Mexican Sign Language recognition using hand-landmarking and a KNN algorithm

Diego Santa Cruz[1]

[1]Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO), Guadalajara, Mexico

Corresponding author: Diego Santa Cruz (e-mail: adrian.santacruz@iteso.mx).

ABSTRACT The purpose of this project is to prepare data for a future project centered on developing a Mexican Sign Language classifier. We obtain 3-dimensional coordinates for hand landmarks from images of Mexican Sign Language alphanumerical signs obtained using Google's MediaPipe module. Following this, the data is transformed to obtain wrist-centered, hand-orientation-independent coordinates of the hand landmarks. A statistical analysis is performed on this data to evaluate it's quality. Finally, we make use the K Nearest Neighbors algorithm to evaluate both the data's usability as well as this algorithms applicability to our data.

INDEX TERMS Hand landmarking, K Nearest Neighbors, Sign Language Recognition

I. INTRODUCTION

Around 300,000 people use Mexican Sign Language (MSL) as a means of communication. This represents around 13% of the hard-of-hearing population in Mexico and 0.2% of the general Mexican population. These low numbers illustrate the isolation that hard-of-hearing people live in. In part, this can be explained by the low awareness and interest in this language and the lack of access to its teaching. It is therefore important to find new ways to teach MSL to both speaking and non-speaking people in Mexico that are easy to access.

Worldwide, a popular solution to address this problem is the development of Sign Language Recognition Systems [1]. Among these, a few have focused on MSL [2], [3]. We are developing this project in the interest of contributing to this research.

II. DATA ACQUISITION AND PREPROCESSING

A. SOURCE DATA ACQUISITION

We will be using the "Mexican Sign Language's Dactylology and Ten First Numbers - Labeled images and videos" dataset ***[4] [5]***.

It is a set of videos where 11 different volunteers perform 39 alphanumerical MSL signs. Specifically, this includes the signs corresponding to the 29 letters of the MSL alphabet (which include the 26 contained in the English alphabet as well as "LL", "RR" and "Ñ") and the signs for the numbers 1 through 10. Each sign is performed by each volunteer 10 times, 5 times with each hand. This means the dataset contains around 4290 videos in which an isolated sign is performed.

The background in the video as well as the volunteers' clothing and their distance to the camera vary, which should aid in making the model trained on this data more robust. Some of the signs are static (meaning there is no movement involved in the action) and some are dynamic (meaning movement is part of the sign).

The videos are filed in the following manner:

- First, per volunteer, for example "person number 9"
- Second, per number repetition and dominant hand used for the signing, for example "Left Hand repetition number 2" or "Right hand repetition number 5"
- Finally, per sign being performed, for example "letter C"

B. DATA PREPROCESSING

1) FRAME EXTRACTION

We have around 110 videos for each of the classes. Considering this and after analyzing the videos in the source data set, we decided to extract the last 15 frames in from each of the videos using the OpenCV python library [6].

2) HAND LANDMARK EXTRACTION

For each of the extracted frames, we use Google's Mediapipe hand-landmarking model [7], [8] to extract 3-dimensional

coordinates for 21 specific points of the hand which can be seen in figure 1. It is worth noting that up to here we are following the same steps used in [3], the paper which was originally developed to make use of our datasets [4] [5].

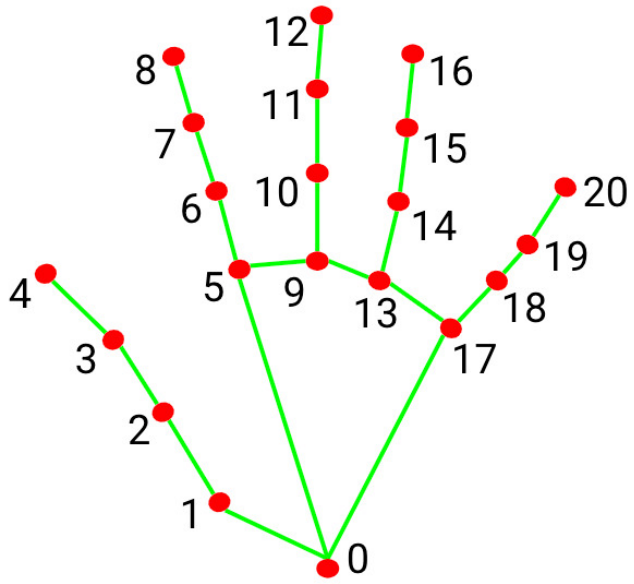

Figure 1:
Hand Landmarks Identified by Google's MediaPipe module.

3) DATA TRANSFORMATION

In the interest of obtaining more meaningful data, the idea for this project was to perform a series of transformations on our 21 coordinate triplets. We obtain data corresponding to the following:

- The center of gravity of the hand
- The vector normal to a plane which approximates the palm of the hand
    - To do so, we first get the coordinates of two vectors which belong to the plane that approximates the palm
        - We choose v1 and v2 as in Figure 2, where A is the point between points #5 and #9, and B is the point between points #13 and #19
    - Then we obtain the vector normal to the plane by performing the cross product of v1 and v2
        - It is worth noting that we decided that we always v3 to "point out of the front" of the hand. Because of this, the order of v1 and v2 will depend on whether we are using the left or right hand.
        - Fortunately, the videos included labels for the hand which is performing the sign, which we can inherit, and MediaPipe's hand landmarker also recognizes handedness. Therefore, we are always able to order v1 and v2 in the desired way in the cross product.
- For each of the 21 hand landmarks, we first transform their coordenates so that they are centered on the wrist (point #0)
    - This is so that regardless of where the camera might have been placed with relation to the signer or other such factors which have no impact on the sign, the coordinates will provide meaningful information
    - Remember that we also obtained the coordinates for the center of gravity of the hand which we will be keeping, so we don't completely lose track of this information
- Then we apply a change a base from the natural base captured by the camera and then deduced by Mediapipe, to the base (v1, v2, v3).
    - This makes the coordinate be more "hand-centric" and not based on arbitrary decisions of camera placement.
    - Also, small differences with regards to the angle in which the palm is pointing which often have no meaning should be less impactful in this new base
    - Remember that we also obtained the coordinates for the vector "pointing out of the palm" which we will be keeping, so we don't completely lose track of this information

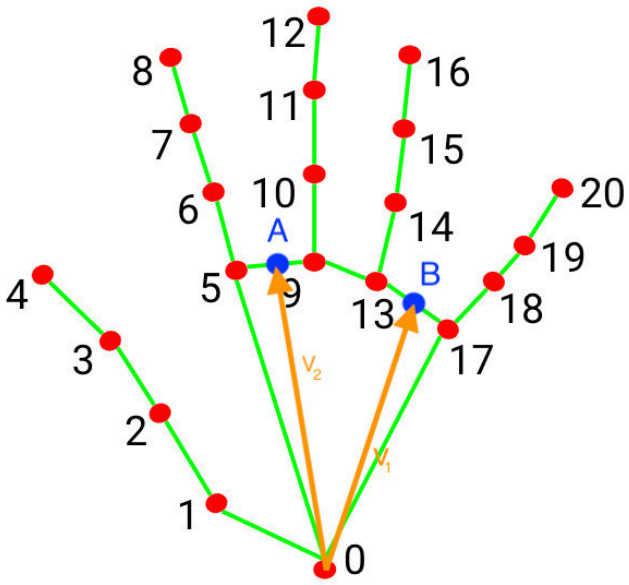

Figure 2:
Vectors v1 and v2 for the new base

After these transformations, we have the following information about each of our frames:

i. 63 columns with the coordinates of each landmark with the wrist as the origin of the frame of reference and (v", v#, v!) as the base. This gives us information about the "hand configuration" in a position-independent and orientation-independent way. "Hand configuration" refers to the shape the hand takes for the sign. It is one of the main factors in Mexican Sign Language.
ii. 3 columns with coordinates of the vector normal to the palm of the hand.
iii. 3 columns with the coordinates of the barycenter with the bottom left corner of the image as the frame of reference and 3 columns with the coordinates of the vector normal to a plane associated to the palm in the original base. This gives information about the position and orientation of the hand and will be important for dynamic signs.
iv. 1 column which indicates the person signing.
v. 1 column which indicates the number of the cycle of repetition.
vi. 1 column which indicates the hand with which the sign is performed with. It is worth mentioning that all the signs in our dataset are performed with one hand, but there are many signs in MSL which use both hands simultaneously. This would present a greater challenge. Nevertheless, MediaPipe's framework can identify landmarks on multiple hands simultaneously, so it is theoretically possible to extend this system to that scenario.
vii. 1 column which indicates the class
viii. 1 column which indicates the class in an "integer" format.

We use the Apache Spark framework [9] to perform these transformations in a parallelized way.

4) DIMENSIONALITY REDUCTION

We will use the Scikit-learn implementation of the Principal Component Analysis (PCA) [10] algorithm to reduce the dimensionality of our dataset.
The number of components we will keep after using PCA is ## (@profe: tengo que escoger esto todavía)

C. STATISTICAL ANALYSIS ON THE QUALITY OF THE DATA

@Profe:
Voy a hacer un análisis de correlación así como la otra métrica de la cual ahorita no recuerdo el nombre que mencionó en clase que es mejor que la correlación
Voy a hacer el análisis para los features antes de utilizar PCA y también para los componentes después de usar PCA.

III. K NEAREST NEIGBORS MODEL

A. TRAINING AND HYPERPARAMETER TUNING

We use the scikitlearn implementation K Nearest Neighbors algorithm [11] to classify our data.

We chose this model because it allowed us to focus on the data preparation transformations and spend less time on tuning hyperparameters. In fact the only hyperparameter we had to chose was k.

@Profe:
aquí voy a meter una grafica intentando muchos valores de k también podría tratar de usar el gridsearch y el k-folds en esta sección

After analyzing, we chose ## as the value for k

B. EVALUATION AND METRICS

As this a classification problem, we chose the following metrics: Accuracy, Recall, Precision and F1 Score.

Here are the results we obtained for our model:
@ Profe aquí voy a meter una tablita con los resultados

IV. REFERENCES

[1] N. Mohamed, M. B. Mustafa and N. Jomhari, "A Review of the Hand Gesture Recognition System: Current Progress and Future Directions," *IEEE Access,* 2021.

[2] G. Garcia-Bautista, F. Trujillo-Romero and S. O. Caballero-Morales, "Mexican Sign Language Recognition Using Kinect and Data Time Warping Algorithm," *IEEE Xplore,* 2017.

[3] M. E. Trejo Rodriguez, Modelos computacionales aplicados a la Lengua de Señas Mexicana, 2019.

[4] M. E. Trejo Rodriguez, O. Oubram, B. Ali and N. Lakouari, "Mexican Sign Language's Dactylology and Ten First Numbers - Labeled images and videos. From person #1 to #5," 30 May 2023. [Online]. Available: https://data.mendeley.com/datasets/67htnzmwbb/1. [Accessed 9 April 2024].

[5] M. E. Trejo Rodriguez, O. Oubram, B. Ali and N. Lakouari, "Mexican Sign Language's Dactylology and Ten First Numbers - Labeled images and videos. From person #6 to #11," 30 May 2023.

[Online]. Available: https://data.mendeley.com/datasets/67htnzmwbb/1.. [Accessed 9 April 2024].

[6] OpenCV, "OpenCV - Open Computer Vision Library," [Online]. Available: https://opencv.org. [Accessed 05 11 2024].

[7] Google, "Mediapipe," [Online]. Available: https://developers.google.com/mediapipe.. [Accessed 9 April 2024].

[8] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," Google Research, 2020.

[9] Apache Spark Foundation, "Apache Spark™ - Unified Engine for large-scale data analytics," [Online]. Available: https://spark.apache.org. [Accessed 05 11 2024].

[10] Scikit-Learn, "PCA -- scikit-learn 1.6.dev0 documentation," [Online]. Available: https://scikit-learn.org/dev/modules/generated/sklearn.decomposition.PCA.html. [Accessed 5 Nov 2024].

[11] Scikit-Learn, "KNeighborsClassifier -- scikit-learn 1.6.dev0 documentation," [Online]. Available: https://scikit-learn.org/dev/modules/generated/sklearn.neighbors.KNeighborsClassifier.html. [Accessed 05 Nov 2024].