AMRITA
VISHWA VIDYAPEETHAM

Wrist Rehabilitation System

K. Nitya - CB.AI.U4AIM24123
B. Pavani - CB.AI.U4AIM24106
Shreeya Ardhra - CB.AI.U4AIM24105
Vinodh Hari - CB.AI.U4AIM24114
Krishna Iyer

DATA ACQUISITION

Collected Surface Electromyography (sEMG) signals by placing the gel electrodes on the Extensor Carpi Radialis (ECR) muscle once at 0°(relaxed position) and the other at 90°(extended position). In total 100 readings were collected at each position, and a dataset of 1000 samples was linearly interpolated for all wrist angles between 0° and 90°, with the corresponding EMG value along with the CTS severity labels based on clinical thresholds.

The four labels were created in the following manner:

- Angle > 35° → No CTS
- 26°–35° → Mild CTS
- 16°–25° → Moderate CTS
- ≤ 15° → Severe CTS

RNN

Recurrent neural networks (RNN) are designed for sequential data (e.g., text, speech, time series) where the order of inputs matters. Unlike regular neural networks, RNNs remember past information using a "memory" called the hidden state.

Model Architecture:

- SimpleRNN layer (64 units) with tanh activation
- Dropout layers (10%) for regularization
- Dense layers (ReLU activation) for feature learning
- Softmax output layer for multi-class classification

Training Process:

- 80-10-10 train-val-test split
- Adam optimizer with categorical cross-entropy loss
- 15 epochs with batch size 32

Conversion of model to TensorFlow Lite:

- Trained model is saved and loaded into memory.
- Prepares it for TFlite conversion.
- Optimization:
  - Quantization reduces the precision of numbers.
  - Pruning removes unnecessary parts of the model.
- Converted model is then saved, ready for deployment.

Setting up of cloud server by Flask API:

- Arduino reads EMG sensor
- Normalizes data
- Sends HTTP POST to local server.
- Server retruns prediction.
- Arduino shows results on OLED display.

LONG SHORT-TERM MEMORY (LSTM)

- Long short-term memory (LSTM) is a type of recurrent neural network (RNN) aimed at mitigating the vanishing gradient problem commonly encountered by traditional RNNs.

- LSTMs include gates and cell state to control what to remember or forget.

- Unlike RNNs, LSTMs add/remove information to the model using gates.

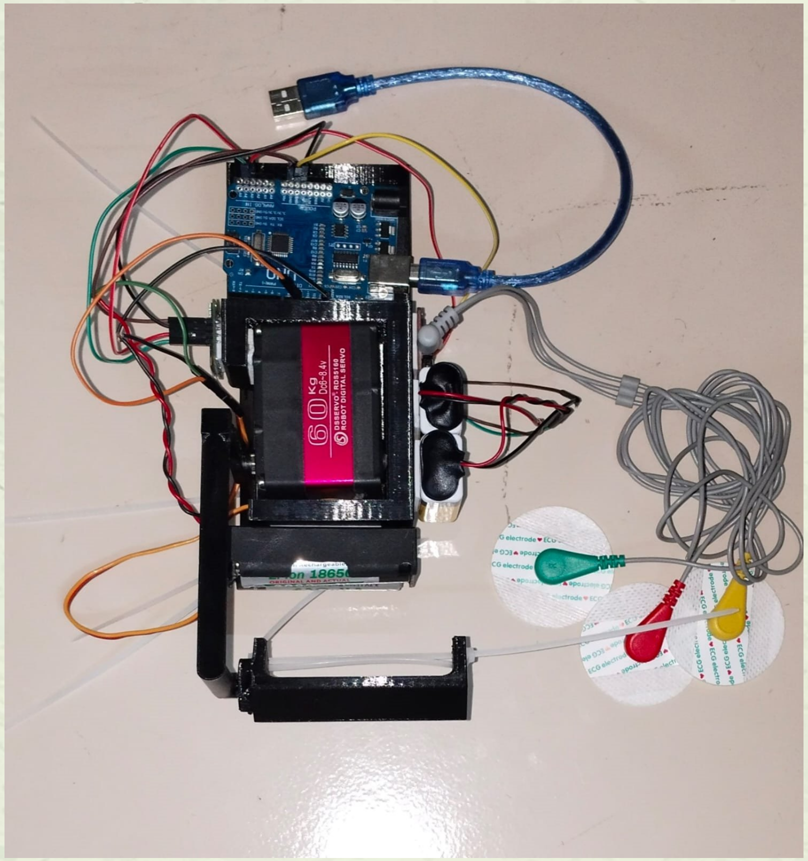
- In LSTMs cell state is carrier of long term memory through time.

- Due to the additive update mechanism, the LSTM’s memeory cell ensures gradients remain consistent over lenghty sequences.

MECHANISM

- Forget Gate: It uses the previous hidden state and current input to decide what to forget. By this the useless information is removed form the model.
- Input Gate: This gate proposes new values using a tanh layer then decides which parts to add to the cell state.
- Update the cell state: here both the forget gate and the input gate are combined.
- Output Gate: Decide what part of cell state to output as the hidden state.

HARDWARE INTEGARATION



The hardware includes:

- Arduino UNO
- RDS5160 Servo motor
- O-LED
- EMG sensor
- 3D printed braces

RESULTS AND COMPARISON

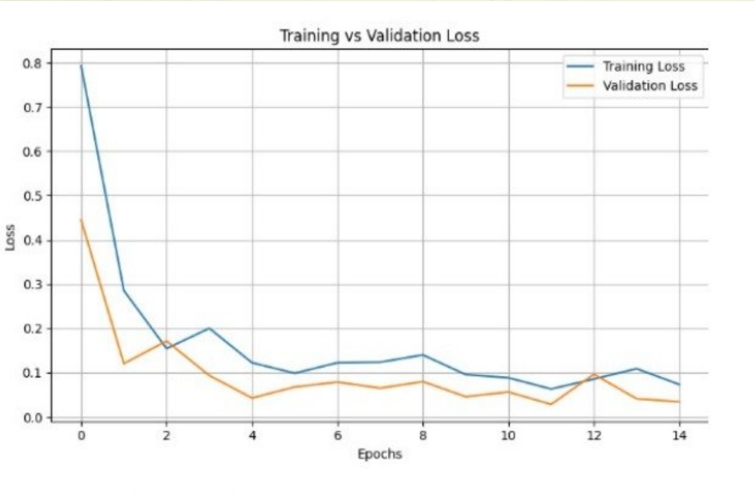


fig1. Training and Validation loss of LSTM model

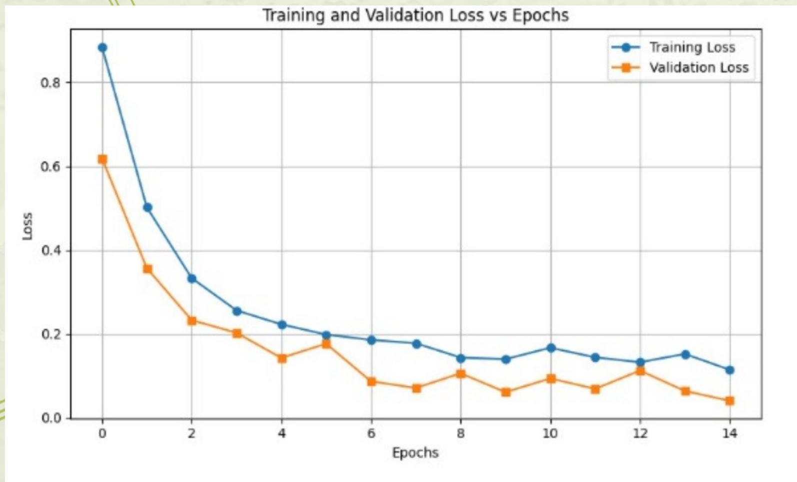


fig2. Training and Validation loss of RNN model

PERFORMANCE COMPARISON

Feature	RNN Model (%)	LSTM Model
Model Architecture	Simple RNN	LSTM
Hidden Layers	1 RNN + 2 Dense	2 LSTM + 1 Dense
Activation Functions	tanh,softmax	tanh,relu, softmax
Train Accuracy	0.9785	0.9937
Validation Accuracy	0.9798	0.9899
Test Accuracy	0.9697	0.9798
Overfitting Risk	Higher	Lower (due to dropout)
Dropout Used	0.1	Yes (0.1 after each LSTM)
Epochs	15	15
Generalization	Moderate	Strong