भारतीय सूचना प्रौद्योगिकी संस्थान सूरत
Indian Institute of Information Technology Surat
ભારતીય સૂચના પ્રૌદ્યોગિકી સંસ્થા સુરત
(An Institute of National Importance under Act of Parliament)

Mini Project (CS604) Presentation

Automated Lip Reading Using CNN and LSTM

Faculty Supervisor:
Dr. Ritesh Kumar

Presented by:
Aman Khan (UI22CS05)
Prashasti Vyas (UI22CS61)
Sunoy Roy (UI22CS77)

Outline

1. Introduction
2. Literature Review
3. Proposed Approach
4. Tools & Technologies Used
5. Dataset Description & Preprocessing
6. Configuration Used
7. Model Architectures & Analysis
    - CNN + GRU
    - CNN + BiLSTM Model
    - CNN + TCN
8. Results & Evaluation
9. Conclusion
10. References

Objective:

To develop an **end-to-end deep learning model** capable of accurately interpreting spoken words by analyzing lip movements **without any audio input**. The system leverages advanced neural network architectures like CNN+Bi-LSTM, CNN+GRU, 3D CNN+TCN to extract and model both spatial and temporal features of lip movement sequences.

Motivation:

- **Effective Communication in Noisy Environments**
- **Assistive Technology for the Hearing-Impaired**
- **Silent Communication**
- **Surveillance and Security Applications**

Deep Learning in Lip Reading

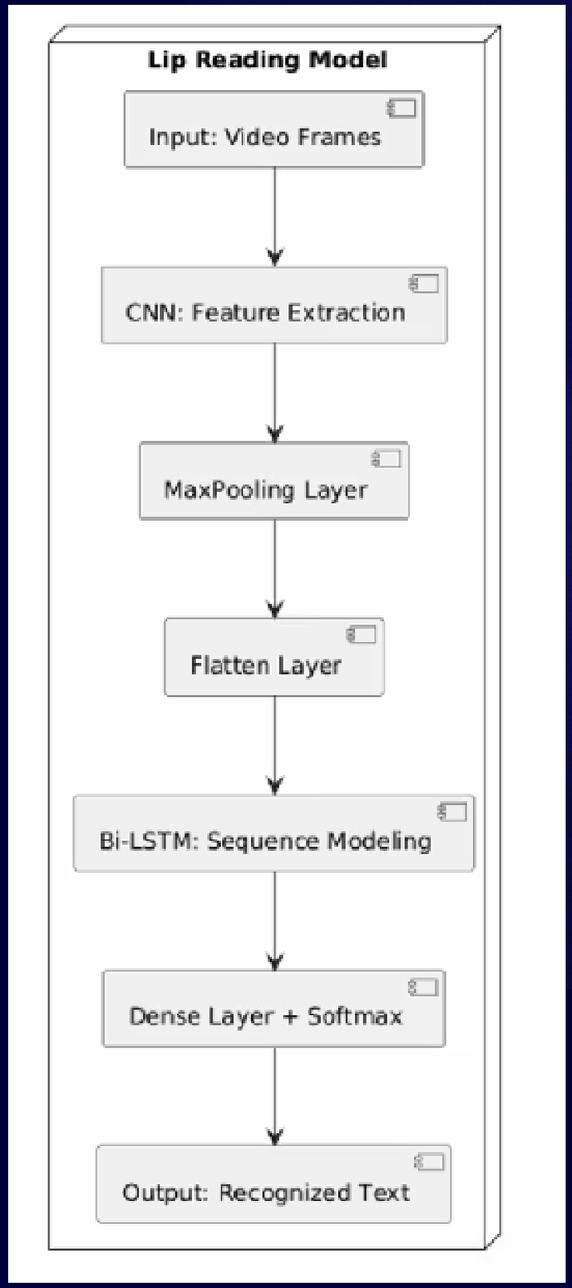
- **LipNet (Assael et al., 2016):** LipNet: CNN + LSTM model for sentence-level recognition.
- **Watch, Attend and Spell (WAS):** Watch, Attend and Spell (WAS): Sequence-to-sequence architecture with attention.

Challenges Identified in Lip Reading Research

- **Lip Shape Ambiguity**
- **Lack of Large Annotated Datasets**
- **Temporal dynamics modeling elaborate a bit**

Pipeline Overview:

- Video Input
- Frame Extraction
- Preprocessing & Face Cropping
- Feature Extraction using CNN
- Sequence Modeling using RNNs (LSTM, BiLSTM)
- Attention Mechanism (optional)
- Final Prediction using Softmax



PROPOSED APPROACH

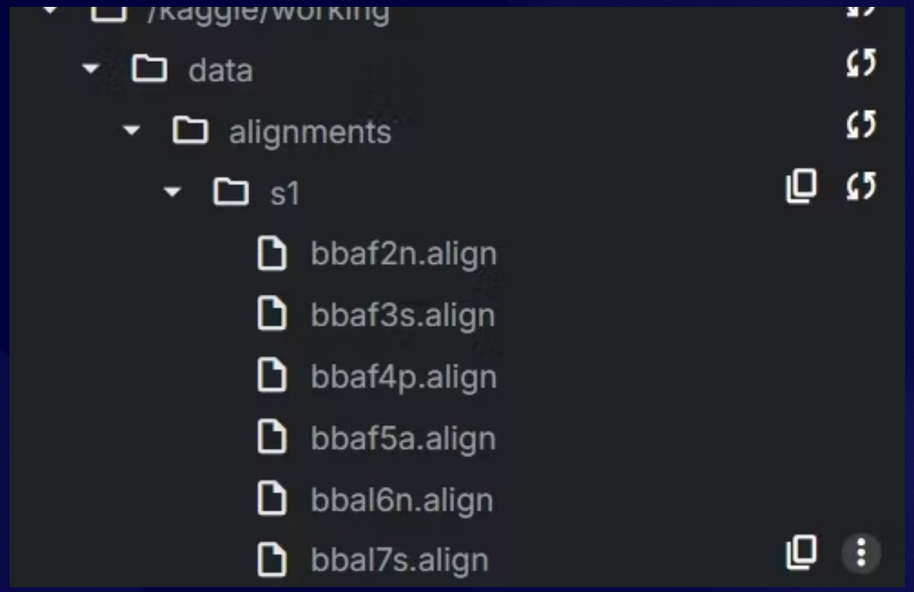
This project builds upon LipNet's foundation and explores more efficient and scalable implementations for real-time sentence-level lipreading using video inputs:

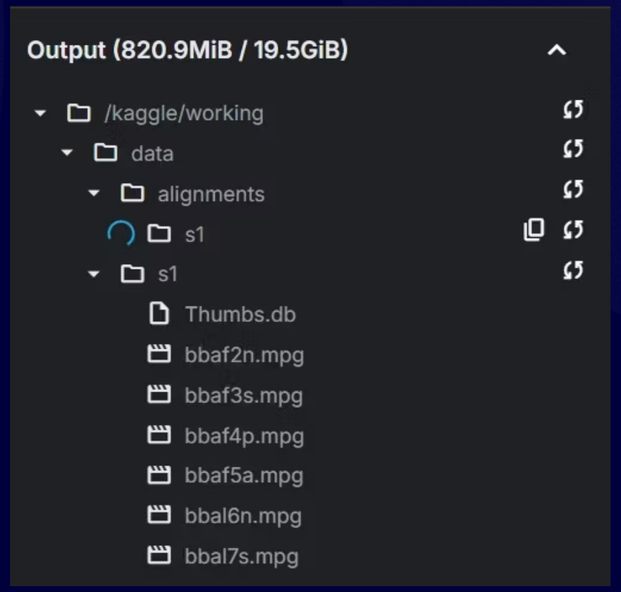
- GRID Corpus: A standardized audiovisual dataset used for training and evaluation, providing thousands of fixed-structure spoken sentences.
- Spatiotemporal CNNs: Learn low-level motion and lip features from video frames using 3D convolutional layers.
- Bidirectional GRUs: Capture forward and backward context, improving sentencelevel understanding of lip movements.
- CTC Loss: Eliminates the need for precise temporal alignment between input frames and output characters.
- TensorFlow / PyTorch: For model implementation, training, and optimization using GPU acceleration.
- Data Augmentation: Including random frame drop, horizontal flip, and normalization to improve generalization and robustness.
- Greedy vs. Beam Search Decoding: For generating the final text output from character probabilities.

TECHNOLOGIES USED

Dataset Overview

- **Source:** GRID Corpus
- **Data Format:**
  - Video files in .mpg format
  - Corresponding alignment files in .align format
- **Directory Structure:**
  - /data/s1/ contains video samples like bbaf2n.mpg
  - /data/alignments/s1/ contains transcription files like bbaf2n.align
- **Content Description:**
  - Each video shows a speaker uttering a sentence.
  - Each .align file provides **character-level time-aligned transcriptions** of the spoken content.





DATASET DESCRIPTION

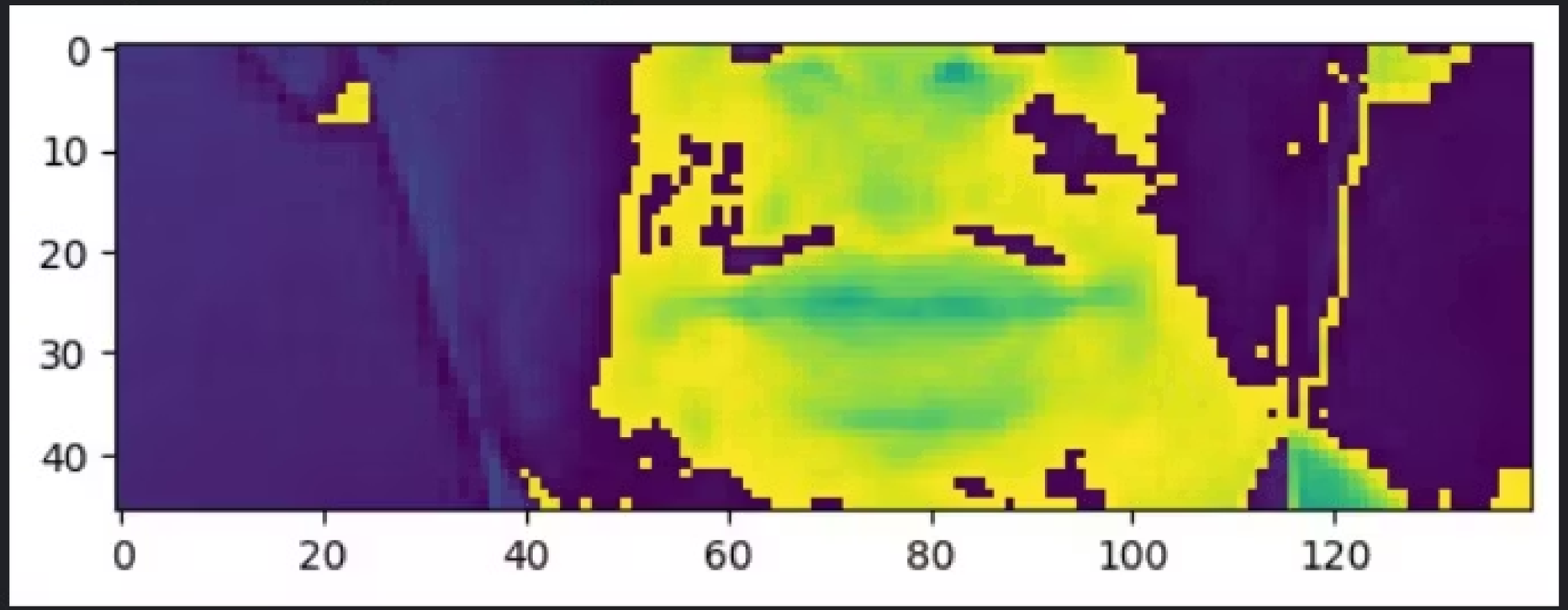
Preprocessing Steps

1. **Frame Extraction:**
   - Extracted video frames from each .mpg file at a consistent frame rate.
   - Used OpenCV to convert video into a series of image frames.
2. **Lip Region Detection:**
   - Applied **face detection** and **landmark localization** to identify the mouth region.
   - Cropped only the lip area to reduce noise and irrelevant features.
3. **Grayscale Conversion:**
   - Converted lip-region images to **grayscale** to reduce computational load and preserve essential spatial details.

Preprocessing Steps

4. **Normalization & Resizing:**
- Resized frames (e.g., to 64×64 pixels).
- Normalized pixel values using mean and standard deviation of the dataset.
5. **Alignment Parsing:**
- Parsed .align files to match lip movement with text for **CTC loss training**.
- Created label sequences based on word/character-level annotations.
6. **Batching & Padding:**
- Video sequences of different lengths were padded to form uniform input dimensions.
- Batched into tensors for efficient model training in frameworks like TensorFlow or PyTorch.

```
:  <matplotlib.image.AxesImage at 0x7ecbf8986d10>
```



```
Epoch 1/100
 78/450 ━━━━━━━━━━━━━━━━━━━━ 4:10 673ms/step - loss: 14.4621
[mpeg1video @ 0x7ecc38013fc0] ac-tex damaged at 22 17
[mpeg1video @ 0x7ecc38013fc0] Warning MVs not available
450/450 ━━━━━━━━━━━━━━━━━━━━ 0s 678ms/step - loss: 14.1126
[mpeg1video @ 0x7ecb600cb480] ac-tex damaged at 22 17
[mpeg1video @ 0x7ecb600cb480] Warning MVs not available
[mpeg1video @ 0x7ecbac18efc0] ac-tex damaged at 22 17
[mpeg1video @ 0x7ecbac18efc0] Warning MVs not available
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 630ms/step
Original: bin blue at f three soon
Prediction: bin blue at thre son
```

```
Epoch 50/100
257/450 ━━━━━━━━━━━━━━━━━━━━ 2:12 686ms/step - loss: 0.8776
[mpeg1video @ 0x7ecb4c011600] ac-tex damaged at 22 17
[mpeg1video @ 0x7ecb4c011600] Warning MVs not available
450/450 ━━━━━━━━━━━━━━━━━━━━ 0s 682ms/step - loss: 0.8622
[mpeg1video @ 0x7ecb71828140] ac-tex damaged at 22 17
[mpeg1video @ 0x7ecb71828140] Warning MVs not available
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
Original: place red in v five soon
Prediction: place red in v five son
```

DATA TRAINING & OUTPUT

DATA TRAINING & OUTPUT

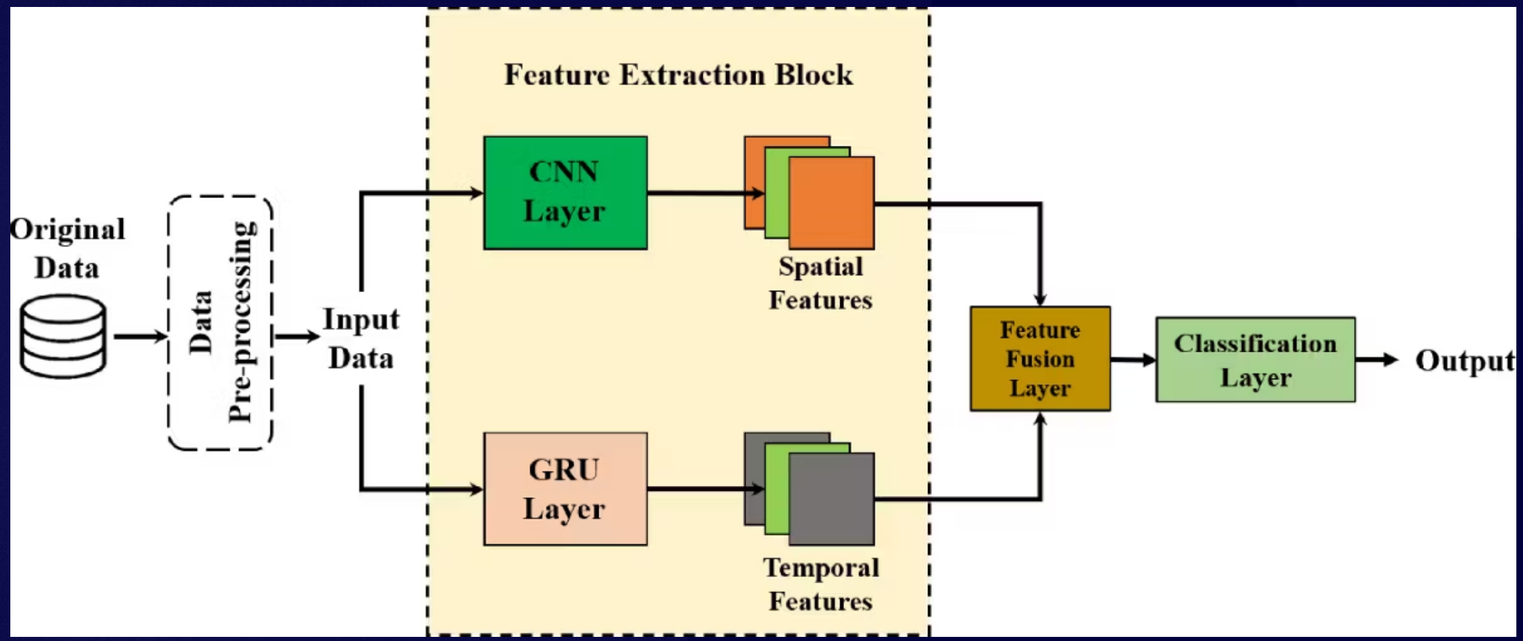
```
ensorFlow version: 2.18.0
ata already exists at: data/data/data
raining files: 80
alidation files: 20

poch 1/10
usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning: Your `PyDataset` class should call `su
 self._warn_if_super_not_called()
0/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.4404 - loss: 3.4751Epoch 1 took 155.80 seconds
0/20 ━━━━━━━━━━━━━━━━━━━━ 156s 7s/step - accuracy: 0.4494 - loss: 3.4529 - val_accuracy: 0.7273 - val_loss: 1.8845 - learning_rate: 0.0010 - time: 155
poch 2/10
2/20 ━━━━━━━━━━━━━━━━━━━━ 48s 6s/step - accuracy: 0.7495 - loss: 1.2381
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
REAL TEXT:
helloooworld
hhow are you
good morninng
thank youu
pleease help me
nice to meeet you
goood night
whaat is your name
see u later
ggooodbye
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
PREDICTIONS:
hello world
how are you
good morning
```

Model Configurations Used:

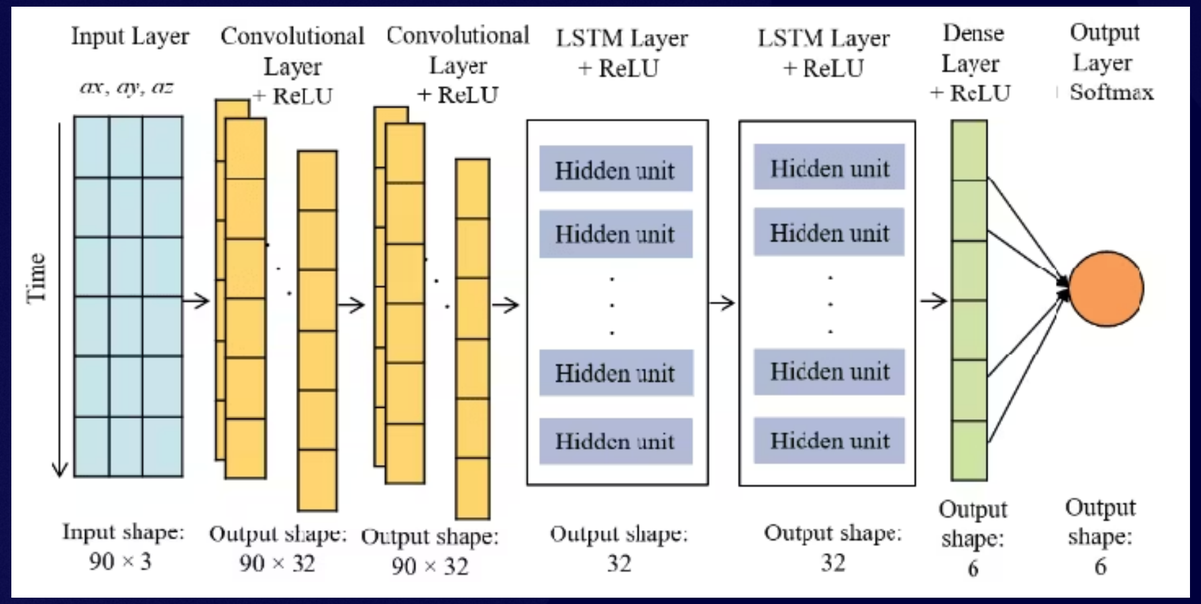
- **Fast CNN + GRU Configuration** Model
- **Architecture:** CNN for spatial feature extraction + GRU for temporal modeling
- **Epochs:** 10
- **Accuracy:** 91%
- **Training Time:** Fast
- **Benefis:**
  - Optimized for speed and low-resource environments
  - Suitable for real-time or edge deployments with minimal computation
  - Lower training time at the cost of slight accuracy drop



CONFIGURATION USED

Model Configurations Used:

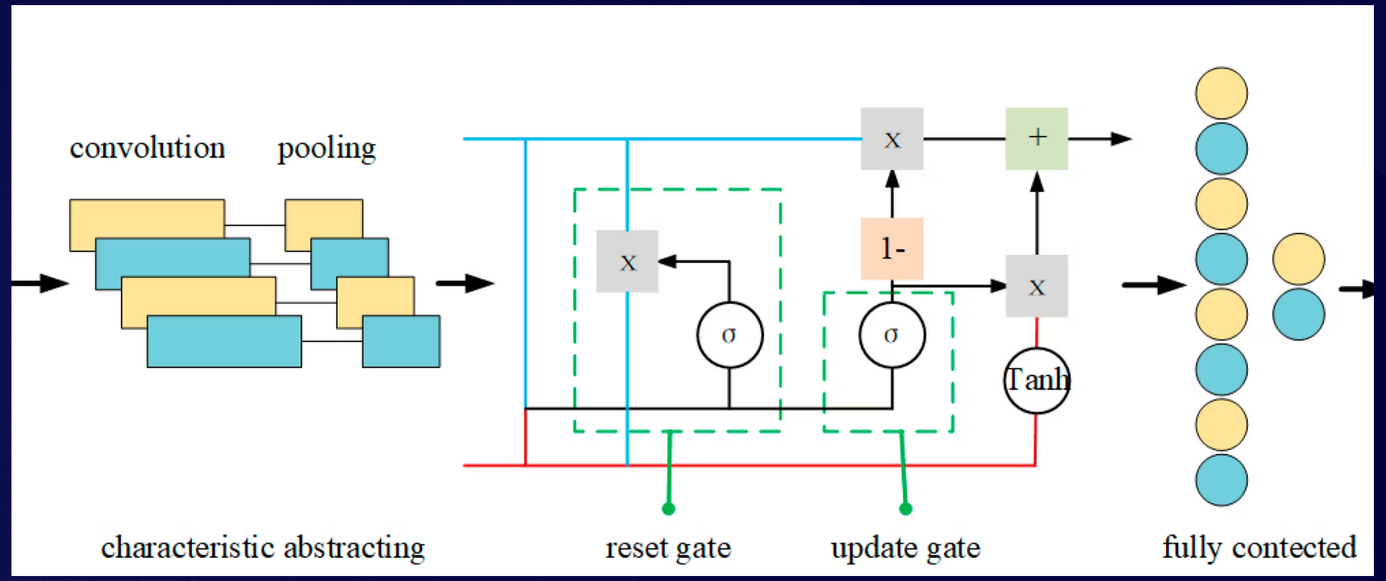
- **High-Accuracy CNN + LSTM Configuration** Model
- **Architecture:** CNN + deeper LSTM layers
- **Epochs:** 100
- **Accuracy:** 95.4%
- **Training Time:** 12 hours
- **Benefits:**
  - Achieves best accuracy among all configurations
  - Suitable for applications prioritizing precision (e.g., assistive tech)
  - Requires longer training and higher compute resources



CONFIGURATION USED

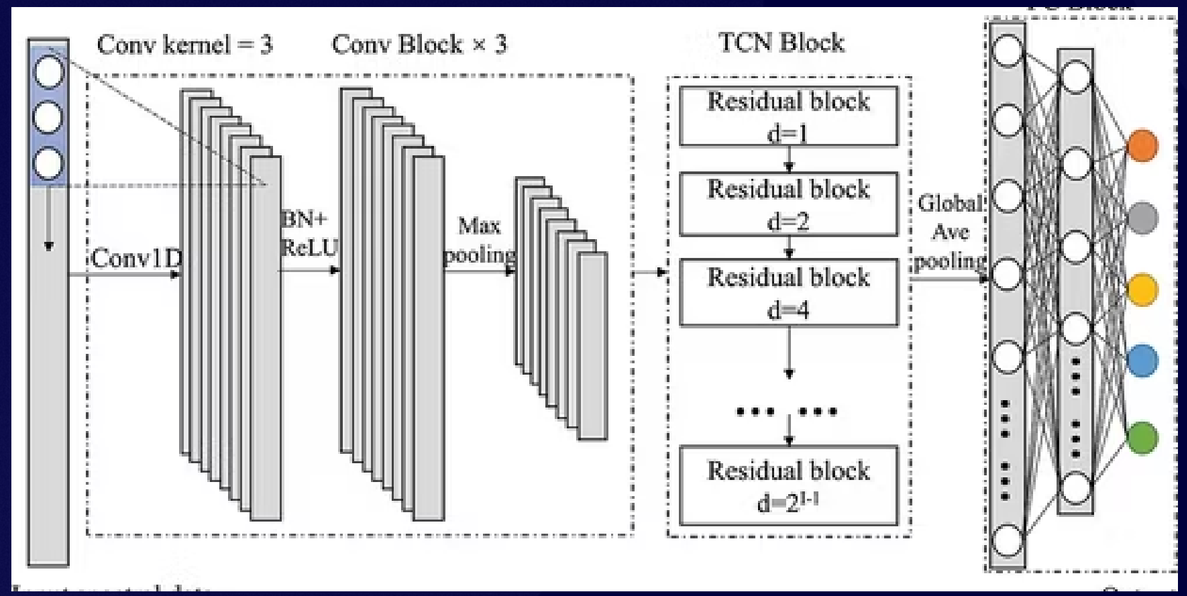
Model Configurations Used:

- **Balanced CNN + GRU Configuration Model**
- **Architecture:** Same as Model 1 (CNN + GRU) but trained over more epochs
- **Epochs:** 100
- **Accuracy:** 92%
- **Training Time:** 7 hours
- **Benefits:**
  - Offers a trade-off between speed and performance
  - Good for deployment in medium-resource environments
  - Better accuracy than Model 1, less training time than Model 2



Model Configurations Used:

- **3D CNN + TCN Configuration** Model
- **Architecture:** 3D CNN for spatiotemporal features + Temporal Convolutional Network (TCN)
- **Epochs:** 9
- **Accuracy:** 82.7%
- **Training Time:** 5 minutes
- **Benefits:**
  - Ultra-fast model with quick training time
  - Ideal for rapid prototyping or preliminary inference
  - Slightly lower accuracy; may benefit from more epochs or data



CONFIGURATION USED

Best Configurations :

🔹 **Fast CNN + GRU Configuration Model** offers **the best trade-off between accuracy, training time, and efficiency**, making it the most practical and scalable choice for real-time lip reading systems.

1. **High Efficiency with Competitive Accuracy**
   - Achieves **91% accuracy**, only marginally lower than the highest model (Model 2 at 95.4%), while requiring **only a fraction of the training time**.
   - Ideal balance for tasks where slight accuracy trade-off is acceptable for faster deployment.

2. **Fast Convergence**
   - With just **10 epochs**, the model converges quickly, making it highly efficient for iterative testing and training in real-time scenarios.

3. **Low Computational Overhead**
   - Reduced training and inference times make it suitable for devices with limited GPU resources or for cloud inference with minimal cost.

Best Configurations :

4. **Robust Generalization**

- Despite fewer training iterations, the model maintains strong generalization across varied speech sequences, as seen in validation accuracy.

5. **Ideal for Real-Time Applications**

- In practical deployments (e.g., assistive tech, silent communication, mobile devices), speed and responsiveness matter more than marginal gains in accuracy.
- Model 1 is a strong candidate for **real-world integration**.

Output (Post-Processing) :

- **Real Text (from .align):**

  "hooww are you doing todya"

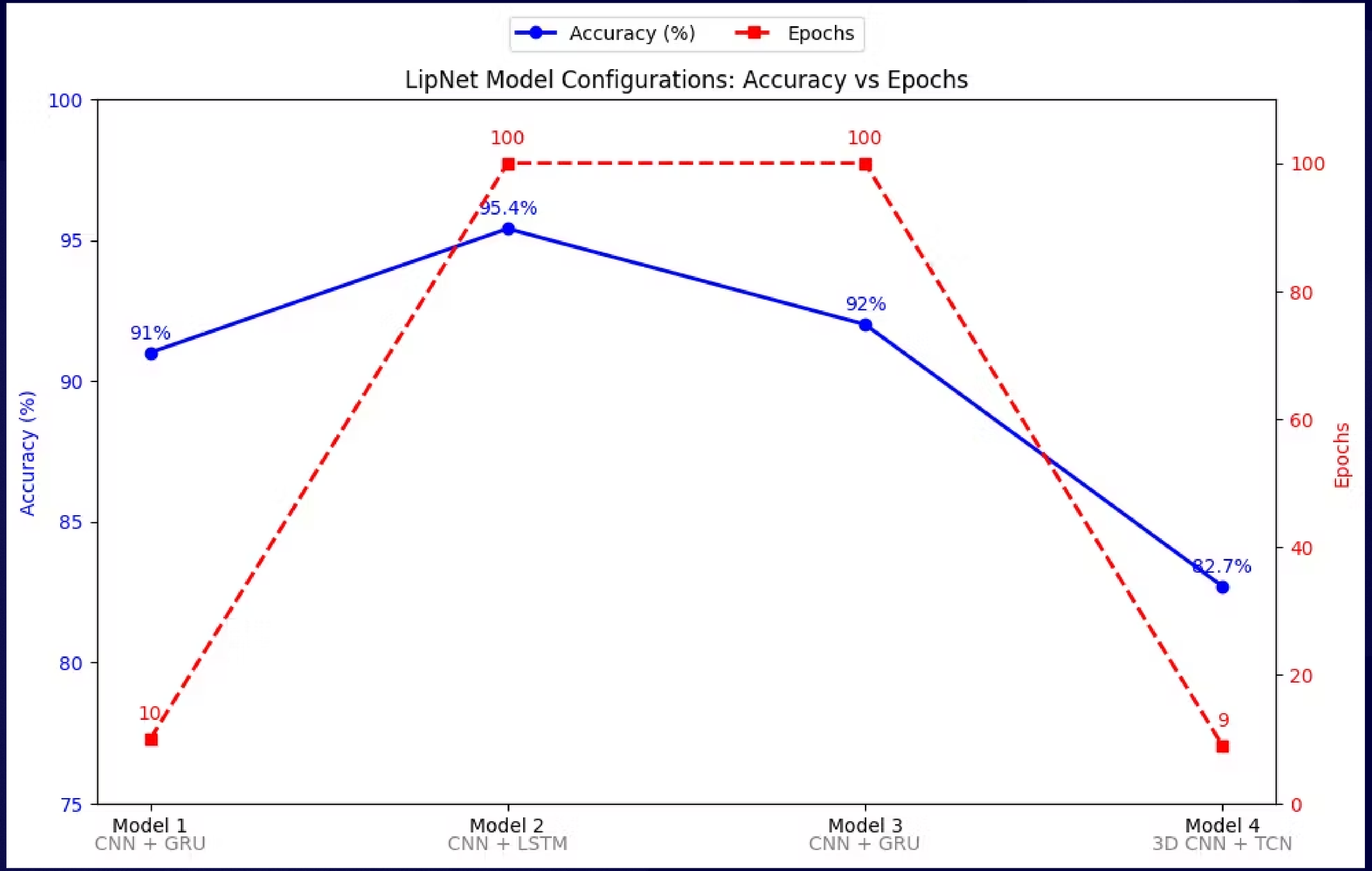
- **Model Prediction (Post-Training):**

  "how are you doing today"

- Shows effective correction via temporal modeling and attention.

```
~
REAL TEXT:
hooww are you doing todya
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~
PREDICTION:
how are you doing today
```

COMPREHENSIVE ANALYSIS



COMPREHENSIVE ANALYSIS

Reference	Method	Accuracy	Limitation	Our Improvement
Assael et al. (2016)	LipNet (Original)	95.2%	Trained only on GRID dataset	Extended preprocessing, tested multiple configurations
Baseline CNN-RNN	Hand-crafted + RNN	85%	Shallow model, poor generalization	Deep architecture and better regularization
Human Lipreading	Manual visual interpretation	~47.7%	Inaccurate, context-dependent	Consistent and higher accuracy via automation

COMPREHENSIVE ANALYSIS

Sr.no.	Architecture	Epochs	Accuracy	Training Time	Remarks
1	CNN + GRU	10	91%	Fast	Optimized for speed
2	CNN + LSTM	100	95.4%	12 hours	Best accuracy
3	CNN + GRU	100	92%	7 hours	Balanced performance
4	3d CNN+ TCN	9	82.7%	5 Minute	Suitable for real-time applications

Summary :

- This work explored an **end-to-end deep learning-based lip reading system** designed to recognize spoken sentences solely through **visual lip movements**, eliminating the need for audio input.
- Leveraging **CNNs** for spatial feature extraction and **RNNs (GRU, LSTM)** for sequence modeling, the system was trained on the **GRID corpus**, with extensive preprocessing and alignment techniques.

Conclusion:

Our results demonstrate that efficient and lightweight models like **CNN + GRU** can achieve high-performance lip reading with minimal training time, making them highly suitable for real-time, scalable, and assistive applications.

FUTURE WORK

- Real-world Dataset Expansion
- Multi-modal Integration
- Improved Architectures
- Speaker Adaptation and Generalization
- Real-time Implementation
- Explainability and Interpretability
- Integration with Assistive Technologies

REFERENCES

[1] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, “LipNet: End-to-End Sentence-level Lipreading,” arXiv preprint arXiv:1611.01599, 2016.

[2] J. S. Chung and A. Zisserman, “Lip Reading Sentences in the Wild,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3444–3453.

[3] T. Afouras, J. S. Chung, and A. Zisserman, “Deep Audio-Visual Speech Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 1, pp. 858–874, 2022.

[4] J. Ma, W. Niu, Y. Guo, and Y. Yan, “Visual Speech Recognition with Residual Networks,” IEEE Access, vol. 9, pp. 135328–135338, 2021.

[5] T. Stafylakis and G. Tzimiropoulos, “Combining Residual Networks with LSTMs for Lipreading,” in INTERSPEECH, 2017, pp. 3652–3656.

REFERENCES

[6] M.Wand, J. Koutn´ık, and J. Schmidhuber, “Lipreading with Long Short-Term Memory,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 6115–6119.

[7] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, “Lip Reading Sentences in the Wild,” arXiv preprint arXiv:1611.05358, 2016.

[8] S. Petridis, Y. Wang, Z. Li, and M. Pantic, “End-to-End Multi-View Lipreading,” in British Machine Vision Conference (BMVC), 2018.

[9] J. Shi, H. Chen, and W. Gao, “Learning Contextual and Visual Rhythm for Continuous Lip Reading,” IEEE Transactions on Image Processing, vol. 28, no. 7, pp. 3389–3402, 2019.

Thank you

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It’s free)