بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

Developing a Speech Recognition System for the Urdu Language

Members:

Muhammad Ibtesam Arshad FA17-BSE-061

Hamza Safdar FA17-BSE-013

Speech Recognition System for the Urdu Language

OUTLINE

- Data Collection and Model Training
- Build Flask API
- Deployment of Flask API
- Development of Mobile Application

Data Collection

DATA:

Center for Language Engineering Speech Corpus

Name:

Phonetically Rich Urdu Speech Corpus

Total Instances:

708 Speeches

Split Dataset

DATASET

The Dataset used for this Project comprises of

- **708 Instances**

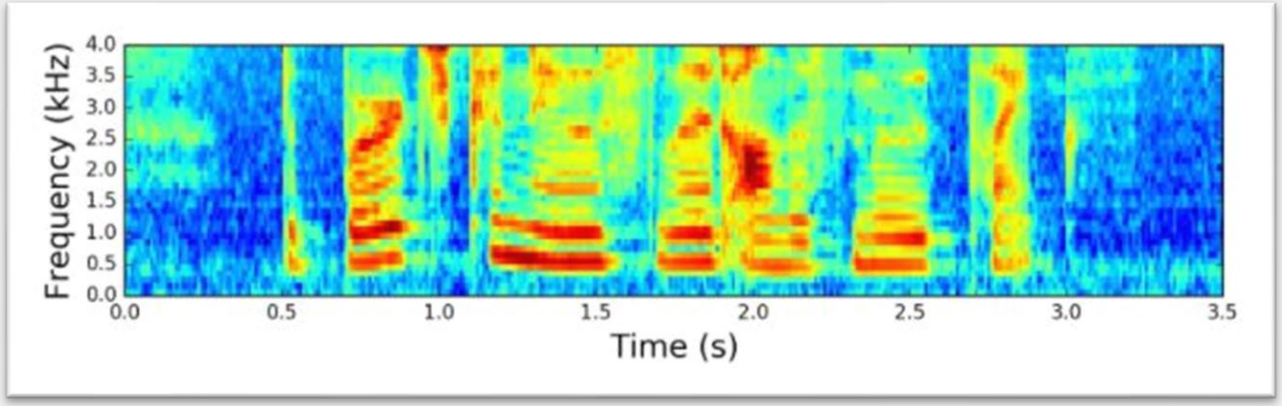
Sample Data Characteristics

- Total Instances in Sample Data = 708

  Training = 658 and **Testing = 50**

Data Preparation

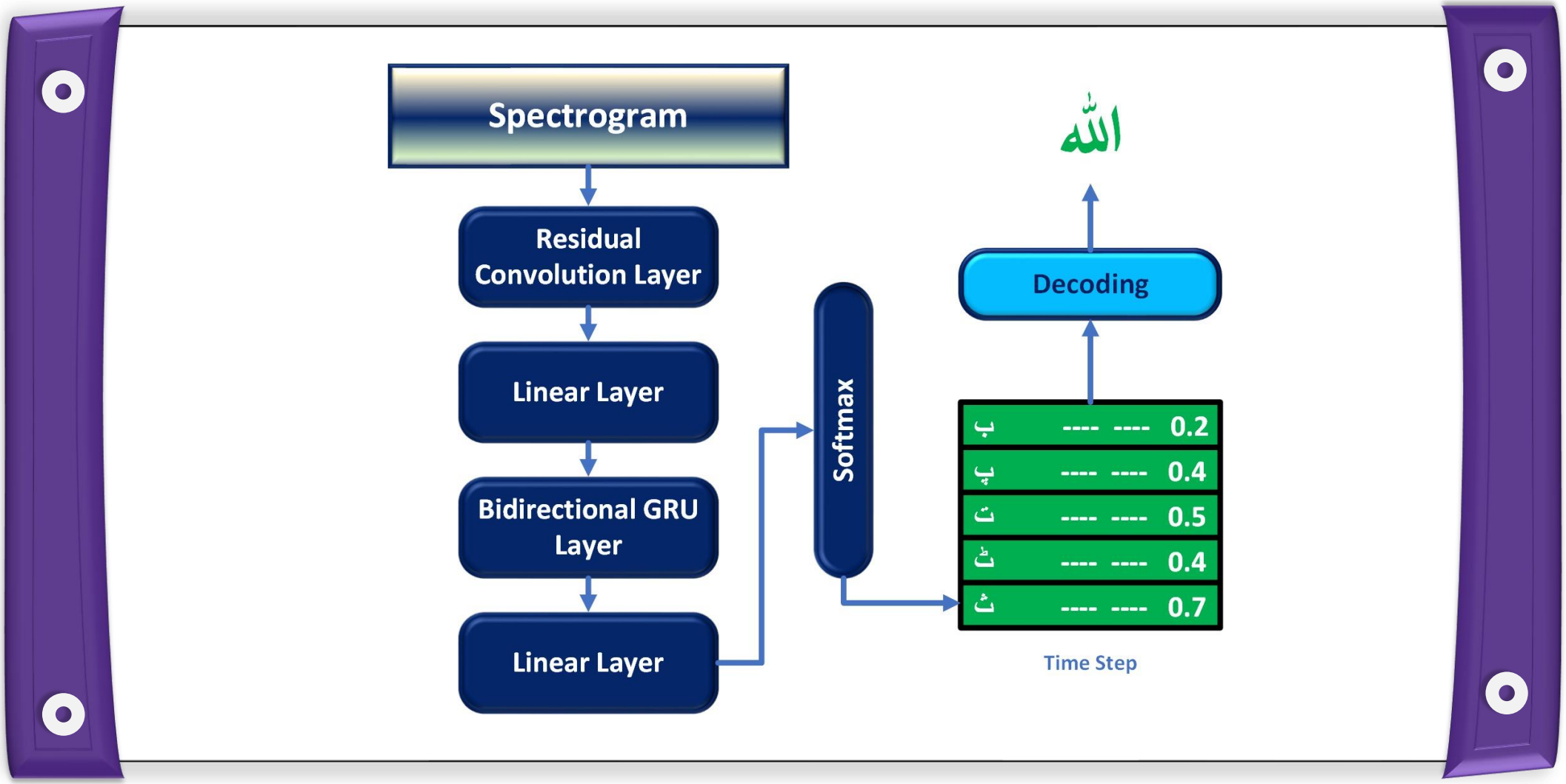
We'll take raw audio waves and transform them into Mel Spectrograms.



Model Architecture

Our model will be similar to the Deep Speech 2 architecture. The model will have two main neural network modules - N layers of Residual Convolutional Neural Networks (ResCNN) to learn the relevant audio features, and a set of Bidirectional Recurrent Neural Networks (BiRNN) to leverage the learned ResCNN audio features. The model is topped off with a fully connected layer used to classify characters per time step.

Model Architecture



Evaluating Speech Model

- We used a **Greedy Decoding** method to process our model's output into characters that can be combined to create the transcript.
- A **Greedy Decoder** takes in the model output, which is a softmax probability matrix of characters, and for each time step (spectrogram frame), it chooses the label with the highest probability. If the label is a blank label, we remove it from the final transcript.

- Then we calculated:
  - **Word Error** and **Character Error** using Levenshtein Distance.

Deploying Model through FLASK

FLASK Framework

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Routing

1

```
# Route to handle HOME
@app.route('/')
def home():
```

2

```
# Route to handle PREDICTED RESULT
@app.route('/predict')
def predict():
```

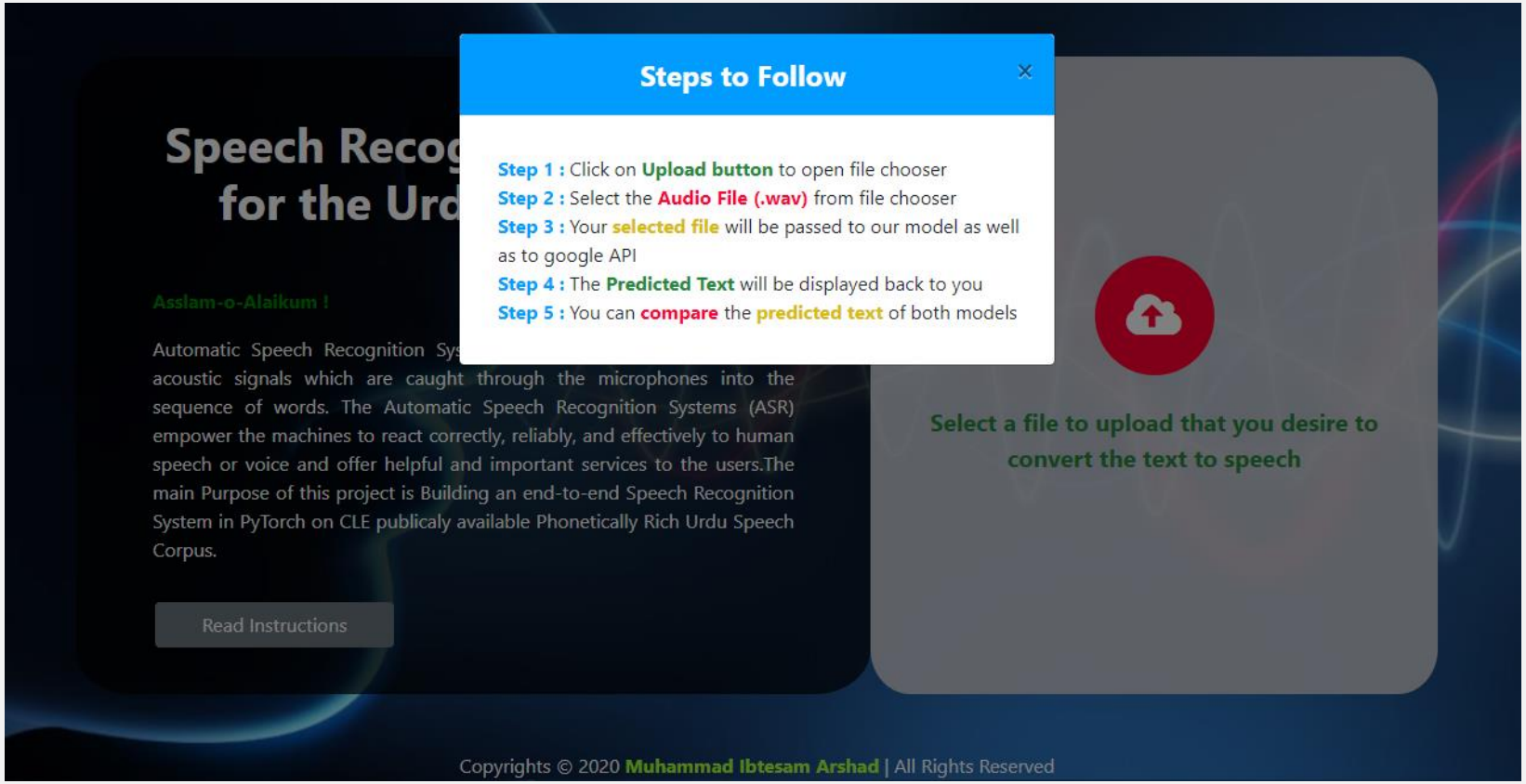
3

```
# Route to Handle API Request
@app.route('/predict-api')
def predict_api():
```

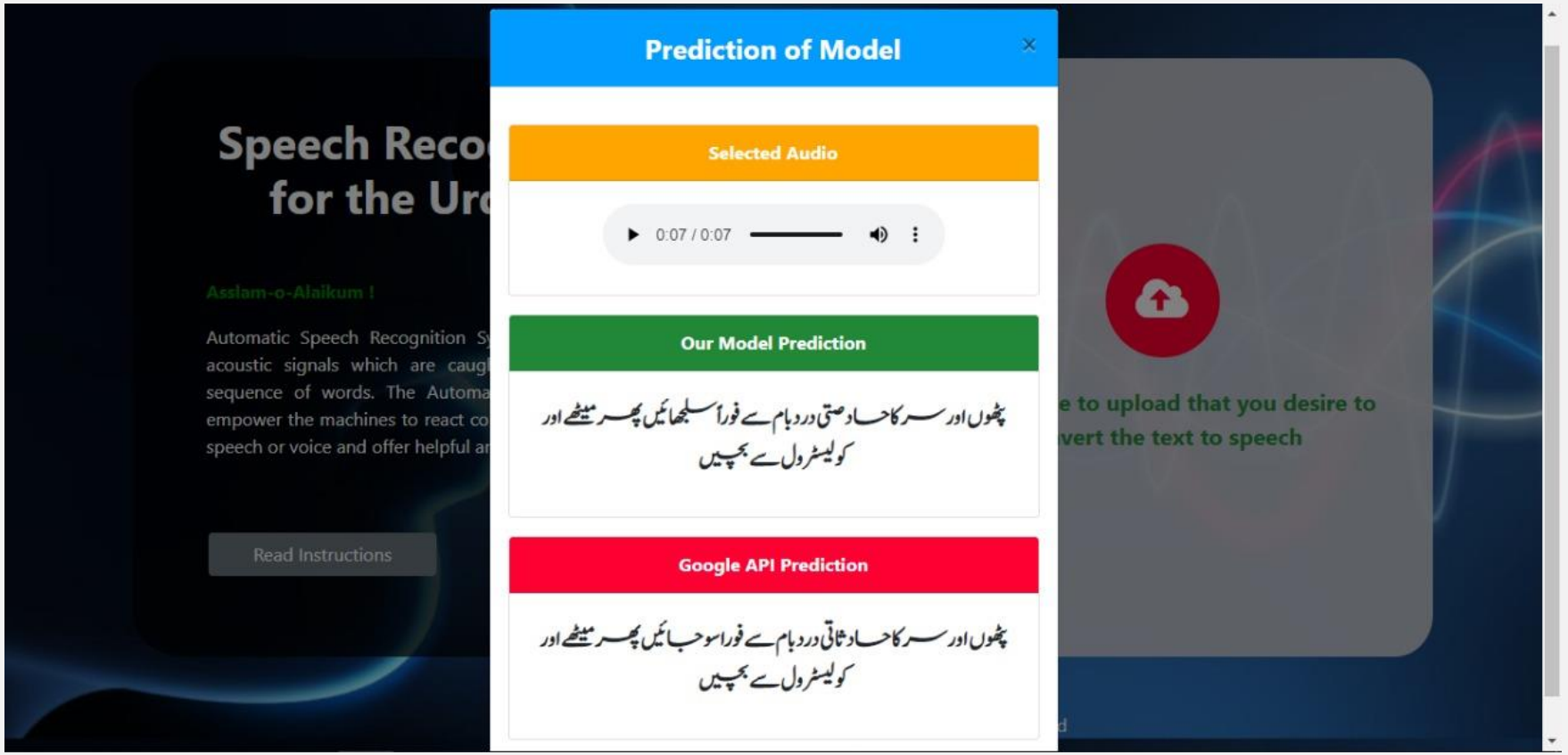
Interfaces
Web Application
and
Mobile Application

Landing Page – Web App

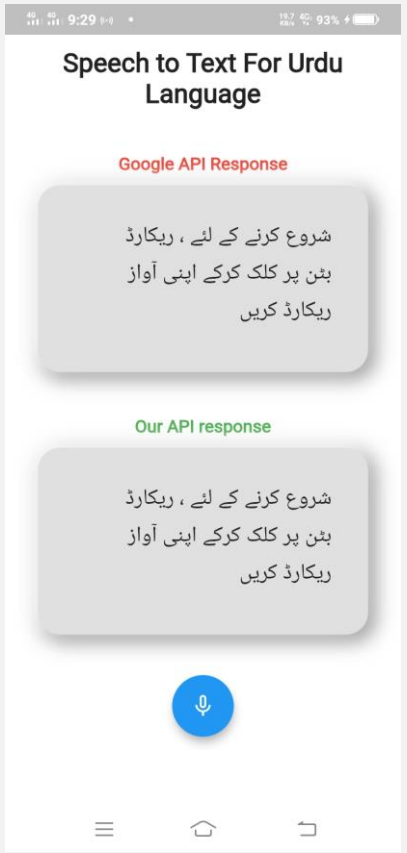
Instructions – Web App



Predictions – Web App



Predictions – Mobile App



DEPLOYMENT

We have Deployed our API on Cloud using Heroku

API Response

+ - View source

```
{
    Gender: "Male",
  - Predicted Text: {
        Our Prediction: "پٹھوں اور سرکا حات کدارد بام سے فوراً سلجھائیں پھر میٹھے اور کولیسٹرول سے بچیں",
        Roman: "Pathar aur sar ka Hadsa t Dard Bam se foreign suljhaen FIR Mithe aur cholesterol se
        bache",
        Urdu: "پٹھوں اور سر کا حادثاتی درد بام سے فورا سو جائیں پھر میٹھے اور کولیسٹرول سے بچیں"
    }
}
```

FUTURE WORK

Future Work

TASK 1

Speech Recognition System for the English Language

TASK 2

Speech Recognition System for the Urdu Language

جَزَاكَ ٱللَّهُ خَيْرًا