Real-Time Recognition and Translation of Kinyarwanda Sign Language into Kinyarwanda Text

Erick Semindu, Christine Niyizamwiyitira

***Abstract*—Despite significant technological advancements, there continues to be a considerable communication gap between individuals with hearing disabilities and the rest of society. This gap is exacerbated by the fact that the development and research of technologies, such as caption glasses, aimed at bridging this divide, primarily focus on sign languages used in countries with prominent tech industries, including European countries and USA. Consequently, there is a lack of resources and attention devoted to sign language recognition and translation systems for languages spoken in Africa. This research addresses this issue by concentrating on twenty-two common gestures in Kinyarwanda sign language. Through extensive exploration and evaluation of various machine learning algorithms, the study identifies the most effective approach for recognizing and translating these gestures. To validate the effectiveness of the developed system, real-world Kinyarwanda sign language video data is utilized for thorough training and testing. The research successfully culminates in the creation of a functional web application capable of accurately recognizing the 22 Kinyarwanda sign language gestures, both in live video feeds and recorded videos. This achievement represents a significant outcome of the research, as it addresses the specific needs of the Kinyarwanda signing community. By providing a reliable and accessible tool for gesture recognition and translation, the research contributes to narrowing the communication gap between individuals with hearing disabilities who use Kinyarwanda sign language and the wider society.**

***Index Terms*—Computer vision, Kinyarwanda, LSTM, Machine learning, MediaPipe, Reduced Inequalities, Sign Language.**

I. Introduction

ACCORDING to the World Report on Disability, 15% (approximately one billion people) of the world's population have some sort of disability, with ninety-three million being children [1]. According to the National Institute of Statistics of Rwanda's 2022 census report, there are 66,272 individuals with hearing/speaking disabilities in Rwanda, with females accounting for 56% and male accounting for 44% [2]. People with disabilities were historically excluded from many social and economic components of society, until recent decades, when attempts were made to integrate people with disability in numerous sectors of education, health, social, and economic activities [3]. It is rare for people without hearing or speech disabilities to invest in learning sign language. Such trends have facilitated the isolation of people who are deaf and nonspeaking in different social and economic activities in societies.

The field of Information and Communication Technology (ICT) is rapidly advancing, creating opportunities for people to express their creativity, solve problems, and make money [4]. Despite such progress, Africa continues to lag in the correct application of ICT tools for innovation in addressing the problems facing people with disabilities. There is a recent trend in ICT that aids in the inclusion of people with hearing and speech impairments by diminishing the communication bridge. Examples of such innovations are live caption glasses that enable real-time translation of speech into text or the use of object detection and machine learning to translate sign language into text [5] [6]. Most of the research and innovation in this field are invested in American sign language since it has a ready database for its use. Such innovation has been used in African sign languages to a lesser extent, with existing few focusing on South Africa Sign Language (SALS) [7]– [9]. From the Ethnologue, a comprehensive database of world languages, there are over 250 sign languages all over the world [10]. Unlike spoken language, even with countries sharing spoken language most of the time do not share sign language [11]. Although much progress has been done in the field of Sign Language Recognition (SLR), with sophisticated technology in some languages such as British Sign Language and American Sign Language, there is still a long way to go, especially in African countries, which lag in both developing tools for sign language recognition (SLR) and conducting research in the field [11].

The issue at hand extends beyond social implications, encompassing ethical and security considerations. It becomes more evident in sectors such as healthcare and law enforcement, where it often necessitates the involvement of intermediaries to address communication barriers. This reliance on third-party intervention might stem from individuals' hesitance to openly communicate, stemming from feelings of shyness or concerns about the confidentiality of shared information. Moreover, certain professional settings lack the provision of interpreters, placing the onus on individuals with disabilities to independently source such aid. Consequently, this practice impedes their access to essential services, ultimately underscoring the multifaceted challenges arising from this scenario.

This work was supported by Carnegie Mellon University Africa and Uwezo Youth Empowerment.

Erick Semindu is working with Cylab-Africa/Upanzi Network at Carnegie Mellon University Africa, Kigali, Rwanda (email: esemindu@andrew.cmu.edu).

Christine Niyizamwiyitira has over 11 years of teaching and research experience, including leading the ICT in education policy implementation at the Rwanda Basic Education Board. She is currently a Scholar in Residence at Carnegie Mellon University Africa, Kigali, Rwanda (email: cniyizam@andrew.cmu.edu).

In this research project, we intend to produce a functioning model and a web application that will be able to accept video files and detect gestures and actions from Kinyarwanda sign language and translate them into Kinyarwanda sentences. The web application will detect and translate Kinyarwanda sign language gestures in real-time from the uploaded videos, and live camera feed. The web application will also provide a feature to add more labels and gestures for model training purposes.

A. Research Rationale

The interaction between people with and without speech and hearing impairment is inevitable. This is because they share a lot of social services. Apart from educational institution which is a bit tuned to allow people with special needs to have dedicated institution. Most other sectors like health, transportation, and many others are not acquitted of such privileged. The existence of such a system will bridge the communication gap between the deaf community and the rest of society. Furthermore, it will pave the way for the establishment of publicly available databases with Kinyarwanda sign language which will pave the way for the research community to work on the data and develop more sophisticated tools.

B. Aim and Objectives

The aim of the research is to develop a machine learning and computer vision-based system that can accurately recognize and translate Kinyarwanda Sign language gestures into Kinyarwanda text. The study focus will be on exploring various techniques of machine learning to achieve high accuracy in gesture recognition. The system will aim to identify twenty-two common Kinyarwanda gestures.

The research has the following objectives aiming at ensuring the achievement of the main objective.

1) To collect and curate datasets of Kinyarwanda Sign language gestures with labels for training and evaluating the machine learning model.
2) To explore and evaluate different machine learning algorithms and computer vision techniques, such as deep learning, convolutional neural networks, and feature extraction.
3) To optimize the performance of the machine learning model by tuning hyper parameters and selecting the best algorithm and features for Kinyarwanda Sign language recognition.
4) To implement and evaluate the developed system on real-world Kinyarwanda Sign language video data, including evaluating the accuracy of the system

C. Research Questions

1) How can machine learning and computer vision techniques be used to develop an accurate and efficient system for recognizing and translating Kinyarwanda Sign language gestures into Kinyarwanda text?
2) What are the key factors that influence the performance of the system, including the selection of machine learning algorithms, hyper-parameter tuning, feature extraction, and training datasets size?

In an ideal scenario with a single signer, in front of an ideal camera, with an ideal background, and in an ideal lighting condition, we anticipate a successful system to be able to distinguish between 23 Kinyarwanda sign language gestures.

II. Literature Review

There are multiple techniques when it comes to SLR systems. Techniques can be categorized by their data acquisition processes. Some of the common data acquisition techniques are Camera, Data glove, Kinetic, and Leap Motion Controller [11]. SLR techniques can be grouped into two main groups: sensor-based approaches and vision-based approaches [12]. The sensor-based solution necessitates the use of different sensors such as electromyography (EMG) sensors, RGB cameras, Kinetic sensors, leap motion controllers, or their combinations [12]. Vision-based technique recognizes and monitors the signer's hand gestures and facial expressions using image data [12].

In [13], A. Z. Shukor et al. proposed a glove-based system to recognize Malaysian sign language. It used ten tilt sensors for finger flexion and an accelerometer for hand motion. The glove relies on Arduino with a Bluetooth module to connect with Android to display recognized gestures. The project achieved 89% average accuracy; however, it had a limited capacity of micro controller memory, hence only recognizing three letters and three numbers. Kau et al. [14] used flex sensors and gyroscope sensors to develop a glove that recognizes a dynamic gesture in Taiwanese sign language. Flex sensors are used to detect finger flexion and gyroscope sensors are used to determine palm orientation. The combination of the sensors allowed the researcher to achieve an accuracy of 94%. Such a technique, if it was to be adopted, will require industrial-level development as all signs have to be coded into the registers of the boards of the Arduino.

In [15] authors used an electromyogram and a 3D accelerometer for the recognition of 60 Greek sign language gestures. The electromyogram is used to capture hand muscle contraction information while the accelerometer is used to capture the spatial information of the hand. The author managed to acquire 93% accuracy. The technique proposed captures both dynamic and static gestures, however, it also comes with financial expenses to a signer. Sensor-based techniques have proven so far to have far better accuracy than their counterpart vision-based techniques [11]. This is because they are not affected by environmental conditions such as the skin color of the signer, background, and lighting conditions [16]– [18]. The accuracy comes with the cost of the comfort of the signer, financial expenses, reduced naturalness of the interaction, and time consumption in setting up the device [11].

SLR techniques can be used to identify static gestures or dynamic gestures. Static gestures utilize hand gestures and facial expression without movement to convey a message. The dynamic gesture does include movement (especially with hands) to convey a message. [19] and [20] demonstrated the vision-based approach with CNN models to develop a static

sign language recognition system with overall training and validation accuracy of $> 95\%$. Although it has good accuracy, such a system cannot be adopted because Kinyarwanda sign language as the latter is made up of dynamic gestures.

In [21] and [22], the authors demonstrated that the LSTM approach fits best dynamic sign language as gesture recognition is a type of sequential data that LSTM networks are well suited for. The technique can also be used in static sign language as they too can be presented in sequential data.

It is without a doubt that the vision-based approach can efficiently be implemented with ease to implement the SLR with accuracy. Arpita Halder and Akshit Tayade reference [23] showed that media pipe technology can be used efficiently to detect complex hand gestures precisely in American sign language. Hence in this project, we employed the MediaPipe framework as it can detect human body parts accurately. And used the LSTM as it works best with less data.

III. Methodology

This research had separate phases and implementations dedicated in setting up research environment, data collection, exploration of best techniques for feature extraction and model creations, and prototype building.

Data collection

We worked with four people, the first person who can speak and hear and who is expert in Kinyarwanda sign language translation, currently working at Rwanda Broadcasting Agency (RBA) to translate the news. The second person with hearing and speaking impairment works as an education specialist to support the design of the content for learners with hearing and speaking impairment. The third person has speaking and hearing impairment but works as a leader in the union of the deaf and blind. The fourth person is one of the researchers, this person has neither hearing nor speaking impairment. We met these people at the Uwezo[1] Youth Empowerment Center. The participants assisted in providing the most used words in Kinyarwanda so that we could use their translated signs. After agreeing on the twenty-two words, we started to collect the sign gestures from every participant. Every participant sign ten times for every gesture. The data collected were used in both training and testing to produce the best techniques for the sign's translation.

Prototyping

To achieve a functional SLR system, we developed a prototype through distinct phases. In recognition of this, this research is divided into three phases to support process efficiency and promote better outcomes. The first phase, the exploratory phase, involved environmental setup and exploring different tunings for the model, and the Mediapipe library for better accuracy. In the second phase of prototype development, we continued using the better strategy from phase 1 and created a model with 23 Kinyarwanda signs. Online setup for the prototype deployment took place in the third phase.

The HP Pro Book 450 G7 with an Intel(R) Core (TM) i7-10510U CPU running at 1.80 GHz and 16 GB of memory was used for the research experiments. For data collection and live testing, we used both the laptop webcam and a Logitech Webcam C930e external camera throughout the research. The Integrated Development Environment (IDE) used for the programming environment are open-source Jupyter Notebook software and Visual Studio Code. Both the data collection and model training scripts were written in Python using the action detection starter code [24].

We used OpenCV [25] in the project to capture live video from either the external camera or the laptop webcam. Following that, the frame was sent to the MediaPipe function for feature extraction. The extracted features is then saved in local files for the training section and used for model training and testing later, as can be seen in Fig. 1. For Kinyarwanda sign language recognition, the extracted feature will be passed to the model for detection, as it can be seen in Fig. 2

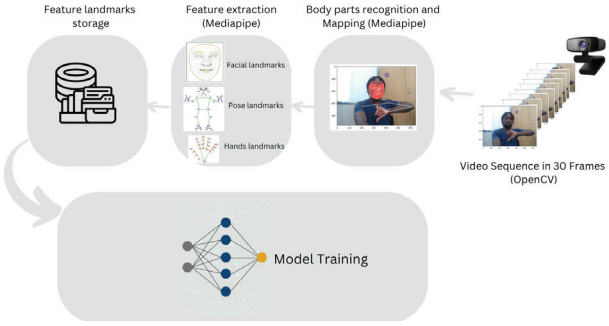

Fig. 1: High-level model training and data collection architecture

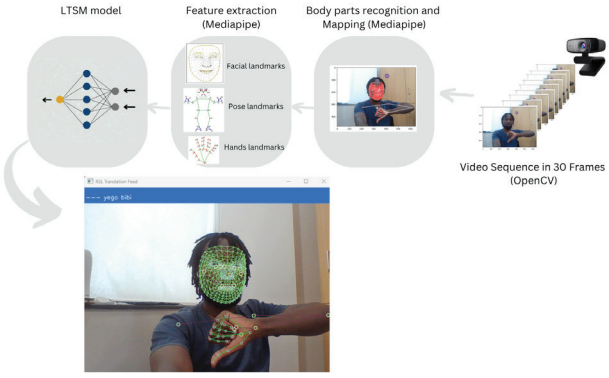

Fig. 2: High-level Kinyarwanda sign recognition architecture

Phase 1: Best techniques for feature extraction

This phase was dedicated to figuring out the best implementation in terms of feature creation and model hyper parameter tanning to yield better results.

Implementation 1: Default Setup

A) Data Input

For this implementation, the main concentration was information on the upper part of the body. MediaPipe library was used for body part identification, determination of the hands' shape, and directions. MediaPipe extracts key points

[1]https://uwezoyouth.org

of the body parts in three dimensions. X, Y, and Z for every frame. In this phase, there are fifty sequences each with thirty frames,each frame with multiple key points for each of the ten labels.

Table I
Sample 10 Labels of Kinyarwanda Sign Language

Label ID	Label	English Translation
1	Yego	Yes
2	Neza	Good
3	Bibi	Bad
4	Urakoze	Thank you
5	Isibo	Sub-village
6	Umurenge	Sector
7	Igihugu	Country
8	Umujyi wa Kigali	Kigali City
9	Kicukiro	Kicukiro District
10	Nyarugenge	Nyarugenge District

Table 1 was extracted from the live video feed sequence of Kinyarwanda's relevant gesture sign for the labels.

B) Extraction of Features

Facial, posing, and hand features were extracted for this implementation. Each landmark point from the MediaPipe library is 3D and has X, Y, and Z coordinates. 468 landmarks can be found on the face Fig. 5. Each hand has twenty-one landmarks Fig. 3. A pose has thirty-three landmarks Fig. 4, but only the upper body, a maximum of twenty-four, was considered the Signer.

$$\text{Total number of hands key points} = (21 \times 3) \times 2 = 126$$
$$\text{Total number of face key points} = 468 \times 3 = 1404$$
$$\text{Total number of posing key points} = 21 \times (3 + 1) = 132$$

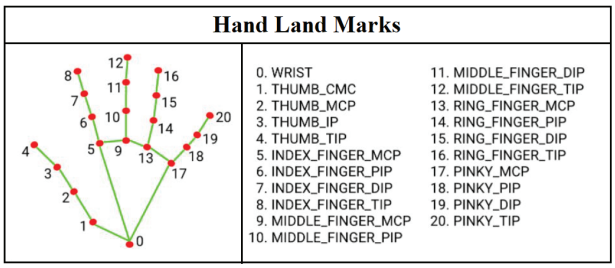

Fig. 3: Hand landmarks [26]

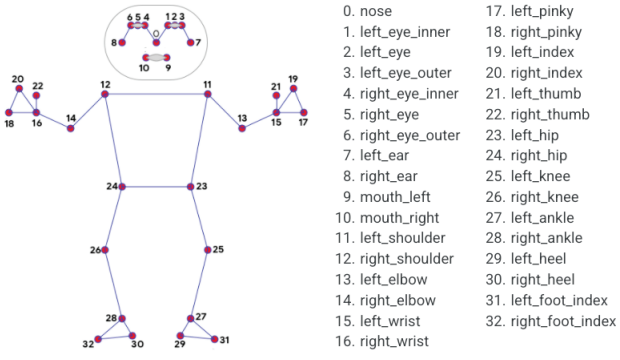

Fig. 4: Pose landmarks [27]

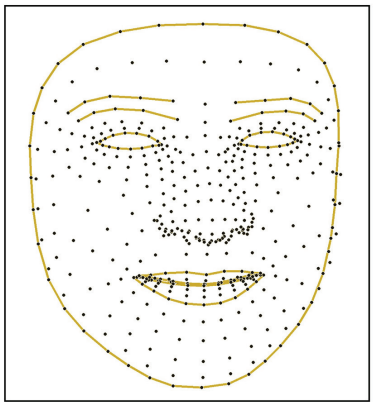

Fig. 5: Facial landmarks [27]

For the pose, the extracted key points also have an extra value for visibility. Hence out of the thirty-three each has four values: 3D coordinates and the visibility value. Eight landmark points for the lower body are ignored for pose detection. From Fig. 4, landmark points 25 to 32 will not have much effect as their values will be filled by zeros. For each frame in this implementation, a total of 1662 landmarks were extracted. That is 126 key points from both hands, 1404 key points from the face, and 132 key points from the pose. For each frame, the extracted face, hands, and pose key points are saved into NumPy arrays, which are later concatenated and saved into a binary file (.npy).

C) Model Creation and Training

Three long short-term memory networks (LSTM) layers and three dense layers of a neural network architecture were constructed for this implementation. 'relu' (A rectified linear unit function that introduces the property of non-linearity to a deep learning model and solves the vanishing gradients issue) activation is present in the top three layers. The output layer had 'softmax' activation. The network had an input shape of (30, 1662), the same size as the concatenated array of extracted features for all thirty steps of the input.

The subsequent LSTM layers in this architecture require sequence input, so the first LSTM layer is configured to return sequences. Because another LSTM layer follows the second LSTM layer, it also returns sequences. However, because it is followed by dense layers, the final LSTM layer does not return sequences.

The dense layers offer a means of mapping the LSTM layer output to the output classes. To add non-linearity to the model and help avoid the vanishing gradient issue, the relu activation function is used. The output layer's softmax activation function offers a probability distribution over the potential output classes, enabling the model to predict. Finally, to prevent over- fitting, we use regularizers l2 (0.01) to specify L2 regularization with a regularization parameter of 0.01 on all layers.

On one hundred epochs, the model was trained using a 0.00001 learning rate. 70% (350 extracted features, thirty-five per label) of the data were used for training, and 30% (150 extracted features, fifteen per label) were used for testing.

D) Implementation Results and Shortcomings

The model performs exceptionally well, with 98% overall categorical accuracy. While doing live testing, the model clearly showed the following shortcoming:

1) Although the model had good accuracy, it produced outputs even when the signer was idle.
2) Multiple output for a single sign language gesture.

Implementation 2: Without facial-landmarks

A) Data Input

In this implementation, data was collected using the same technique used in the previous implementation. A hypothetical label was introduced to train the model to recognize when one is not communicating. The introduced label was '—,' the gesture for the label, the signer is just sitting without any upper body movement. The introduction of this gesture intends to address shortcoming number 1 from the previous implementation.

B) Extraction of Features

This implementation used the same feature extraction method as implementation one. However, the landmarks on the face were not included in this implementation. 468 key points are provided by MediaPipe from the face. We found that there are not many differences between the values of the key points for the facial landmarks for the ten gestures. Since they made up more than 84%, facial landmarks had a significant impact on the trained model. The similarity of the gestures is increased by this influence. The removal of facial landmarks from the input data intends to address the shortcoming number 2: multiple outcomes for one gesture.

The head position and movement will still be captured as there are ten points from the pose features that map the head. The total extracted features per frame in this implementation were 258, thus 132 pose key points, and 126 hands key points.

C) Model Creation and Training

In this implementation, the same model setup technique as in the first implementation was used with differences in epoch number and network input. In this implementation, the model was trained using 150 epochs, with the purpose to get as much accuracy as possible. The model input size was (30, 258). The size changed to reflect the size of the extracted features as the facial landmarks are excluded for every frame the size of the concatenated array was (30, 258).

D) Implementation Results and Shortcomings

The categorical accuracy of the trained model was 99%. Additionally, the first implementation flaw was resolved. The model would output a "—" label to show that the signer is not signing when they are in their idle position. Although this implementation worked well for all eleven gestures, it still had a flaw.

Most of the time, the model was unable to distinguish between signs with subtle differences, such as a hand's orientation or a difference in one finger. Such gestures include "Neza", "Bibi." "Neza," which means "good" involves making a closed fist with the thumb raised and moving the hands up and down. "Bibi" means bad. The same hand structure is used in the gesture, with the thumb finger pointing down and the first turning vertically. Another gesture involving a closed fist was called "Yego", which means yes.

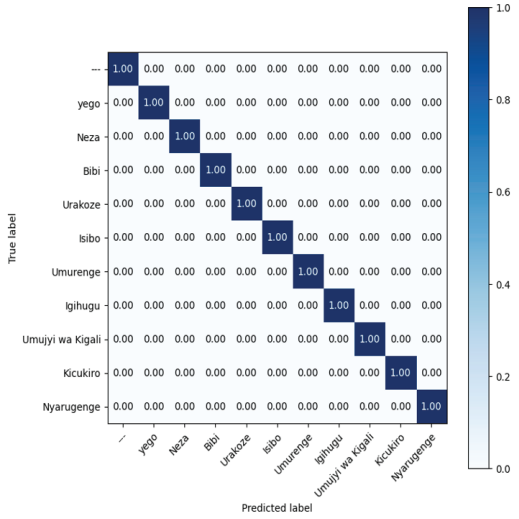

Fig. 6: Confusion Matrix heat-map for the 11 Labels for the third implementation.

Implementation 3: More weight on hands key points

A) Data Input

For this implementation data was collected in the same style as from the second implementation. Eleven labels/gestures were involved. For each label, fifty videos, each with thirty frames, were recorded.

B) Extraction of Features

Like the earlier implementation, each frame had hands and pose landmarks extracted from it. Each frame's extracted features from both hands were entered ten times in the concatenated array of extracted features to magnify the weight of the hand's key points. This magnification intends to overcome the shortcoming from the previous implementation of the model failing to distinguish signs with almost similar hand structure.

$$\text{ExHf} = (33 \times (3 + 1)) + (21 \times 2 \times 3 \times 10) = 1392$$

where:

- **ExHf**: Extracted hands features

A thumb is mapped with five key points, as seen in Fig. 3. Given that each point is in three dimensions, a thumb can be represented by a total of fifteen values. The learning of the model is hardly affected by the slight difference in these fifteen points out of 258 (from the second implementation). A change to the thumb or any other finger will have an impact on 150 values due to the hand's key points being magnified.

C) Model Creation and Training

The model setup method used in this implementation was the same as in the second implementation but with a different epoch number and network input size. In this implementation, the model was trained over 180 epochs to achieve the highest level of accuracy. The input size for the model was (30, 1392). As the facial landmarks were removed and hand landmarks multiplied ten times, the size changed to reflect the size of the extracted features; the concatenated array's size was (30, 1392).

D) Implementation Results and Shortcomings

The trained model had a categorical accuracy of 100%. When it comes to distinguishing between similar signs, the model performed exceptionally well.

Phase 2: Prototype Model Development Phase

The third implementation from the exploratory stage served as the foundation for this stage. The main objective for this phase was to efficiently implement and collect data by involving experts to develop a model with more signs.

A) Data Input

For this phase, the datasets were gathered using four signers. There were twenty-two signs in Kinyarwanda altogether. The same environmental setup was used to collect fifty video clips for each of the twenty-two labels (the same laptop and webcam). Data collection was made possible by skilled Kinyarwanda sign language users. During this stage, fifty sequences are collected for each of the twenty-two labels' key points in thirty frames. The gesture signs for the labels in Table 2 will be taken from a live video feed sequence of Kinyarwanda. The videos were shot outside in bright light, with a signer against a white background. The first and second signers contributed fifteen video sequences each, for each label. The third and fourth signers contributed ten videos for each, for each label.

Table II
22 SAMPLE LABELS OF KINYARWANDA SIGN LANGUAGE

Label ID	Label	English Translation
0	—	No Sign
1	Yego	Yes
2	Neza	Good
3	Bibi	Bad
4	Urakoze	Thank you
5	Isibo	Sub-village
6	Umurenge	Sector
7	Igihugu	Country
8	Umujyi wa Kigali	Kigali City
9	Kicukiro	Kicukiro District
10	Nyarugenge	Nyarugenge District
11	Akarere	District
12	Muraho	Hello
13	Amakuru	How are you
14	Imodoka	Vihicle
15	Moto	Motorcyle
16	Akazi	Work
17	Bayi	Goodbye
18	Mwaramutse	Good morning
19	Papa	Father
20	Mama	Mother
21	Oya	No
22	Imyaka	Age

B) Extraction of Features

Each frame had hands and pose landmarks extracted from it like the implementation three outperforms others from the earlier phase. To emphasize the importance of the hand's key points, the extracted features from each frame from both hands were entered eleven times in the concatenated array of extracted features.

$$\text{ExHf} = (33 \times (3+1)) + (21 \times 2 \times 3 \times 11) = 1518$$

where:

- **ExHf**: Extracted hands features

C) Model Creation and Training

This implementation followed the same model setup procedure as the third implementation but with a different epoch number, network input size, and layer count. An additional LSTM with 128 units and relu activation was added for this model. For the model to be as accurate as possible, 250 epochs of training were completed. The model's input size was (30, 1518). The size changed to reflect the size of the extracted features as the facial landmarks were eliminated and the hand landmarks multiplied by eleven; the concatenated array's size was (30, 1518).

Fig. 7: Kinyarwanda Sign Language Expert participating in the data collection.

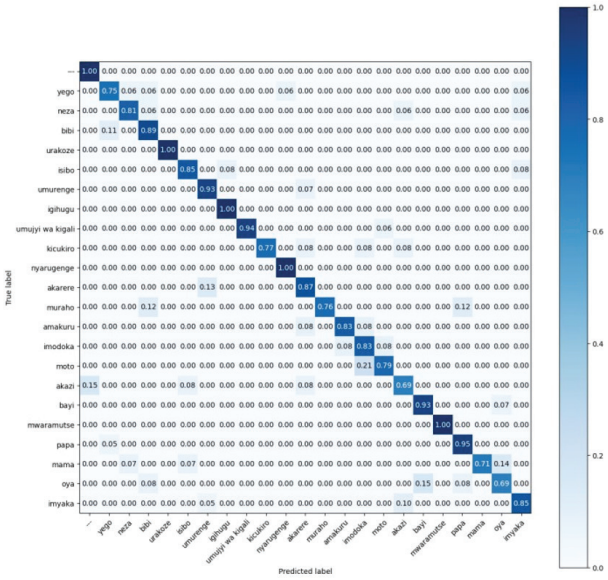

Fig. 8: Confusion Matrix heat-map for the 22 Labels on the second phase.

D) Implementation Results and Shortcomings

The trained model had an accuracy of eighty-seven accuracies. Some signs doing very well and others doing exceptionally poorly as can be seen in Fig. 8. Akazi and Oya labels had the lowest accuracy of 69%. Urakoze, Igihugu, and Kicukiro are some of the signs with 100% categorical accuracy. The difference is influenced by the complexity of the hand gesture (signs like Kicukiro involve finger movement in a squeezing-like pattern). Other reasons might be influenced by the data collection. This stage involved individuals who were not familiar with the timing and transitions between different signs during the data collection process. This also influenced the poor result on some signs

Phase 3: Prototype Development Phase

This stage was dedicated to the development of the prototype: a web application integrated with the LSTM model developed and trained from the previous phase. The web application was designed to serve three main functionalities aligning with the project objectives:

1) Translation of Kinyarwanda Sign Language from a live video feed.
2) Translation of Kinyarwanda Sign Language from a recorded video feed.
3) A label and Video contribution functionality, for voluntary collection of model training data.

The web application is built on top of a Flask framework. Flask is a lightweight, Web Server Gateway Interface (WSGI) web application framework written in Python. The decision for the selection of the Flask platform is based on its easy integration with Python codes as the detection and extraction of feature processes were written in Python. The Website was built through the utilization of the Flask dashboard template [29]. The template was customized and improved to enable the ability to perform the three-functionality mentioned earlier.

The web app was first integrated with essential functions to enable it to detect and translate Kinyarwanda Sign Language. The functions included the importation of the trained model to the app, drawing of the landmarks, and extraction of the key points. Then the application setup focused on three pages each dedicated to one of the main objectives of the web application. The following paragraphs explain how the application was set up to enable it to perform its main three objectives.

A) Translation for the live video feed.

This page was responsible for the translation of the video feed directly captured from the camera device. This page had mainly four essential elements (1) a button (To toggle the start of the translation and the stop of the translation), (2) a video displaying element for displaying what the camera is currently capturing), (3) an element to display the translated text, and (4) a button to clear the translated text.

Once the Translation button is clicked, it calls a function that is responsible for initiating the camera, it triggers the display of the video displaying element, also calls the function responsible for the translation. The translation function calls several other functions responsible for drawing the landmarks and extraction of the key points. The translation function translates the gesture and adds the signed label to the sentence variable.

The translation page checks and fetches the value of the sentence once the translation begins and displays it on the page. The clear button clears the content of the sentence variable and updates the sentence-displaying element on the page.

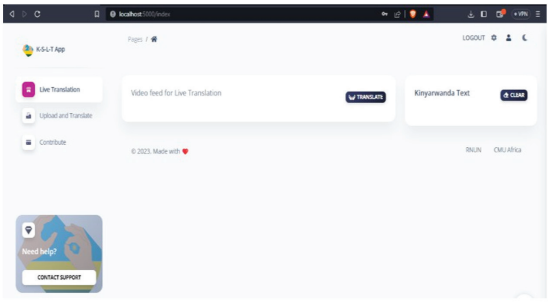

Fig. 9: Live Translation Page (not translating)

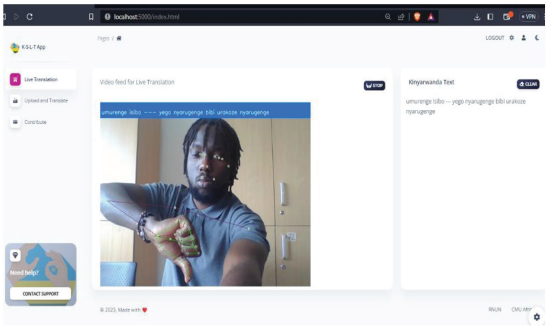

Fig. 10: Live Translation Page(translating)

B) Translation for the recorded video feed.

The second page of the web application is responsible for the translation of the recorded video. The video to be translated is first uploaded to the application and then translated. The translation begins automatically upon successfully uploading the video. The page has a form element to upload the video which is a file input element. Other elements on the page are the element to display the video, the element to display the translated text, and a button to clear the translated text.

Once the video has been selected, the clicking of the upload button will set in motion the series of the following processes. The selected video will be uploaded and saved in the application, and the name for the uploaded video will be returned and passed to the translate function for the recorded video function. The translate function will fetch the video and set it as the source of frame input (not the camera as it is in the translate function for the live feed). The translate function will process the video and set the output in the sentence variable. Like the live feed translation page, the page checks and fetches the value of the sentence once the translation begins and displays it on the page. The clear button clears the content of the sentence variable and updates the sentence-displaying element on the page.

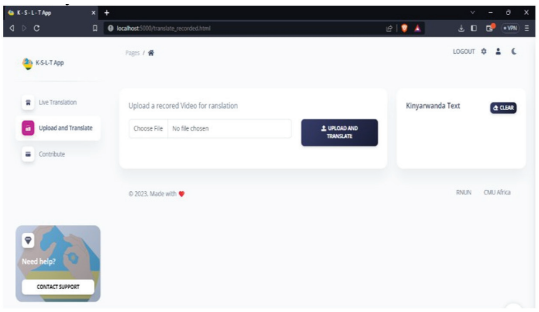

Fig. 11: Recorded Translation Page (not translating)

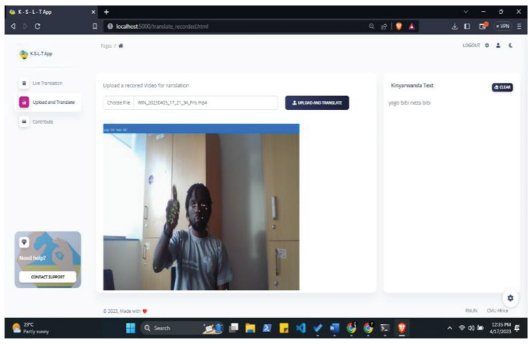

Fig. 12: Recorded Translation Page(translating)

C) Contribute (Uploading video and label)

This is the third page of the web application It serves the purpose of voluntary collection of data (Video and Labels) for the model training purpose. The page has a form with file and text input fields. The form has a validation mechanism that checks if either the text (label) or the file (video) field is empty and returns the appropriate error. The forms validation mechanism also checks the sign about to be contributed if they are already included in the trained model. If so, it returns an error advising to contribute another sign that is not yet familiar to the system.

IV. Results and Discussion

This research was successful in developing a working web application that can distinguish 22 Kinyarwanda sign languages from live and recorded video feeds. The following are the key results we found through this research:

Through our study, it emerged that OpenCV and MediaPipe LSTM models can effectively serve as the foundation for developing a robust Kinyarwanda Sign Language recognition system. Notably, while OpenCV's usage can be tailored to accommodate alternate image input libraries based on specific model requirements, MediaPipe presents a potent solution by enabling the exclusion of facial data and the focused utilization of pose and hand landmarks.

In the context of implementing a user-friendly web application, Flask demonstrated its prowess consistently throughout the research, exhibiting seamless efficiency without encountering any significant issues.

The following factors emerged as pivotal influencers on the system's performance and accuracy in translating gestures:

For MediaPipe, in normal settings, when a full body is being mapped, face landmarks assume 84% of the extracted features. Although Kinyarwanda Sign Language does include facial expressions, a substantial percentage of such information overshadows other key information such as hand and pose landmarks. As for the twenty-two signs analyzed in this research, the facial expression did not have that much difference. The similarity in the facial landmarks negatively influenced the model to confuse the signs. The system provided much better accuracy when the facial data were excluded.

In the resting pose of a problem (before and/or after the sign), signs start and/or end in a pose where the signer is resting (not signing). The resting pose does increase the similarity between signs and leads to the problem that when a person is not signing, the model still gives multiple outputs of signs. Through this research, we addressed this problem by registering a fictitious label '—-' on the model to represent when a person is not signing. The introduction of such a sign improved the perdition by eradicating the false positive when a signer is not signing.

One major problem in SLR is the differentiation of signs with high similarity. In the twenty-two signs we analyzed in this research, we concluded that the magnification of the extracted feature could magnify the difference between the signs. As elaborated in the third implementation of the first phase. We magnify the weight of the finger data by incorporating extracted hand features ten and then eleven times. The technique improved the accuracy of the model by enabling it to easily distinguish between similar signs like Neza and Bibi. It is also essential to use a better camera for better results, not only when collecting data but also when using sign language recognition systems.

We faced the following challenges when developing the Kinyarwanda Sign Language recognition and translation system:

Variation of the time for a sign, gestures do differ in the duration of signing. This poses a challenge and makes it hard to distinguish the beginning and the end of a sign. As in this implementation, the model does consider a particular number of frames to make a prediction. In this research, we implemented thirty frames in one second for all twenty-two signs. The different gesture signing periods pose a challenge in prediction and reduce accuracy for the model. Further research can be done, and building systems that run multiple models that take into consideration the different number of frames might be a probable solution to such a problem.

Moreover, the lack of a large data set significantly hinders the full utilization of existing artificial neural network-based algorithms. Research in such a field and having a publicly available data set can solve the problem.

The course of this research project gave us exposure not only to the solution domain for Kinyarwanda Sign Language but also to the problem domain for the topic. Through the series of interactions with some of the people with hearing and speaking disabilities, we become exposed to other challenges that demand the existence of such a system. Some of the most important social services, like police officers or hospitals, rely on a third person to interpret or aid in communication between

people with hearing and speech disabilities. Such a trend poses security and privacy concerns for people with disabilities. In a setting like a hospital, deaf or mute patients might not feel comfortable revealing their health condition to a third person, sometimes leading to them providing false information. The same situation also occurs in the crime investigation process.

To support our findings and ensure reproducibility, the dataset used in this study has been made publicly available. You can access the dataset and its accompanying metadata at Mendeley Data [30].

V. Conclusion

To translate Kinyarwanda Sign Language into Rwandan text in a user-friendly web application, we recommend using the LSTM model, MediaPipe, and Flask framework as demonstrated in this research. Through our experiment, we showed how MediaPipe could be used to collect data and how to properly process the features that were extracted. More specifically, we added more hand landmarks to the extracted feature by incorporating the extracted hand landmarks' key points eleven times. We were able to distinguish between similar gestures by enlarging the minor changes, particularly in hands, thanks to our improved feature extraction technique. Another key finding, we found in the development process, it is crucial to introduce the fictitious gesture to show when the signer is not signing.

Because this research was conducted with limited resources and time, the final model produced in the final phase of the prototype did not have high accuracy. Participants in the data collection procedure were not fully familiar with the system data collection technique due to time constraints, thus, they struggled with the process. They had to catch up quickly (within 4 hours), which harmed the collection of the data set for the model that was used for the final web application. We strongly believe the technique we present can be used in the future, and with a big enough dataset provided by experts in both the system and Kinyarwanda sign language, it will produce much better results than it can in the third implementation of phase one in the methodology section.

We integrated the trained model with the Flask web application, which we developed efficiently and cost-effectively. As mentioned in the results and discussion section, we fervently support further study and application of the technique because it will benefit those with disabilities by addressing several of their issues.

References

[1] World Health Organization, "World report on disability. World Health Organization", 2011

[2] National Institute of Statistics of Rwanda, "RPHC5 Thematic Report: Socio-economic status of Youth from fifth population and housing census,2022 [Online]. Available: https://www.statistics.gov.rw/file/14204/download?token=B7uvDxmj

[3] Disability Rights Task Force, "From Exclusion to Inclusion: A Report of the Disability Rights Task Force on Civil Rights for Disabled People Exclusion Inclusion," UK, 1999.

[4] B. M. Leiner et al., "A brief history of the Internet," ACM SIGCOMM Computer Communication Review, vol. 39, no. 5, pp. 22–31, 2009.

[5] Evanne Yu, "Live-caption glasses let deaf people read conversations using augmented reality — South China Morning Post," Jan. 30, 2022. https://www.scmp.com/video/technology/3187127/live- caption-glasses-let-deaf-people-read-conversations-using- augmented

[6] Z. Zhou et al., "Sign-to-speech translation using machine- learning-assisted stretchable sensor arrays," Nat Electron, vol. 3, no. 9, pp. 571–578, 2020.

[7] I. Achmed, "UNIVERSITY OF THE WESTERN CAPE Upper Body Pose Recognition and Estimation towards the Translation of South African Sign Language," 2011.

[8] W. Nel, M. Ghaziasgar, and J. Connan, "An integrated sign language recognition system," in ACM International Conference Proceeding Series, 2013, pp. 179–185. doi: 10.1145/2513456.2513491.

[9] W. Ndlovu and M. Alatise, "Application of Artificial Neural Network on South African Sign Language Recognition System," in 5th International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, icABCD 2022 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/icABCD54961.2022.9856317.

[10] "Ethnologue — Languages of the world." https://www.ethnologue.com/ (accessed Mar. 06, 2023).

[11] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis," Intelligent Systems with Applications, vol. 12, p. 56, 2021, doi: 10.1016/j.iswa.2021.20.

[12] M. Kumar, "Conversion of Sign Language into Text," International Journal of Applied Engineering Research, vol. 13, no. 9, pp. 7154–7161, 2018, [Online]. Available: http://www.ripublication.com

[13] A. Z. Shukor, M. F. Miskon, M. H. Jamaluddin, F. bin Ali Ibrahim, M. F. Asyraf, and M. B. bin Bahar, "A New Data Glove Approach for Malaysian Sign Language Detection," Procedia Comput Sci, vol. 76, pp. 60–67, Jan. 2015, doi: 10.1016/J.PROCS.2015.12.276.

[14] L. J. Kau, W. L. Su, P. J. Yu, and S. J. Wei, "A real-time portable sign language translation system," Midwest Symposium on Circuits and Systems, vol. 2015-September, Sep. 2015, doi: 10.1109/MWSCAS.2015.7282137.

[15] V. E. Kosmidou and L. J. Hadjileontiadis, "Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data," IEEE Trans Biomed Eng, vol. 56, no. 12, pp. 2879–2890, Dec. 2009, doi: 10.1109/TBME.2009.2013200.

[16] R. Singh and S. Choudhury, "Advances in Intelligent Systems and Computing 479 Proceeding of International Conference on Intelligent Communication, Control and Devices." [Online]. Available: http://www.springer.com/series/11156

[17] N. Pugeault and R. Bowden, "Spelling It Out: Real-Time ASL Fingerspelling Recognition." [Online]. Available: http://personal.ee.surrey.ac.uk/Personal/N.Pugeaulthttp://personal.ee.surrey.ac.uk/Personal/R.Bowden

[18] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," IEEE Trans Cybern, vol. 43, no. 5, pp. 1318–1334, Oct. 2013, doi: 10.1109/TCYB.2013.2265378.

[19] L. K. S. Tolentino, R. O. Serfa Juan, A. C. Thio-ac, M. A. B. Pamahoy, J. R. R. Forteza, and X. J. O. Garcia, "Static sign language recognition using deep learning," Int J Mach Learn Comput, vol. 9, no. 6, pp. 821–827, 2019, doi: 10.18178/ijmlc.2019.9.6.879.

[20] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," Neural Comput Appl, vol. 32, no. 12, pp. 7957–7968, Jun. 2020, doi: 10.1007/s00521-019-04691-y.

[21] L. Zhang, G. Zhu, P. Shen, J. Song, S. Afaq Shah, and M. Bennamoun, "Learning Spatiotemporal Features using 3DCNN and Convolutional LSTM for Gesture Recognition."

[22] M. Ur Rehman et al., "Dynamic hand gesture recognition using 3D-CNN and LSTM networks," Computers, Materials and Continua, vol. 70, no. 3, pp. 4675–4690, 2022, doi: 10.32604/cmc.2022.019586.

[23] A. Halder and A. Tayade, "Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning," 2021. [Online]. Available: www.ijrpr.com

[24] Renotte Nicholas,"GitHub -nicknochnack/ActionDetectionforSignLanguage: A practical implementation of sign language estimation using an LSTM NN built on TF Keras.," Jun. 19, 2021. https://github.com/nicknochnack/ActionDetectionforSignLa nguage (accessed Mar. 08, 2023).

[25] "Home - OpenCV." https://opencv.org/ (accessed Apr. 26, 2023).

[26] venugopalkadamba, "Face and Hand Landmarks Detection using Python - Mediapipe, OpenCV - GeeksforGeeks," 2023. https://www.geeksforgeeks.org/face-and-hand-landmarks- detection-using-python-mediapipe-opencv/ (accessed Mar. 08, 2023).

[27] "Pose — mediapipe." https://google.github.io/mediapipe/solutions/pose.html (accessed Mar. 14, 2023).
[28] G. H. Samaan et al., "MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition," Electronics (Switzerland), vol. 11, no. 19, Oct. 2022, doi: 10.3390/ELECTRONICS11193228.
[29] "Soft UI Dashboard - Open-Source Flask Starter — AppSeed." https://appseed.us/product/soft-ui-dashboard/flask/ (accessed Apr. 14, 2023)
[30] Semindu, Erick; Niyizamwiyitira, Christine (2024), "Real-Time Recognition and Translation of Kinyarwanda Sign Language into Kinyarwanda Text (2023) Mediapipe NumPy array Hands and Pose extracted key points for 22 Sign Language", Mendeley Data, V1, doi: 10.17632/p6zc5g9bdy.1

Erick Semindu has a solid foundation in computer science and information technology, specializing in cybersecurity. He earned his Master of Science in Information Technology with a Cyber Security Concentration from Carnegie Mellon University Africa (CMU-Africa) and his Bachelor of Science in Computer Science from Ruaha Catholic University (RUCU). Currently, he is a Research Associate at CMU Africa, working with Cylab Africa on projects related to internet measurement and cybersecurity education.

With a passion for teaching and research, he has served as a Teaching Assistant at both undergraduate and graduate levels. He is dedicated to promoting inclusivity and representation, particularly for individuals with disabilities in Africa, and actively advocates for ICT for All. His research interests include Cyber Security, Augmented Reality (AR), Virtual Reality (VR), and Assistive Technology. His academic journey has been driven by a commitment to excellence, and he continues to champion diversity and innovation in the tech industry.

Christine Niyizamwiyitira has more than 13 years of teaching and research experience, including as a lecturer at the University of Rwanda. She has also led the Department of ICT in Education at Rwanda Basic Education Board for nearly five years. Niyizamwiyitira led the ICT in education policy implementation in alignment with the competency-based curriculum for basic education. She led the ICT technical working group for ICT in Education, which brings together the government and education partners towards ICT in education.

Niyizamwiyitira has a Ph.D. in computer systems engineering from the Blekinge Institute of Technology in Sweden, as well as a master's in telecommunication engineering and a bachelor's in computer science from Korea University of Technology and Education in South Korea and the National University of Rwanda, respectively. She has research interests in education technology, data science, and emerging technologies.

CONTENTS PAGE