Automated Lip Reading Using Deep Learning Techniques in Python

Dr. L Mohana Sundari

Assistant Professor Senior Grade 1

Department Of Software Systems

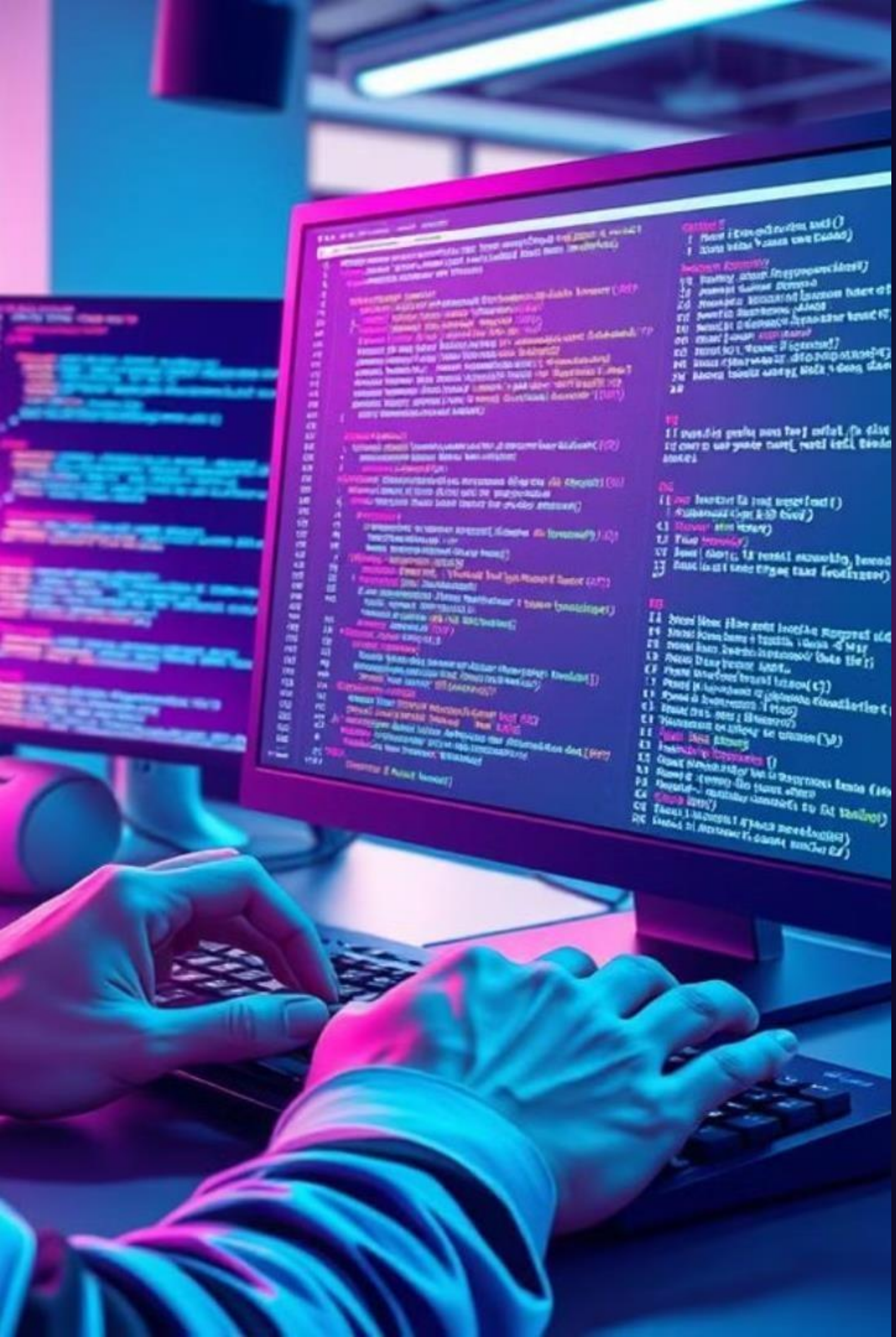
SCOPE

Batch Members :

21BCE3077 : Piyush Kumar

21BCE3239 : Madhav

21BCE3244 : Anas Khan



Objectives

Utilize advanced deep learning techniques such as Conv3D for extracting spatiotemporal features, MaxPooling for downsampling, and LSTM for modeling sequential data, ensuring high precision in word detection.

Improve accessibility for individuals with hearing impairments by offering an alternative communication tool that relies on visual cues from lip movements.

Enhance performance in noisy environments where traditional audio-based speech recognition systems struggle, making lip reading a robust alternative.

Develop a web-based application using the Streamlit framework to provide a user-friendly interface, allowing real-time demonstration and interaction with the lip reading model.

preencoded.png

Motivation

Lip reading, also known as speechreading, is the ability to understand speech by observing the movements of a speaker's lips. It's a valuable tool for individuals with hearing impairments and can be beneficial in noisy environments.

1 Growing Demand

The demand for lip reading technologies is increasing due to the rise of hearing loss and the need for more accessible communication methods.

2 Advances in AI

Recent advances in deep learning have opened new possibilities for automated lip reading systems with improved accuracy and performance.

3 Applications

Applications of lip reading systems range from assistive technologies for the hearing impaired to security systems and human-computer interaction.

Literature Survey

A comprehensive review of existing lip reading systems was conducted, covering various techniques and architectures.

Study/Model	Summary	Limitations
LipNet (Assael et al., 2016)	Developed an end-to-end deep learning model combining CNN and RNN for sentence- level lip reading	Struggles with unseen speakers and large variations in lip movements.
Shillingford et al. (2018)	Proposed a large-scale dataset and used 3D CNNs to capture spatiotemporal features, improving lip reading in noisy conditions.	Requires large datasets, computationally expensive
Martinez et al. (2020) - Transformers	Applied transformer models to lip reading, demonstrating improvements in modeling long-term dependencies	Transformers require large datasets and high computational power.

preencoded.png

Gap Identification

The existing lip reading systems often struggle with low accuracy in real-world scenarios, especially with noisy backgrounds and variations in speaker's lip movements.

Challenges

Limited Models Focused on Audio Inputs:

- Most existing research and systems focus primarily on audio-based speech recognition.
- Few models investigate or implement visual-based lip reading techniques.

Need for Real-Time, Efficient Models in Noisy Environments:

- Current models often fail to perform reliably in noisy or dynamic environments.
- Real-time processing and adaptability to various conditions are lacking.

Opportunities

Real-Time and Robust Model Implementation:

- Create solutions that can handle noisy backgrounds and offer real-time performance.
- Design efficient models with improved accuracy and adaptability for practical applications.

Integration of Visual-Based Models:

- Develop models that leverage lip reading and visual cues for speech prediction.
- Explore and advance visual-based deep learning techniques to address this gap.

preencoded.png

Proposed Methodology

A novel deep learning approach is proposed, combining convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to effectively extract features and model temporal dependencies in lip movements.

1

Data Acquisition

A comprehensive dataset of lip movements will be collected, covering various speakers and environments.

2

Preprocessing

The collected data will be preprocessed to normalize the lip movements and enhance the quality of the data.

3

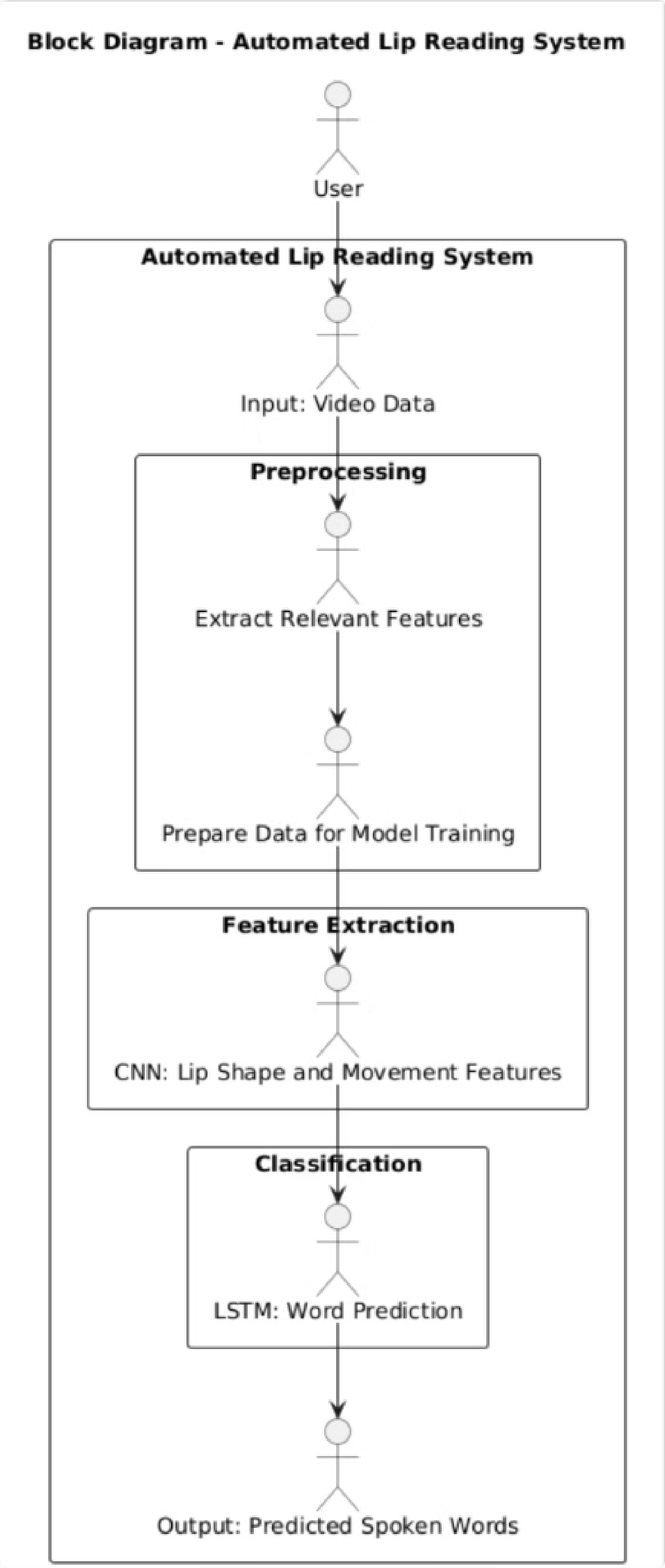
Model Training

A CNN-RNN model will be trained on the preprocessed data to learn patterns and relationships in lip movements.

4

Evaluation

The trained model will be evaluated on unseen data to assess its performance and accuracy.



Block Diagram

1

Data Acquisition

The system begins with data acquisition, collecting video sequences of speakers' lip movements.

2

Preprocessing

The acquired data is preprocessed to normalize the lip movements and remove any noise or artifacts.

3

Feature Extraction

A convolutional neural network (CNN) extracts features from the preprocessed lip images, capturing spatial information.

4

Temporal Modeling

A recurrent neural network (RNN) models the temporal dependencies in the extracted features, capturing the dynamic aspects of lip movements.

5

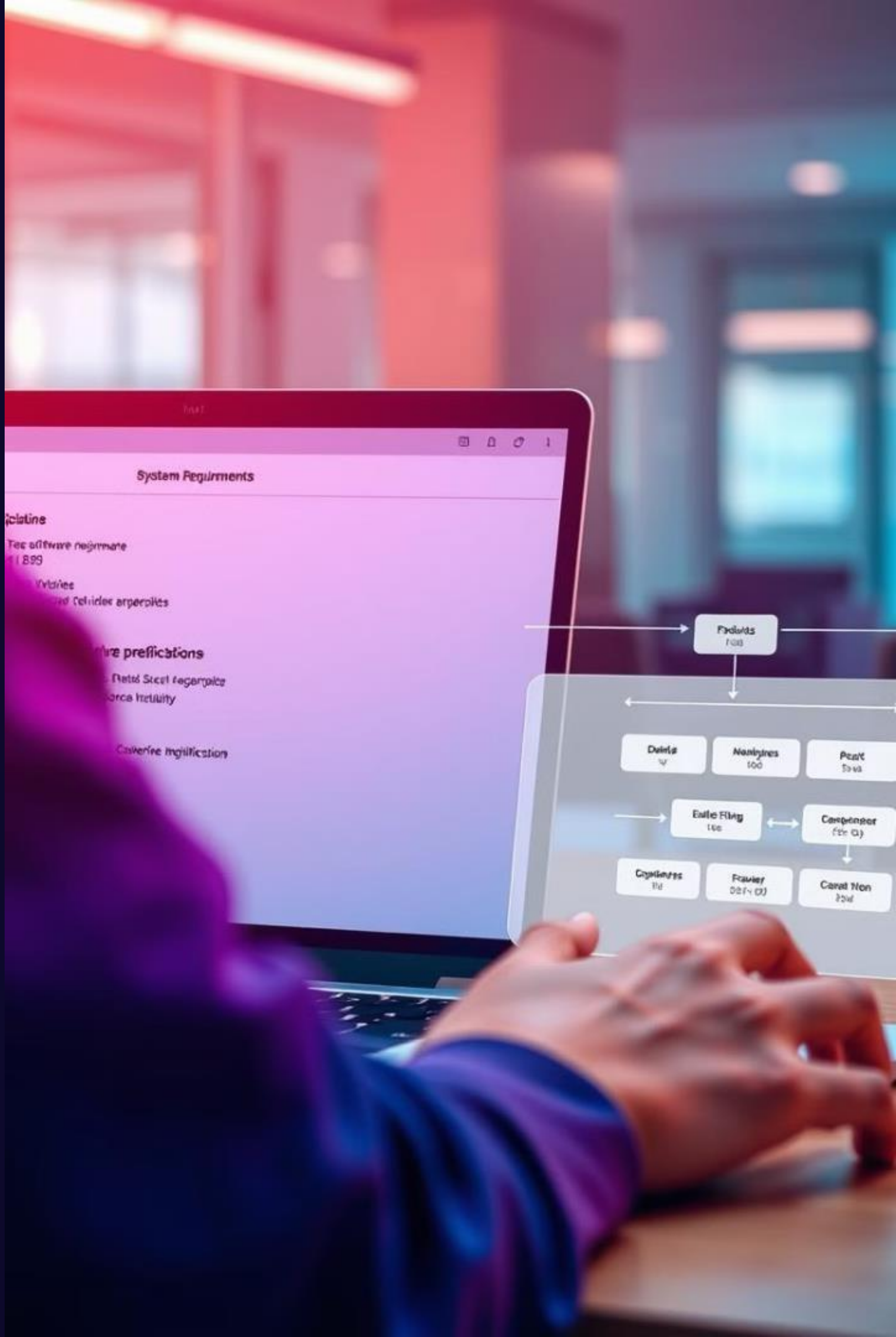
Output

The system outputs the recognized words or phrases based on the learned features and temporal patterns.

Problem Statement

The challenge addressed by this project is the development of an effective automated lip-reading system capable of accurately interpreting spoken words from video footage in real-world conditions. Existing systems often struggle with variations in lighting, facial occlusions, and speaker diversity, leading to reduced accuracy and reliability. Moreover, many current solutions are limited by their computational demands, hindering their applicability in real-time scenarios such as security and surveillance.

Additionally, there is a significant gap in integrating lip reading technologies with assistive tools for individuals with hearing impairments, as well as addressing ethical concerns related to privacy and consent in surveillance applications. This project aims to overcome these limitations by creating a deep learning-based lip-reading system that performs well under varied conditions, processes information in real-time, and incorporates ethical considerations, thereby providing a more robust and practical solution for both accessibility and security purposes.



Requirement Analysis

The system requires a high-performance computer with sufficient processing power to handle the computationally intensive deep learning tasks.

Hardware Requirements

The system requires a powerful CPU, GPU, and sufficient RAM to handle the training and inference processes.

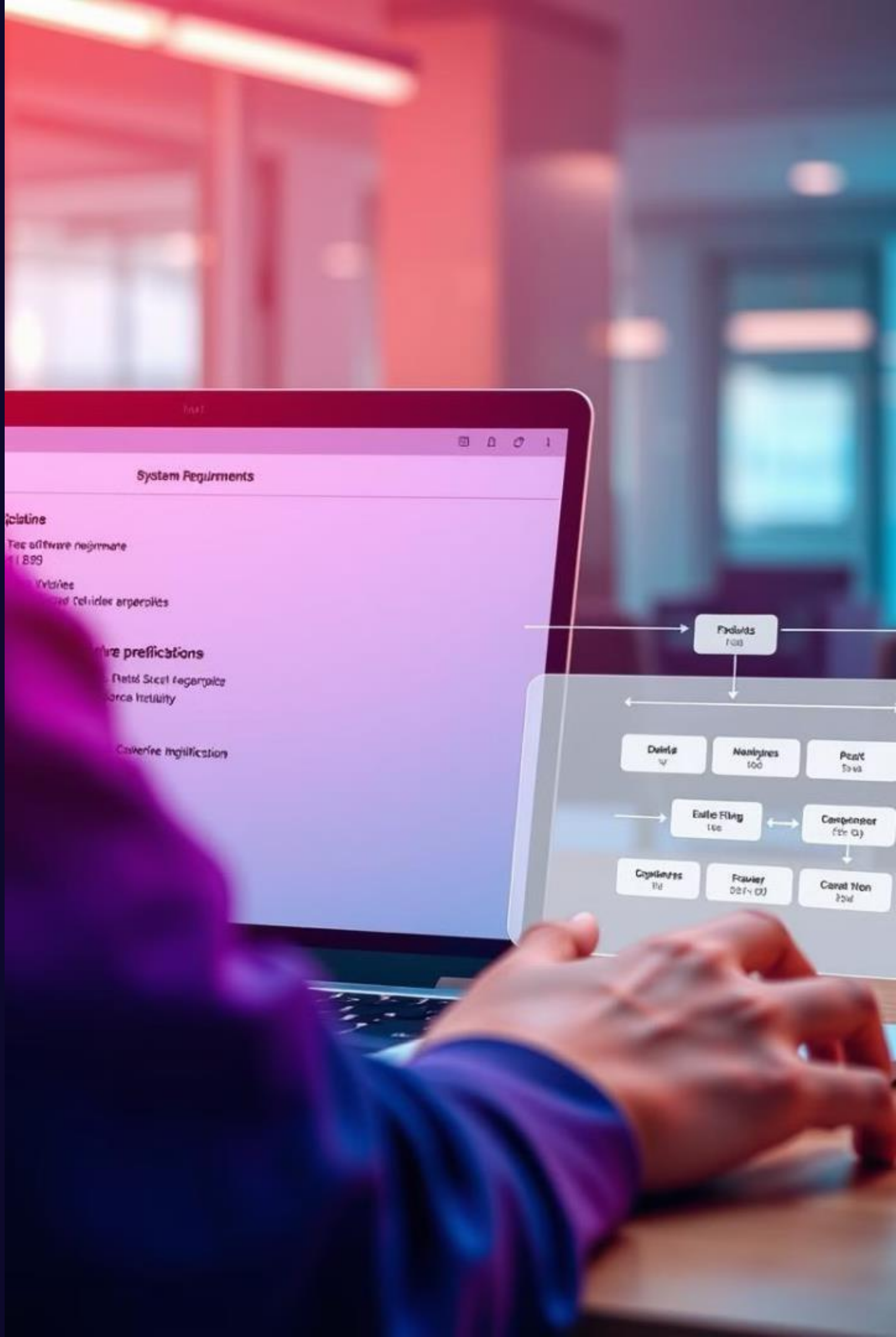
Software Requirements

The system requires a suitable software environment with libraries like TensorFlow or PyTorch for deep learning.

For developing and training deep learning models (Conv3D, LSTM).

Data Requirements

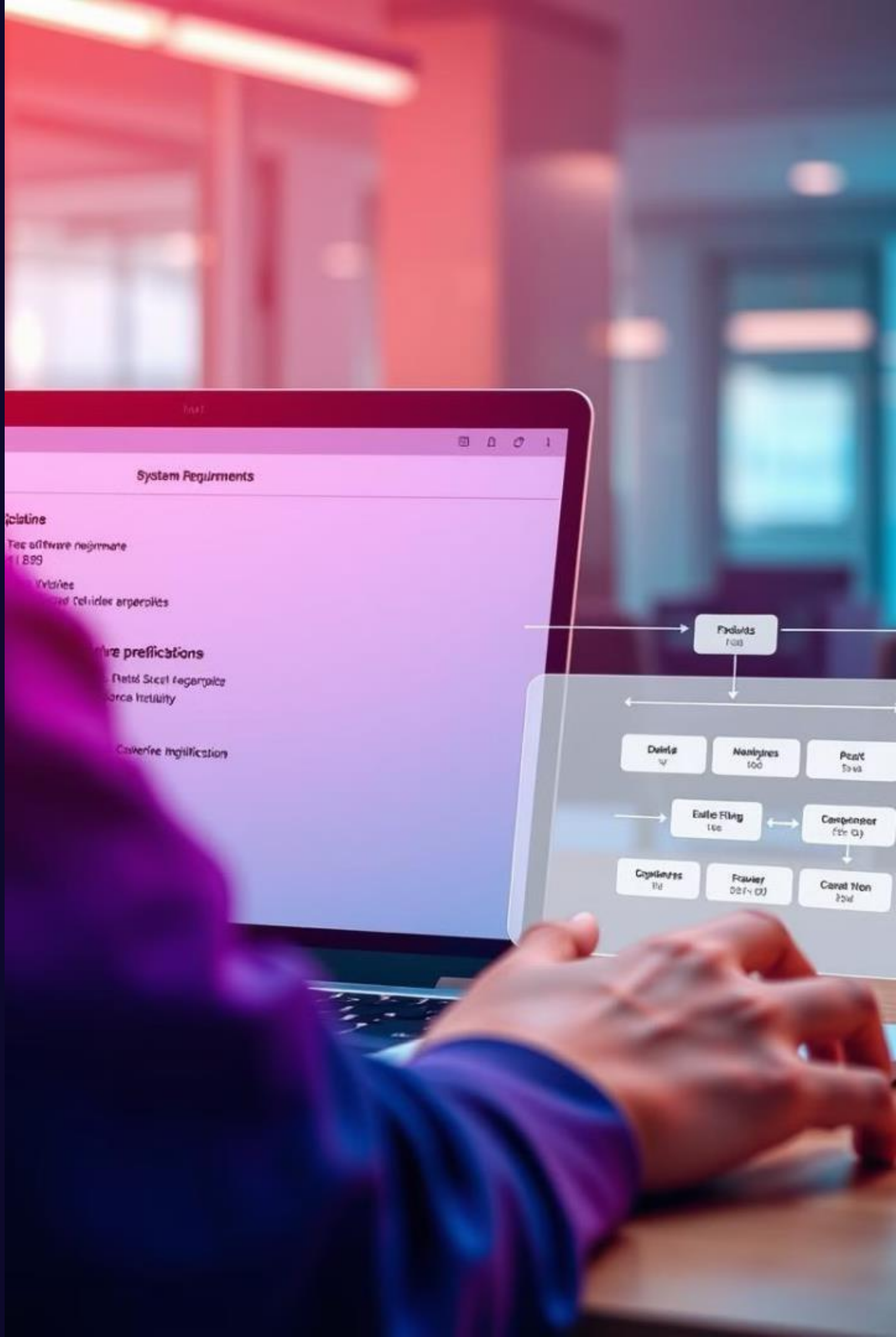
A large and diverse dataset of lip movements is essential for training the deep learning model.



Proposed Output

The system requires a high-performance computer with sufficient processing power to handle the computationally intensive deep learning tasks.

Input: Drive Link of a Video



Copy of LipNet.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

..
data
alignments
s1
models
sample_data
__temp__.mp4
checkpoints.zip
converted_video.mp4
data.zip

Disk 77.46 GB available

+ Code + Text

```
[81] sample = load_data(tf.convert_to_tensor('./data/s1/bras9a.mpg'))
```

```
[82] from typing_extensions import Text
     print('REAL TEXT')
     text=[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in [sample[1]]]
     print(text[0].numpy().decode('utf-8'))
```

```
REAL TEXT
bin red at s nine again
```

```
[83] yhat = model.predict(tf.expand_dims(sample[0], axis=0))
```

```
1/1 [==============================] - 5s 5s/step
```

```
[84] decoded = tf.keras.backend.ctc_decode(yhat, input_length=[75], greedy=True)[0][0].numpy()
```

```
print('PREDICTIONS')
text = [tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in decoded]
print(text[0].numpy().decode('utf-8'))
```

```
PREDICTIONS
bin red at s nine again
```

hence we can see real text and predictions are matching !!

0s completed at 16:03

Lip_Reading.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

Files

bgwu7s.mpg
bgwu8p.mpg
bgwu9a.mpg
braf8n.mpg
braf9s.mpg
brag1a.mpg
bragzp.mpg
bram2n.mpg
bram3s.mpg
bram4p.mpg
bram5a.mpg
bras6n.mpg
bras7s.mpg
bras8p.mpg
bras9a.mpg
brba1a.mpg
brbazp.mpg
brbg2n.mpg
brbg3s.mpg
brbg4p.mpg
brbg5a.mpg

Disk 77.46 GB available

+ Code + Text

T4 RAM Disk Gemini

```
50
```

```
[28] frames, alignments = data.as_numpy_iterator().next()
```

```
len(frames)
```

```
2
```

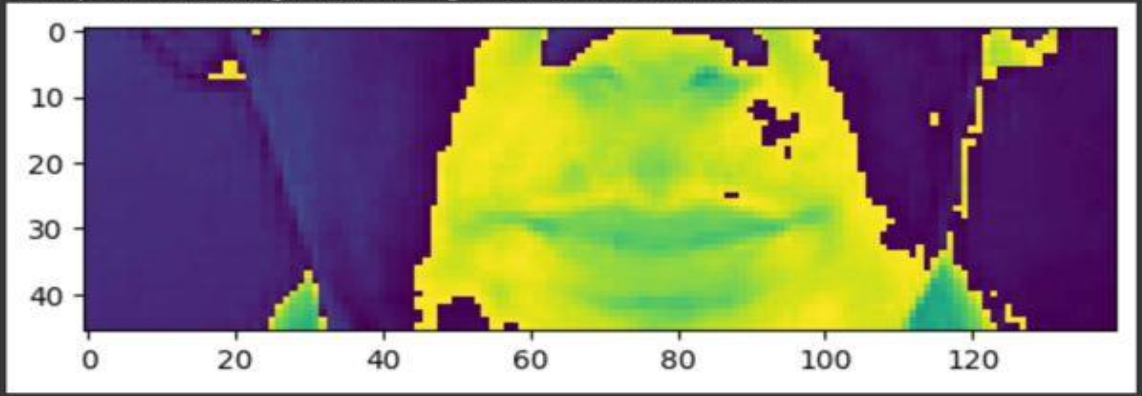
```
[30] sample = data.as_numpy_iterator()
```

```
[31] val = sample.next(); val[0]
```

Show hidden output

```
[32] # 0:videos, 0: 1st video out of the batch,  0: return the first frame in the video
plt.imshow(val[0][0][35])
```

<matplotlib.image.AxesImage at 0x78373a357e50>




```
[33] tf.strings.reduce_join([num_to_char(word) for word in val[1][0]])
```

```
<tf.Tensor: shape=(), dtype=string, numpy=b'place blue with c nine soon'>
```

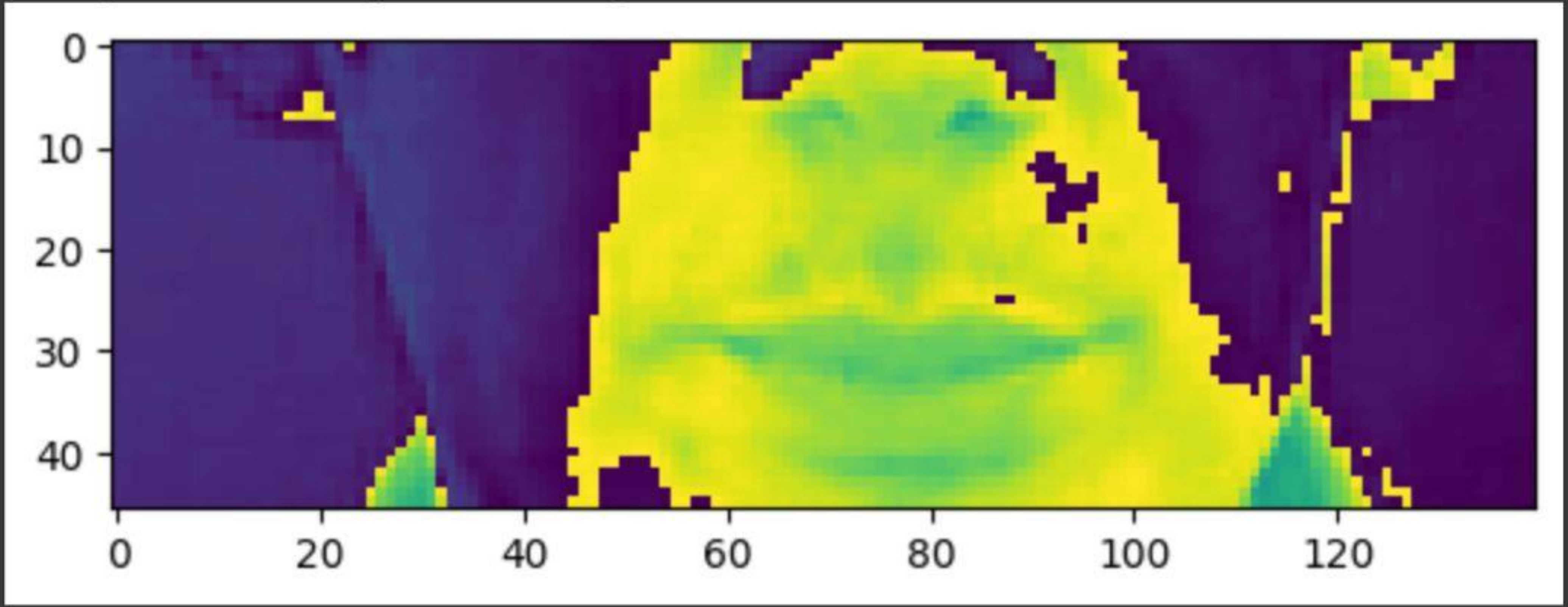
Design the Deep Neural Network

0s completed at 16:03

0s [32]

```python
# 0:videos, 0: 1st video out of the batch,  0: return the first frame in the video
plt.imshow(val[0][0][35])
```

```
<matplotlib.image.AxesImage at 0x78373a357e50>
```



Copy of LipNet.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

Files

- ..
- data
  - alignments
  - s1
- models
- sample_data
- __temp__.mp4
- checkpoints.zip
- converted_video.mp4
- data.zip

Disk 77.46 GB available

+ Code + Text

T4 RAM Disk

Gemini

```
# Display the video inline
clip.ipython_display(width=640, height=480)
```

```
Moviepy - Building video __temp__.mp4.
MoviePy - Writing audio in __temp__TEMP_MPY_wvf_snd.mp3
MoviePy - Done.
Moviepy - Writing video __temp__.mp4

                                                    Moviepy - Done !
Moviepy - video ready __temp__.mp4
```


```
[81] sample = load_data(tf.convert_to_tensor('./data/s1/bras9a.mpg'))
```

0s completed at 16:03

Conclusion

This project proposes a novel approach to automated lip reading using deep learning techniques, aiming to address the challenges of existing systems and improve the accuracy of lip recognition.

1 Future Work

Model Improvements: Explore more advanced models and techniques for better accuracy and efficiency.

Broader Applications: Expand the system for different languages, accents, and applications beyond assistive technology.

2 Impact

Enhanced Accessibility: Provides a crucial tool for individuals with hearing impairments, facilitating better communication through lip reading technology.

Advancement in AI: Demonstrates the application of advanced deep learning techniques (Conv3D, LSTM) for solving complex problems in visual speech recognition.