datasets for Sindhi emotion recognition is a major barrier along with people's lack of willingness.

Manual Pre-proccessing

A lot of the audios needed manual cleaning and clipping. which was very time consuming. Hence, processing scrapped audios was not feasible.

Emotional Variability

Sindhi has around 12 dialects. Additionally, each individual expresses emotions with variable intensity.

Compatibility and Integration

The only substantial data-set for Sindhi provides only the extracted feature sets from the audios they acquired. However it was not compatible with our models.

FUTURE WORK

Plans	Explanation
Expansion of the Dataset:	The current dataset is limited, collected through WhatsApp from a relatively small set of speakers; with just 4 emotion classes.
Real-Time Emotion Recognition:	The current system likely operates offline, requiring the entire speech signal to be processed first. It could be improved to be deployed in local customer service etc.
Fine-Tuning and Transfer Learning	The model might not generalize well to unseen data or different dialects of Sindhi.Utilize transfer learning to fine-tune models trained on other languages for better generalization across dialects and accents.

THANK YOU