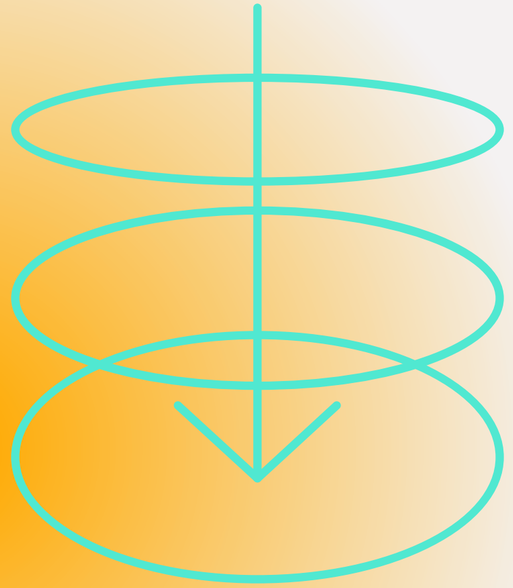
ML & Mime

Presentation by

Pampa, Pratiksha, Sayantika, Shubham



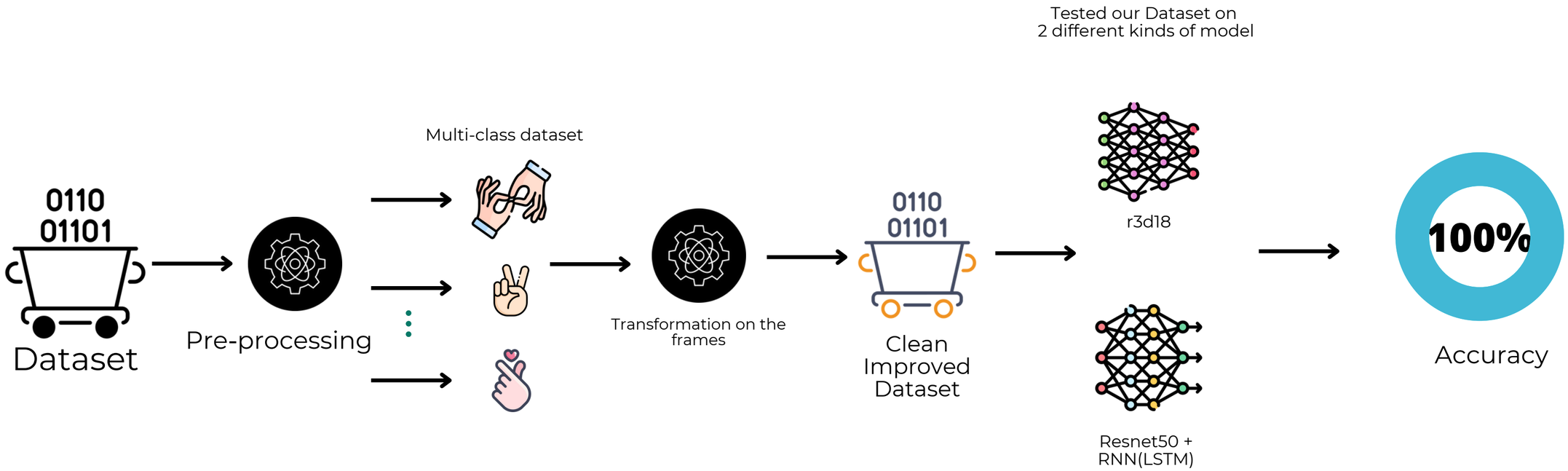
Project Overview

Goal:

Develop **robust system** for **Sign Language gesture recognition** from video sequences **using** a combination of Convolutional Neural Networks (**CNN**) and Recurrent Neural Networks (**RNN**).

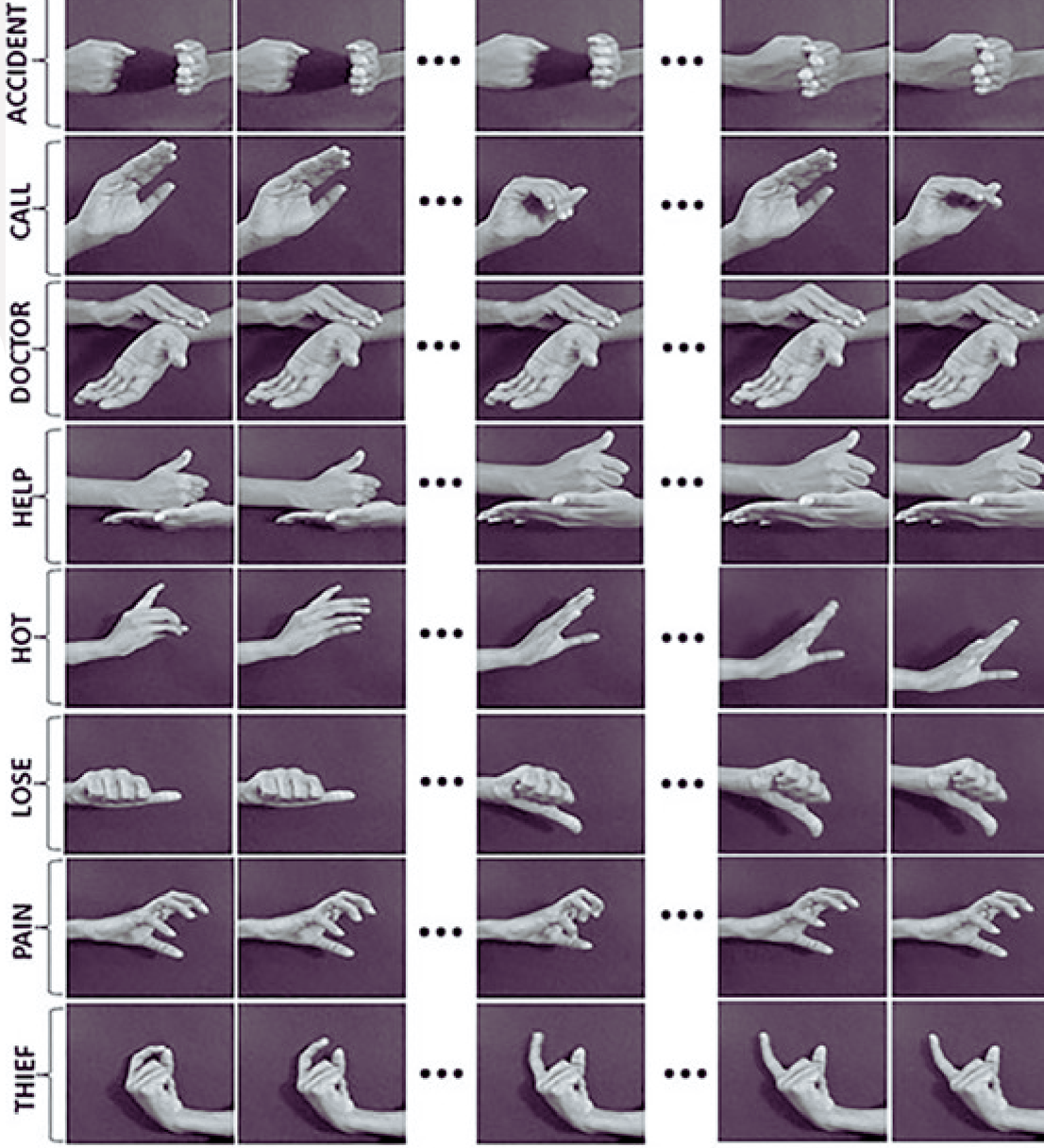
Goal is to **interpret** and **translate sign gestures** into **text. Enabling seamless communication** for individuals who use Sign Language.

Architecture

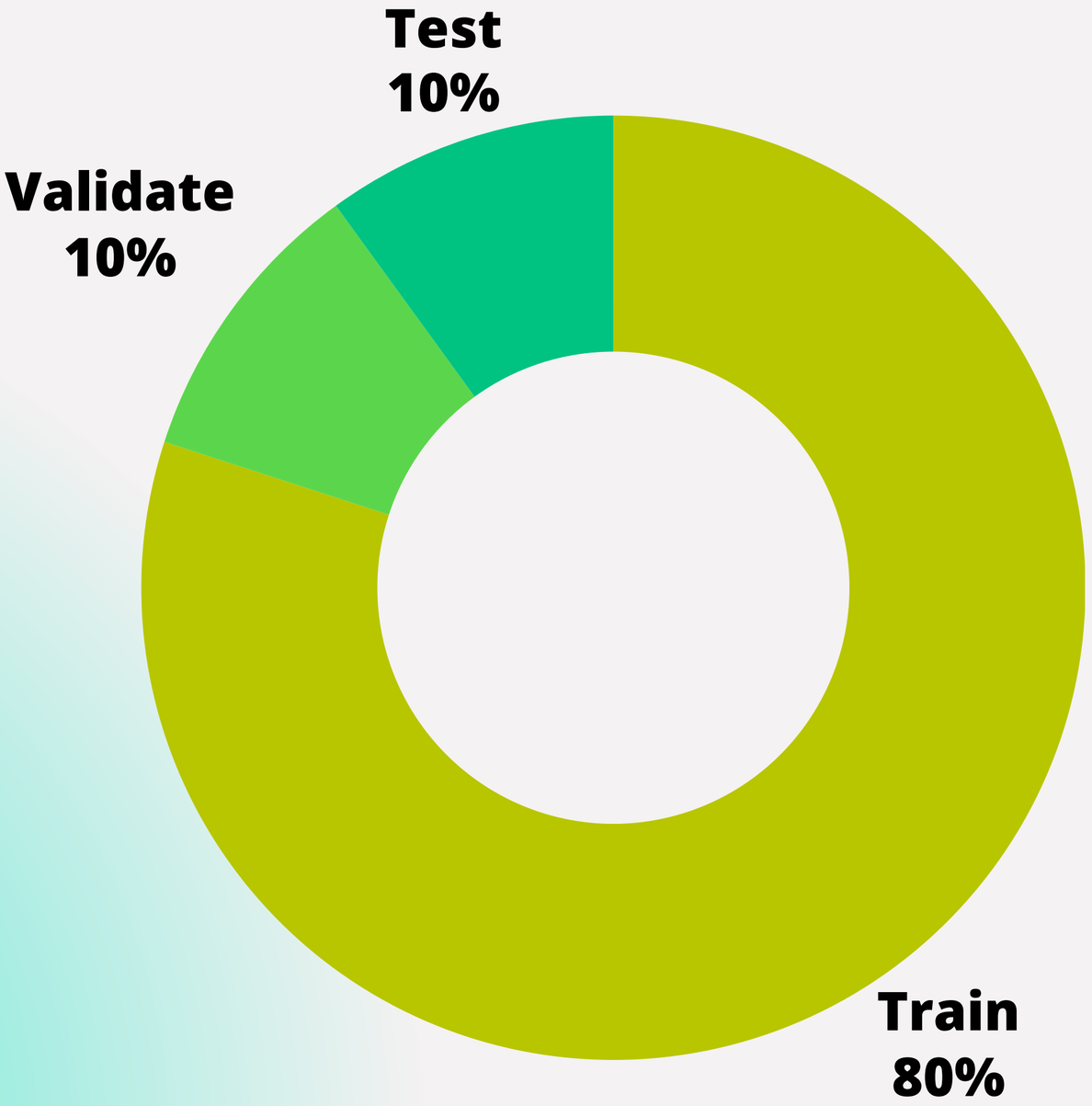


DataSet

- Dataset **contains** Hand Gestures of **Indian Sign Language words** used in **Emergency Situations.**
- The dataset **includes videos files** of the hand gestures of **eight words** (shown in the image)



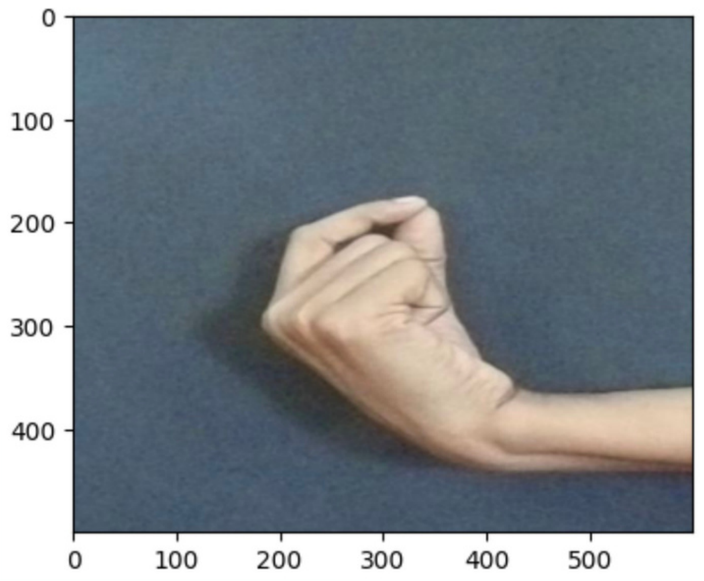
Dataset Split

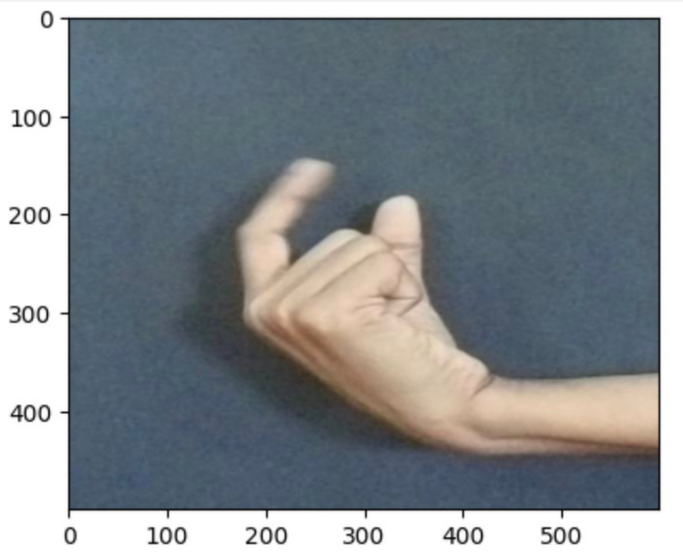


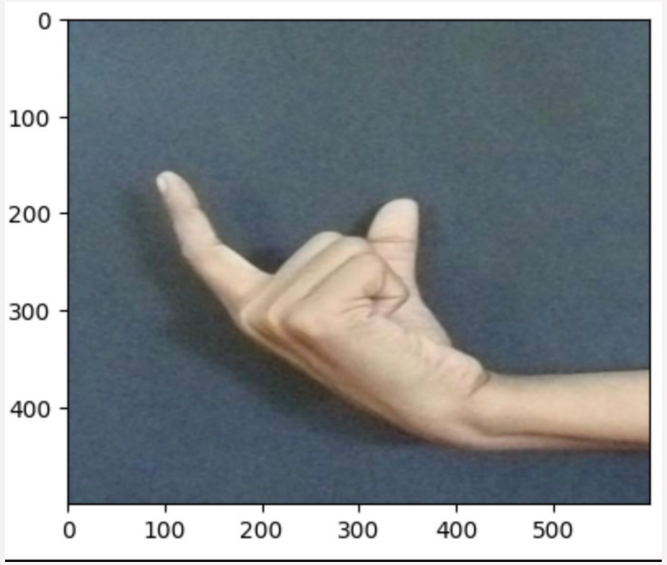
Data preprocessing

- The initial stage involves **extracting video data from zip files** and arranging it into dedicated folders.
- Leveraging **OpenCV's** capabilities (cv2), **frames** are systematically **extracted** from these videos
- We defined two helper functions to get frames (get_frames) and store the frames (store_frames) from a video.
- We used **16 frames** from each **video**.
- Extracted frames are cataloged as JPEG images within a freshly created folder, setting the stage for subsequent analysis and seamless integration into machine learning models for deeper processing and interpretation.

Extracted frames from video



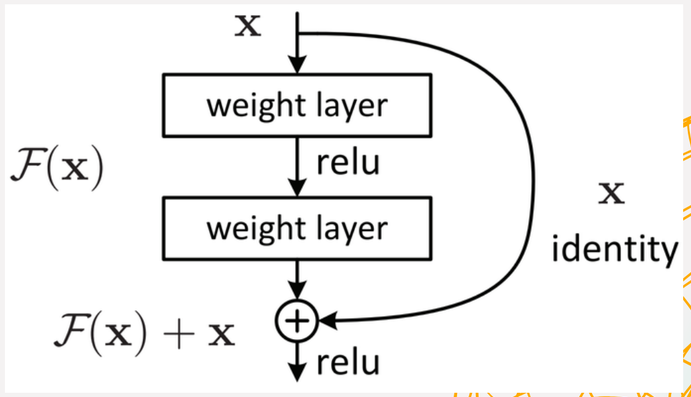




Techniques Implemented

- **ResNet50 + LSTM**
- **R3D_18**

Model-1

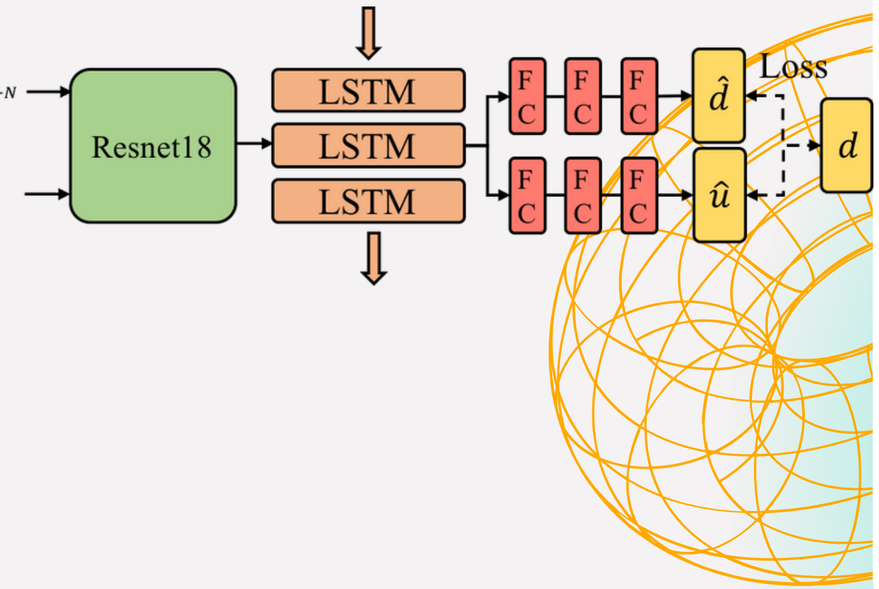


r3d_18:

- Residual networks basically allow **short-circuiting path** between different layers which in turn **benefits training**
- 3D convolutions focus on low/mid-level motion modeling, while 2D convolutions handle spatial reasoning, improving action recognition accuracy.

- **Our model**
  - **4** layers each containing 2 Blocks
  - one basic block downsamples the incoming nodes

Model-2



ResNet50 + LSTM:

- **Our model:**
- Utilised a **ResNet-50 architecture** as the backbone for an **RNN-based** model for **sequence analysis**.
- **Integrated** an **LSTM layer** into the **ResNet-50 architecture** to process sequential data, incorporating a three LSTM layer with **100 hidden units**.
- Leveraged a **pre trained ResNet-50 model** and adapted its final fully connected layer (fc) to accommodate sequential processing by replacing it with an identity layer.
- Configured the model's **output layer** to facilitate classification into **eight distinct categories**, aligning precisely with the task requirements and the number of classes in the dataset.

Model-2

ResNet50:

- ResNet-50: 50-layer architecture with residual blocks and skip connections.
- Utilizes skip connections to address vanishing gradient issues.

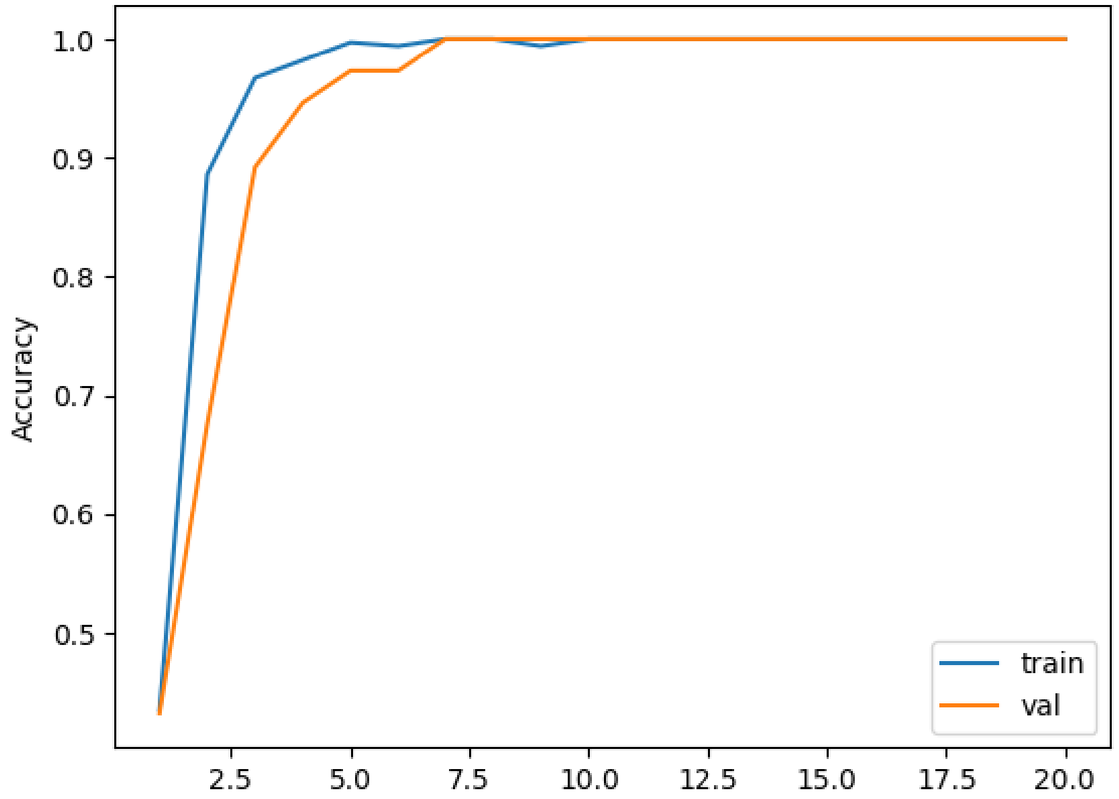
LSTM:

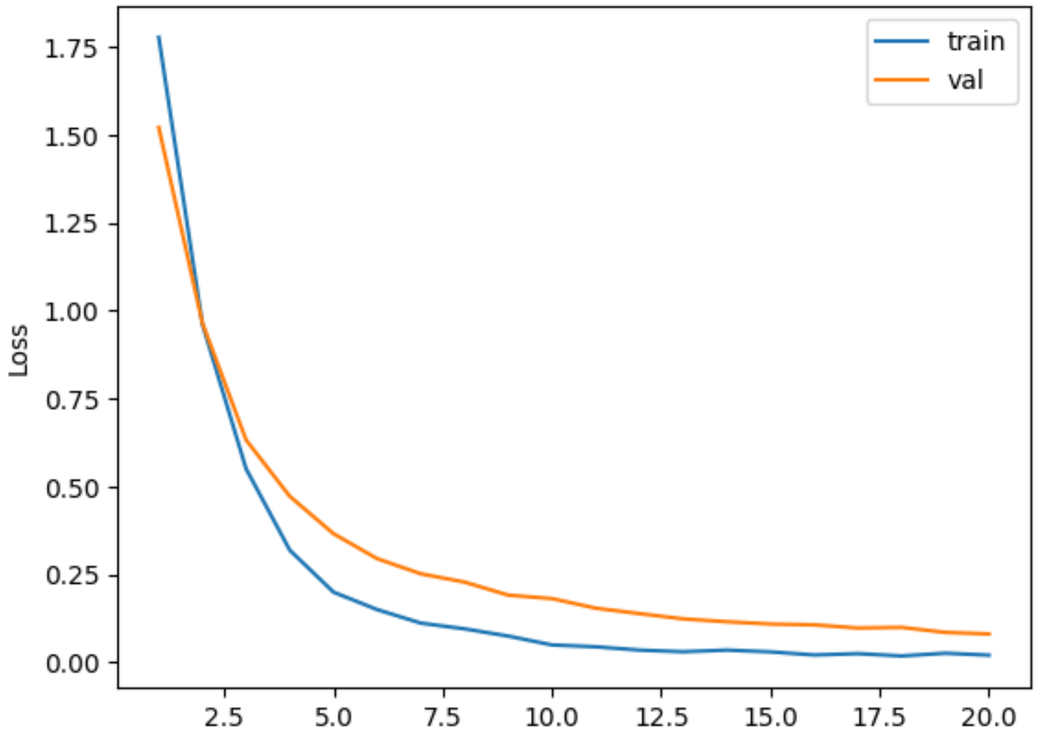
- Sequential Modeling: LSTMs excel in modeling sequences and time series data.
- Memory Cells: They possess memory cells to retain information over long sequences.

Results

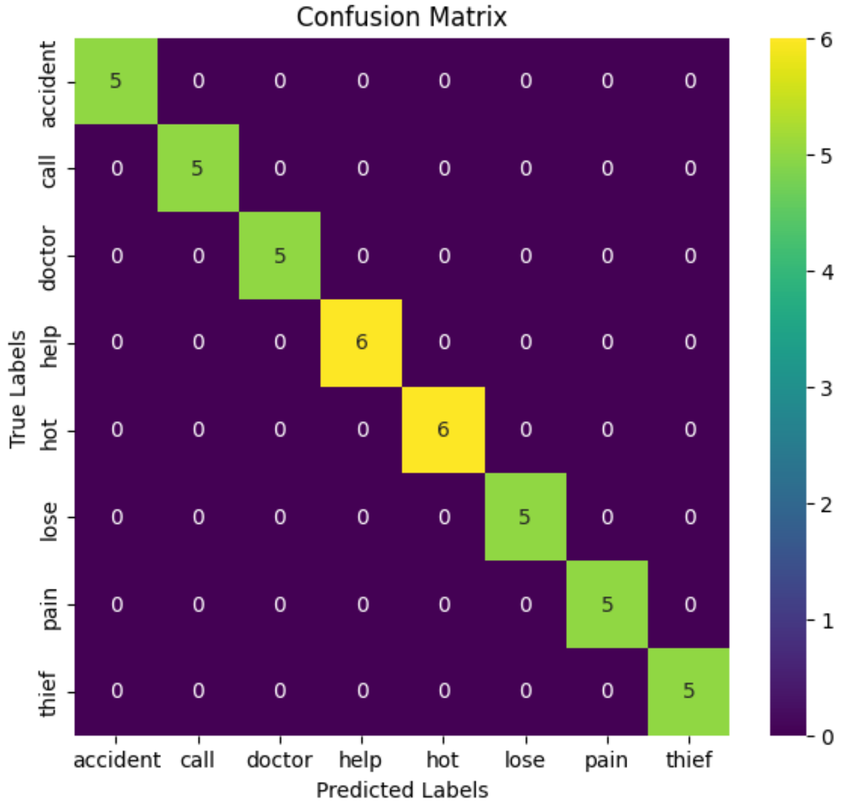
	Number of epoch	Batchsize	Precision	F1 score	Accuracy
r3d_18	**20**	**32**	**100**	**100**	**100**
ResNet50 + LSTM	**20**	**32**	**100**	**100**	**100**

Results (r3d_18)

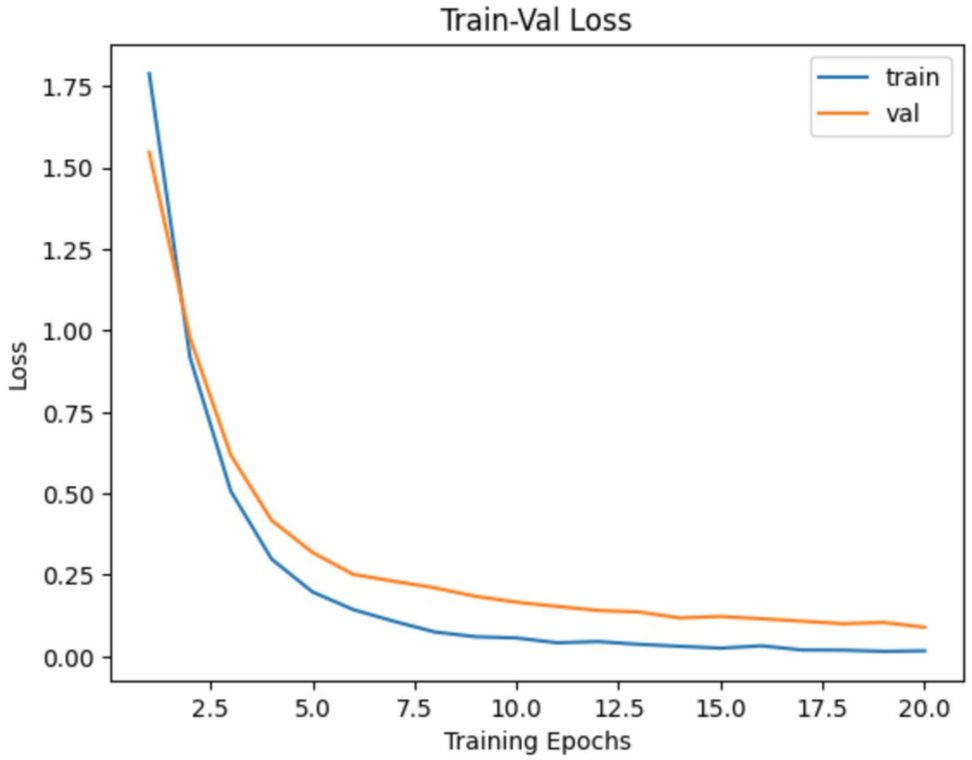


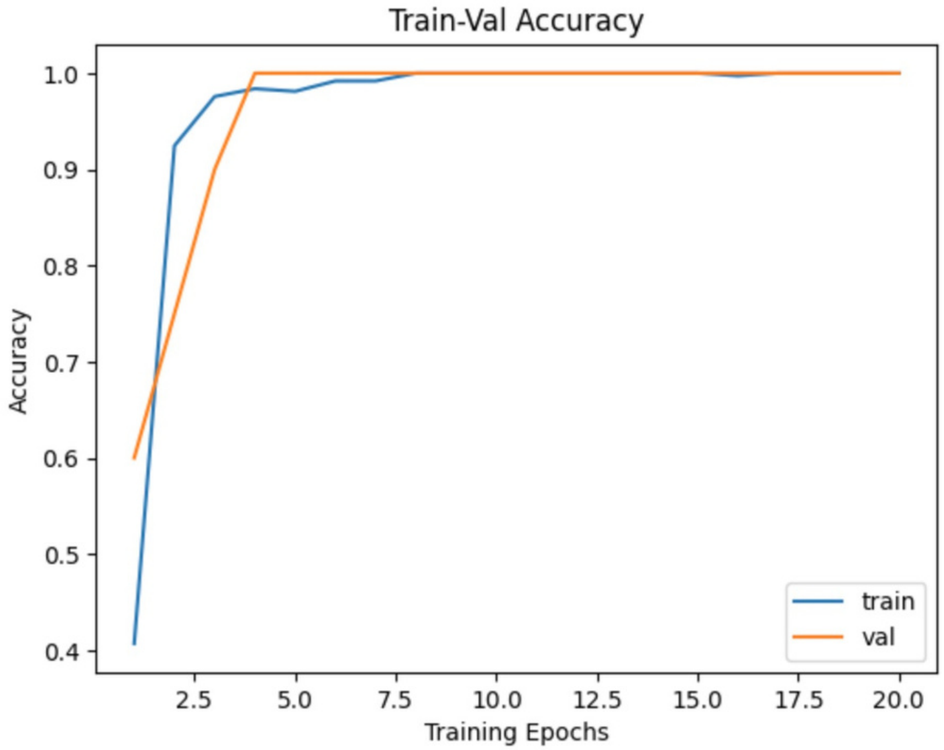


Results (r3d_18)

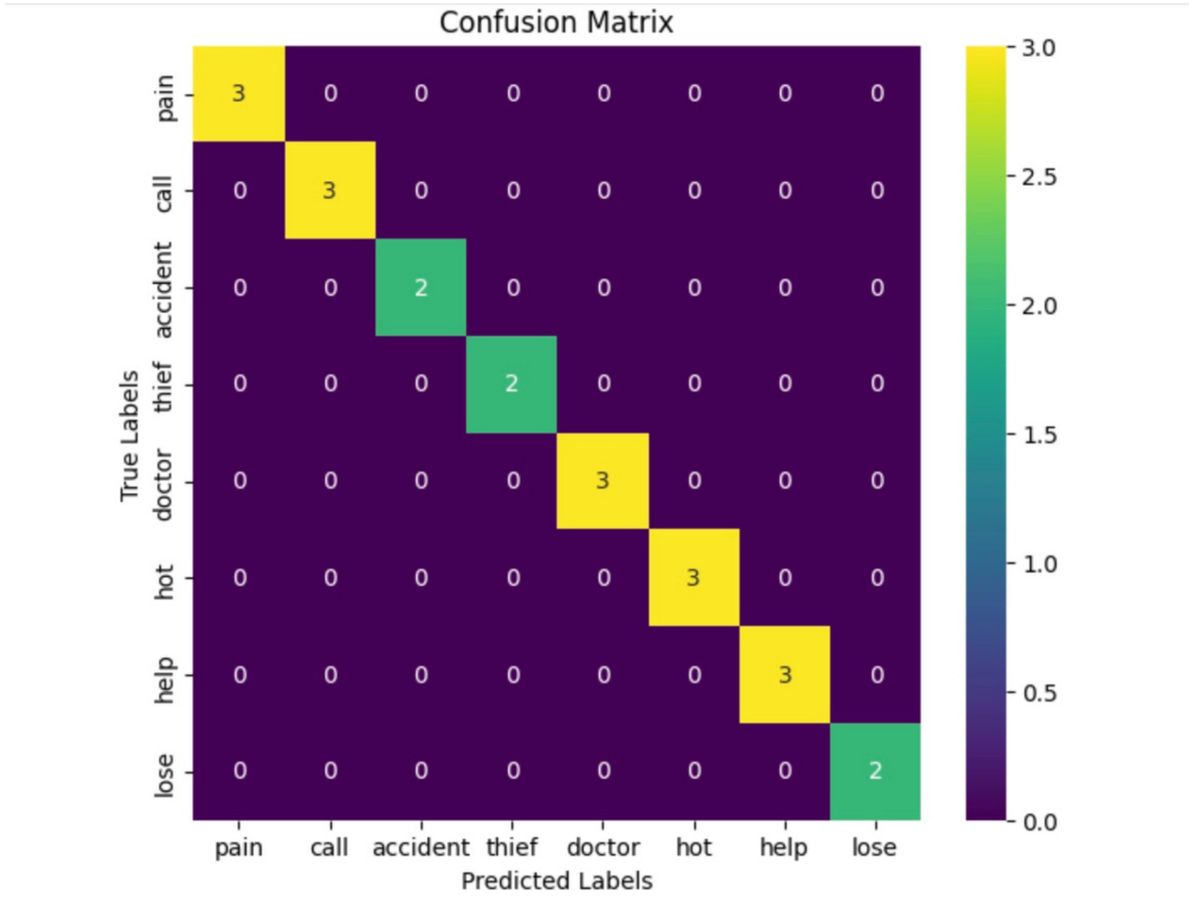


Results (Resnet50 + LSTM)





Results


Confusion Matrix
True Labels
pain
call
accident
thief
doctor
hot
help
lose
Predicted Labels
3.0
2.5
2.0
1.5
1.0
0.5
0.0

Sources

DataSet: https://data.mendeley.com/datasets/2vfdm42337/1
Paper: https://link.springer.com/chapter/10.1007/978-981-10-7566-7_63
Code Reference:
https://github.com/PacktPublishing/PyTorch-Computer-Vision-Cookbook
Other References:
https://medium.com/howtoai/video-classification-with-cnn-rnn-and-pytorch-abe2f9ee031

Contribution

Pampa
(model
development)

Sayantika
(model
development)

Pratiksha
(Pre processing)

Shubham
(pre processing)

Thanks