Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

LinguSign

A Bidirectional Romanian Sign Language Translator

Candidate: Radu-Cătălin MICEA

Scientific coordinator: Lect.Dr.Eng. Răzvan-Dorel CIOARGĂ

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Introduction

- Project inspired by a friend who volunteers with deaf and hard-of-hearing people.
- Currently no automated tool for sign language translation
  => difficult communication between deaf and hearing people.
- The tool should be portable and provide real-time bidirectional Romanian Sign Language translation.

The solution is a mobile application that:

1. Animates a 3D model to gesture the signs of the spoken words
2. Predicts and speaks out loud the recorded gestured signs

<<<<<

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

MediaPipe

A suite of solutions developed by *Google* to quickly apply AI and ML.

1. ***MediaPipe Hands*:** extracts the 21 landmarks on the human hand as 3D points.

2. ***MediaPipe Pose*:** extracts the 33 landmarks on the human body as 3D points.

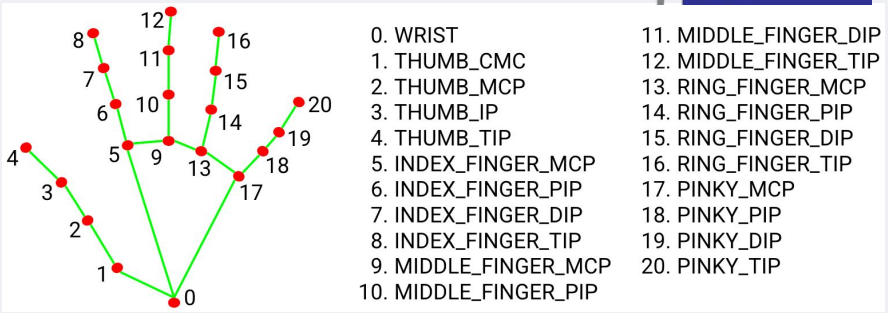

Fig. 1: Hand Landmarks. Accessed: 2024-06-04.

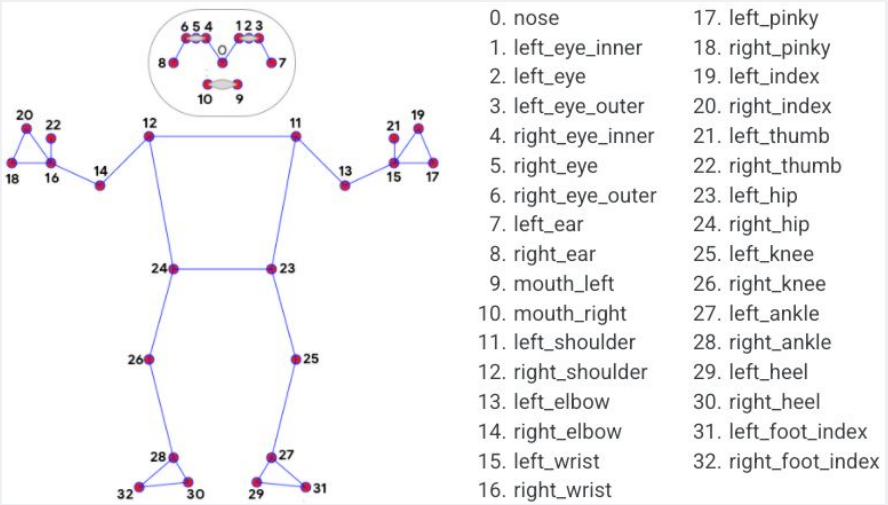

Fig. 2: Pose Landmarks. Accessed: 2024-06-04.

Kalman Filter

- Uses inaccurate or noisy measurements to estimate the state of that variable or another unobservable variable with greater accuracy.
- First two measurements initialize the system parameters.
- From the third measurement, it consists of two steps:
  1. Prediction step
  2. Update step

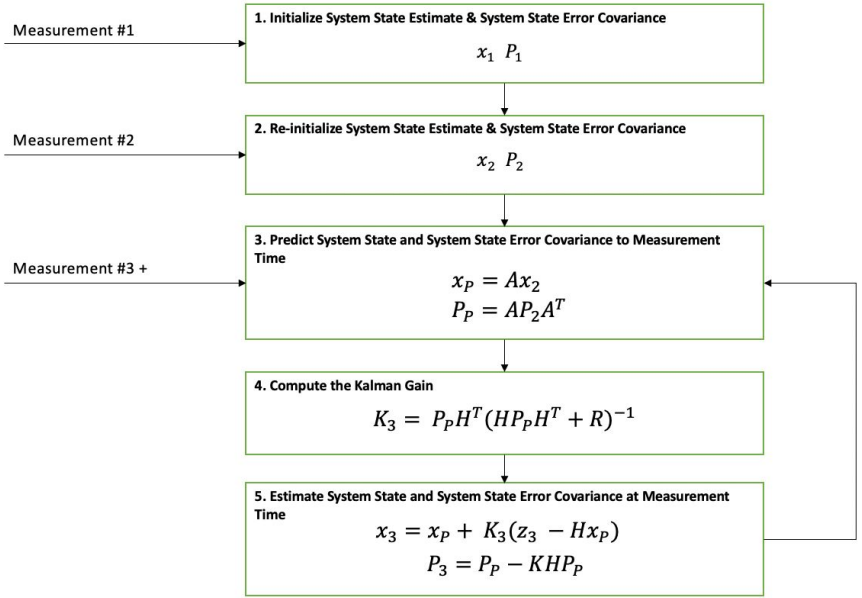

Fig. 3: Kalman Filter Overview. Accessed: 2024-05-29.

Moving Average Filter

- Calculates the unweighted mean of the *m* data points surrounding a central value (including the central value):

$$SMA_n = \frac{1}{m} \sum_{i=-k}^{k} p_{n+i}$$

where $m = 2k + 1$.

- Efficiently implemented using the convolution operation:

$$(x * h)[n] = \sum_{i=0}^{m-1} x[i]h[n-i]$$

where x is the input data and $h = \frac{1}{m}[1, 1, ..., 1]$ (m elements).

S. W. Smith, *The scientist and engineer's guide to digital signal processing*. USA: California Technical Publishing, 1997, ch. Chapter 15: Moving Average Filters, ISBN: 0966017633.

Architecture of the Backend

The backend consists of:

1. **Web API**
2. **Database** – contains user info and references to resource files
3. **Internal NLP service** – matches the Romanian words to the constrained RSL vocabulary
4. ***ChatGPT* API service** – builds coherent sentences from given words

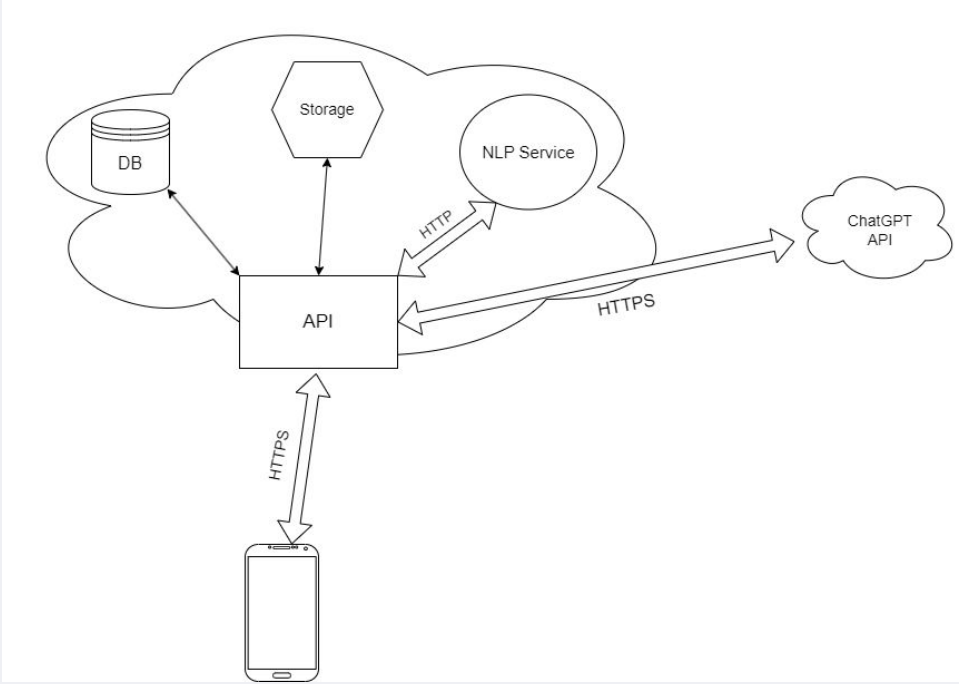

Fig. 4: High-Level Overview of the Backend.

Architecture of the Client Application

1. AnimationFragment

- Performs the RO to RSL translation.
- *WordsRecognizer*, for each spoken word, sends the list of words spoken thus far to the backend.
- The response consists of the matches in the constrained vocabulary of only the new words.
- *AnimationFragment* polls the responses queue when ready to animate and applies the filters first.

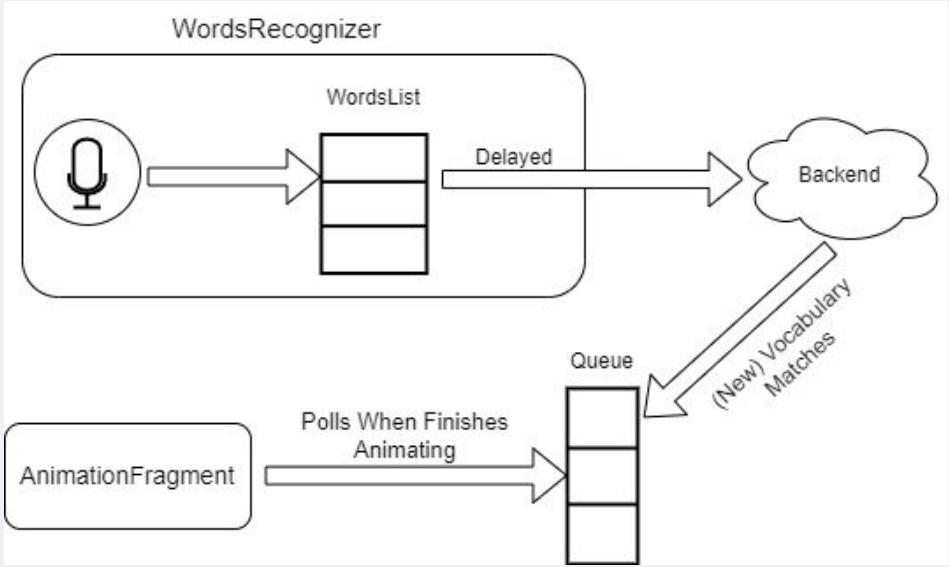

Fig. 5: High-Level Overview of *AnimationFragment* and Helpers.

Architecture of the Client Application

2. CameraFragment

- Performs the RSL to RO translation.
- *Landmarker* extracts the landmarks from the camera feed.
- *Translator* extracts features from the landmarks and enqueues them into a one second window.
- For every frame, the *classifier* is ran on the window. Its prediction is further analyzed for consistency.
- The predicted word is either spoken out loud, or sent along with other words to the backend. The resulting coherent sentence is then spoken out loud.

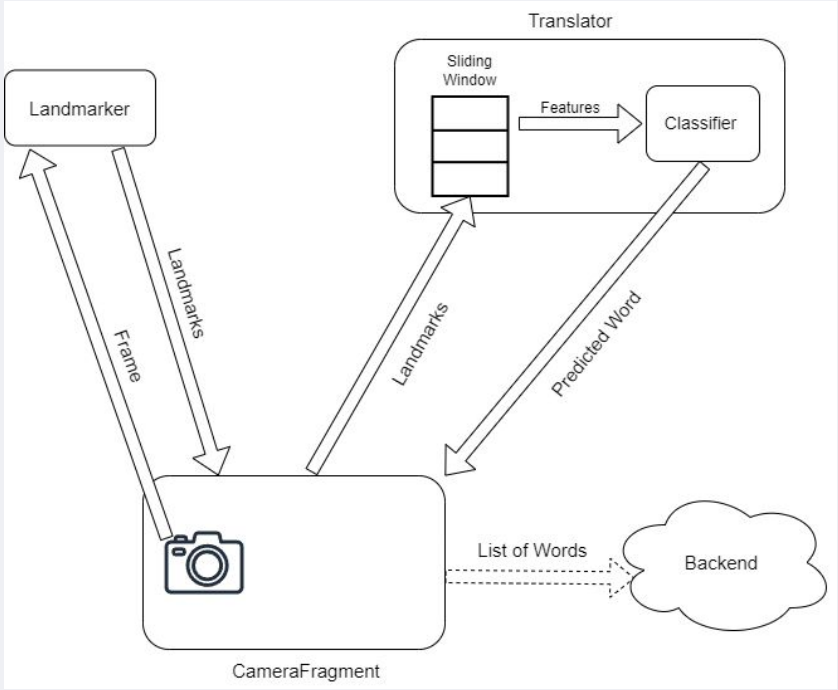

Fig. 6: High-Level Overview of *CameraFragment* and Helpers.

<<<<<

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Implementation of the RO to RSL translator

1. Creating the Constrained Vocabulary

1. Scrape videos from https://dlmg.ro/ and https://pesemne.ro/.

2. Extract landmarks from the videos using *MediaPipe*.

This is the **vocabulary**.

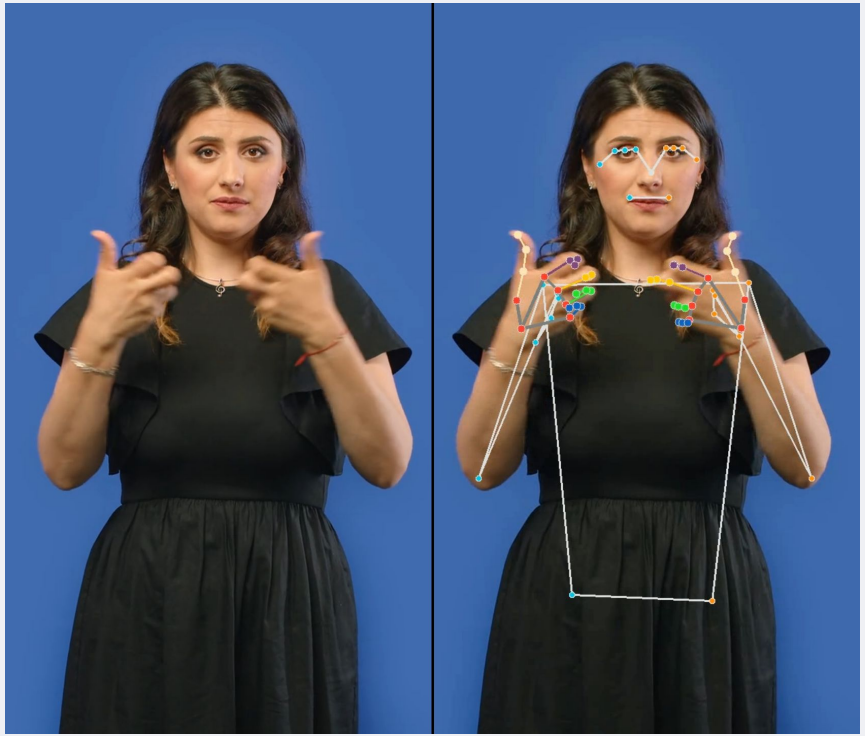

Fig. 7: Frame for the Sign "Examen" before and after Processing.

Implementation of the RO to RSL translator

2. Correct the Extracted Landmarks

The Kalman Filter is applied to predict missing landmarks and adjust existing ones.

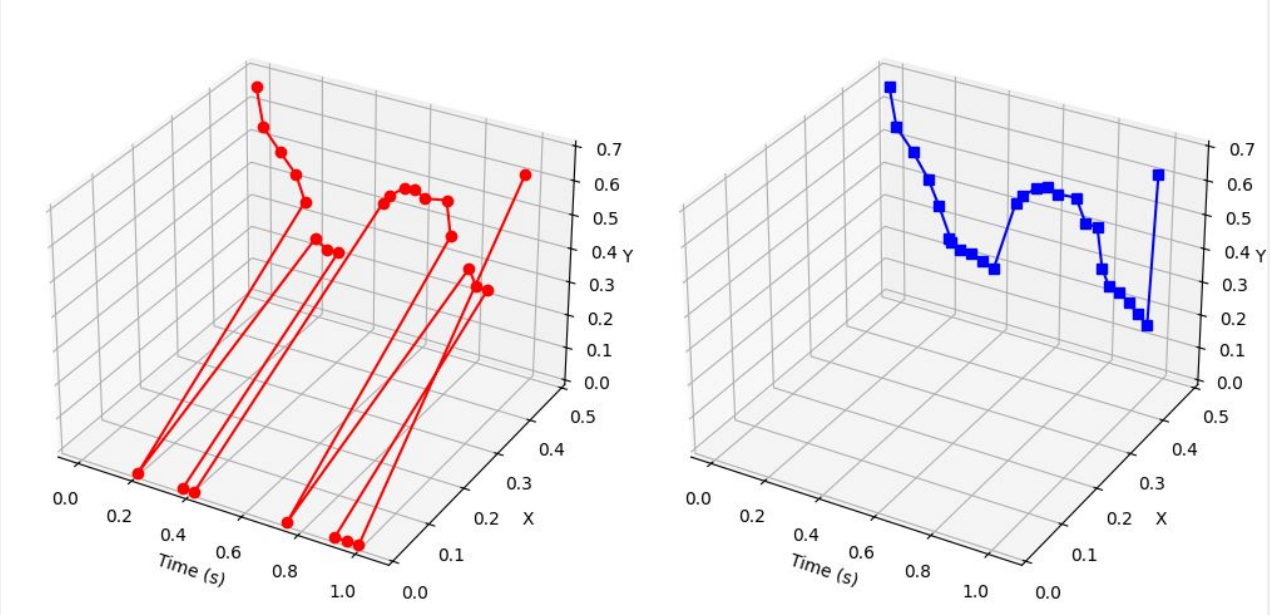

Fig. 8: "Poezie" XY Index Finger Tip before and after Kalman Filter.

Implementation of the RO to RSL translator

3. Smooth the Corrected Landmarks

The moving average filter is used to smooth the sequence of landmarks.

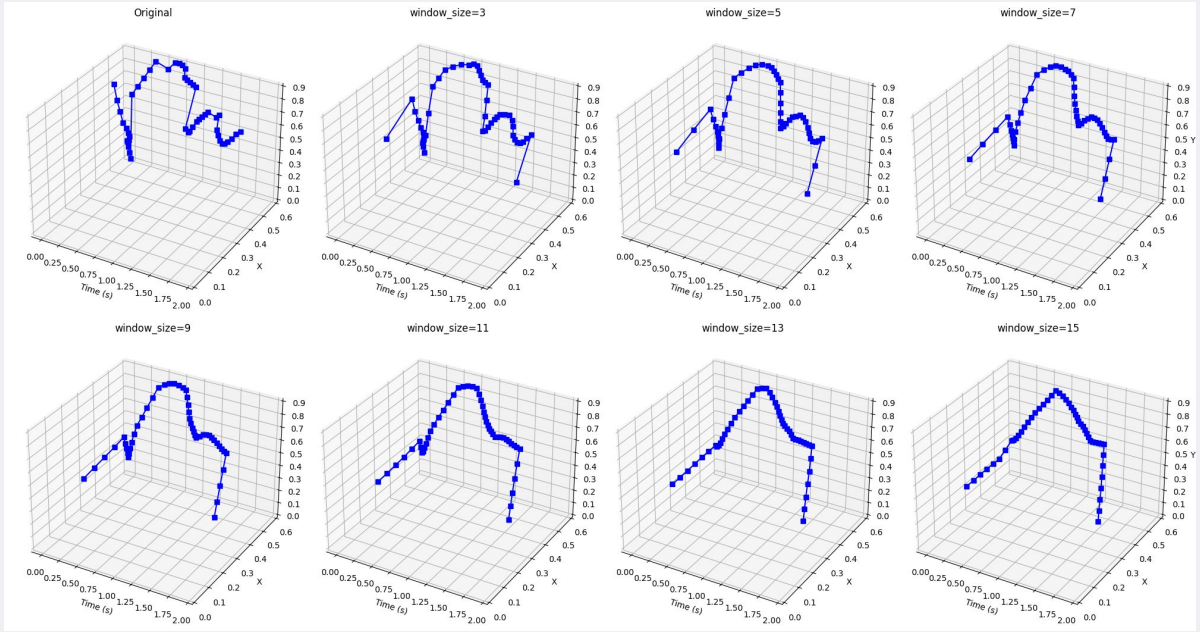

Fig. 9: "Volan" XY Index Finger Tip for Different SMA Window Sizes.

Implementation of the RO to RSL translator

4. Animate a 3D model based on the landmarks

Fundamental technique used to animate the model: **Aligning one unit vector (A) to another (B).**

$$R_{uvw} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} A \cdot B & -\|A \times B\| & 0 & 0 \\ \|A \times B\| & A \cdot B & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The matrix represents the rotation in the following base: $C = [uvw]$

Where u, v, w column vectors:

- u – normalized vector projection of B onto A
- v – normalized vector rejection of B onto A
- w – normalized cross product of B and A

Final transformation: $C^{-1} R_{uvw} C = C^T R_{uvw} C$

https://math.stackexchange.com/users/76513/kuba-hasnt-forgotten-monica, Calculate rotation matrix to align vector a to vector b in 3d? Mathematics Stack Exchange, URL:https://math.stackexchange.com/q/897677 (version: 2018-09-12). eprint: https://math.stackexchange.com/q/897677. [Online]. Available: https://math.stackexchange.com/q/897677.

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Implementation of the RO to RSL translator

4. Animate a 3D model based on the landmarks

1. Extract the initial transformation components from the 3D model to be animated.
2. Transform the 3D landmark positions of the video into world coordinates.
3. Compute the rotations needed to align the model's T-Pose to the position defined in the extracted landmarks along the kinematic chain.
4. Align the hand's extracted landmark's basis to the lower arm's and continue along the kinematic chain to compute the fingers' rotations.

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Implementation of the RO to RSL translator

5. The NLP Service

- Takes as input a natural, spoken sentence.
- Returns a list of matches from the vocabulary.
- Matches the words or their lemmas or stems through a heuristic combining the Levenshtein and cosine distances.

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Implementation of the RSL to RO translator

1. Creating the Dataset

- Recorded 30 one second videos containing only the main part of the gesture for each sign.
- Extracted the landmarks with MediaPipe.

2. Feature Engineering

The model should focus on the gesture and be invariant to the person's position relative to the camera.

=> Extract features relative to the person from the landmarks.

Implementation of the RSL to RO translator

2.1. World Landmarks

- *MediaPipe* also extract "World Landmarks"
- For *MediaPipe Hands*, they are the distance in meters from the hand's center.
- For *MediaPipe Pose*, they are the distance in meters from the hips' center.

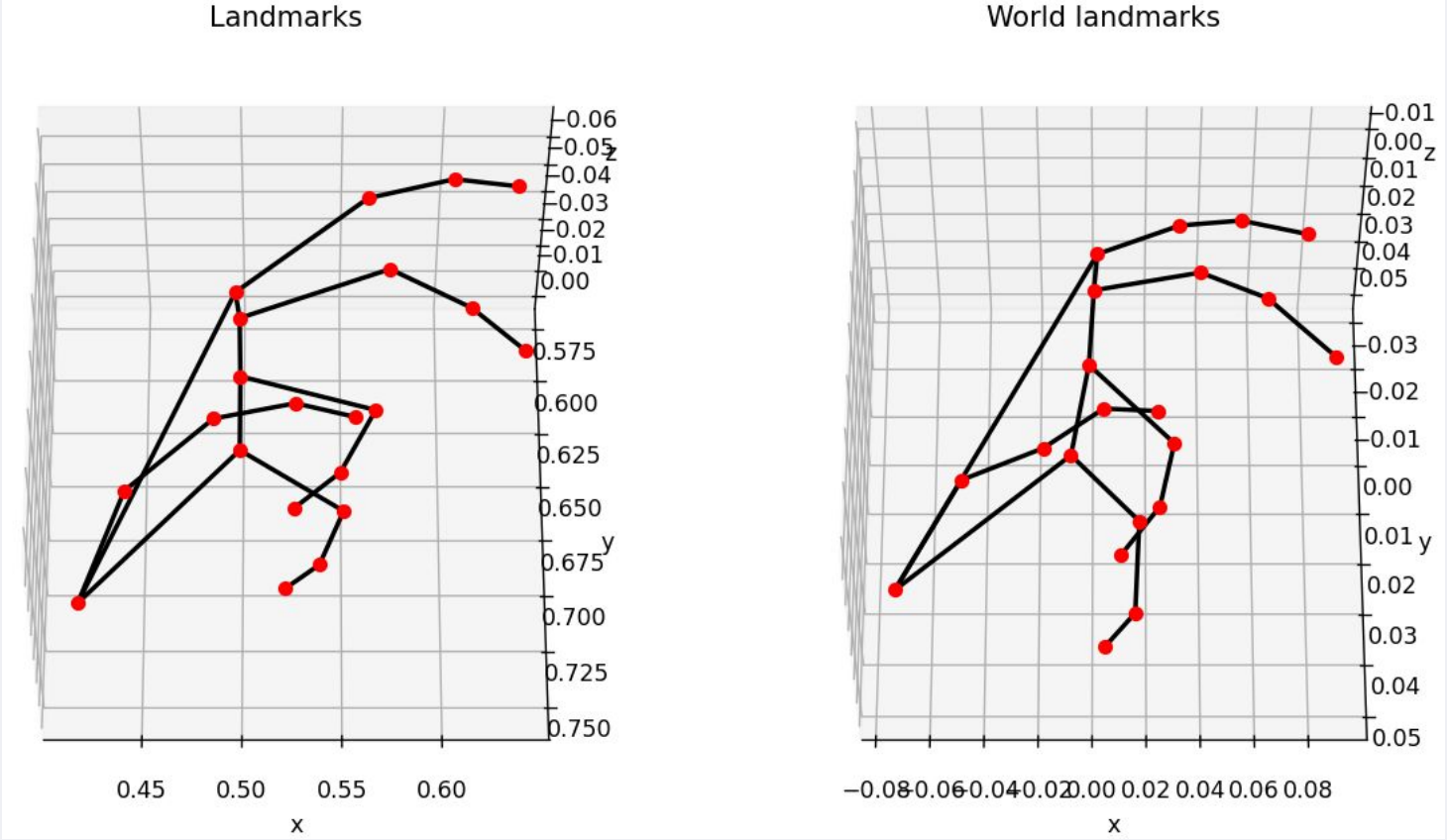

Fig. 10: High-Level Overview of the Backend.

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Implementation of the RSL to RO translator

2.2. Movement Direction

- **Problem:** World Landmarks lose information regarding the movement's direction.
- **Solution:** compute and normalize the difference in position between corresponding landmarks in consecutive frames.

2.3. Joint Rotations

- The rotations to align the model to the landmarks.

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Implementation of the RSL to RO translator

3. Gestures Classifier

- 3 LSTM layers followed by 3 dense layers.
- Recurrent dropout after every LSTM layer to prevent overfitting.
- Activation function: *ReLU* for hidden layers and *Softmax* for output layer.
- Loss function: *Categorical cross-entropy*.
- Optimizer: *Adam*
- Callbacks: LR scheduler, early stopping and model checkpoint.

```
gestures_model = Sequential([
    LSTM(64, return_sequences=True, activation='relu',
         input_shape=(X.shape[1], X.shape[2]), recurrent_dropout=0.2),
    BatchNormalization(),
    LSTM(128, return_sequences=True,
         activation='relu', recurrent_dropout=0.2),
    BatchNormalization(),
    LSTM(64, return_sequences=False,
         activation='relu', recurrent_dropout=0.2),
    BatchNormalization(),
    Dense(64, activation='relu'),
    BatchNormalization(),
    Dense(32, activation='relu'),
    BatchNormalization(),
    Dense(y.shape[-1], activation='softmax')
])

gestures_model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss='categorical_crossentropy',
    metrics=['categorical_accuracy']
)
```

Snippet 1: Gestures Classifier Architecture.

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Implementation of the RSL to RO translator

4. *ChatGPT* API Service

- Builds a coherent sentence from the given words.
- Passes a list of predicted words to *ChatGPT* along with the conversation history and a system message.
  - The history provides context for better sentence building.
  - The system message explains the task and gives examples for few shot learning.

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Conclusions

1. Main Results

- Can translate:
  - over 4000 words from RO to RSL
  - 60 gestured RSL signs into RO
- Application is performant enough to run on mid-range Android devices.
- Gestures classifier achieves on the **test dataset**:
  - 100% accuracy
  - 99.89% confidence
- Animated gestures were found to be understandable by an interpreter.

2. Future Work

- Use updated *MediaPipe Holistic* version once available.
- Larger, improved and more varied dataset created by recording an expert.
- Implement a sign language segmentation algorithm and not rely on heuristics.

<<<<<

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

Conclusions

3. Summary of contributions

- Created RSL vocabulary from scraped and processed videos.
- Applied Kalman and moving average filters to correct and smooth data.
- Used inverse kinematics to create animations.
- Created a service using NLP and various metrics to match natural speech with vocabulary entries.
- Created a RSL dataset recording pipeline which was used to create the dataset necessary for training a custom gestures classifier.
- Derived representative and robust features as the model's input.
- Developed a service interfacing with ChatGPT to form coherent sentences from predicted words.
- Bundled all functionalities into an Android app for portable, real-time, bidirectional translation.

Thank You! <<<<<