PSLT - Portuguese Sign Language Translation

João Amoroso
Instituto Superior Técnico
Lisbon, Portugal
joao.amoroso@tecnico.ulisboa.pt

João Teixeira
Instituto Superior Técnico
Lisbon, Portugal
joao.n.teixeira@tecnico.ulisboa.pt

***Abstract*—Machine learning and Artificial Intelligence (AI) have become integral parts of our daily lives, providing assistance in various domains, including speech-related tasks. While there are algorithms designed to translate between different spoken languages, there are fewer algorithms available for translating sign language into written text. This paper aims to present a model we have developed specifically for translating Portuguese sign language into text. Due to time constrains, we narrow the problem into translating only the alphabet. Our architecture makes use of a model from Mediapipe, that recognizes hand landmarks and its coordinates given an image, and a MLP for the classification. We also apply feature generation to the coordinates before feeding them to the classification model. We created our own dataset by taking images of ourselves and applied data augmentation techniques. The experiments show that our solution offers a good performance in the given dataset and in real-time scenarios.**

***Index Terms*—Sign language, Machine Learning, MLP, Hand Landmarks, Portuguese Sign Language**

I. Introduction

Communication is said to be the key to success, but what about people who cannot communicate like others? There are about 70 million [1] deaf people in the world. They usually communicate with each other using Sign Language and, since they cannot speak, there is a communication gap between deaf and normal hearing people.

This communication gap is not restricted to deaf and normal hearing people, since there are at least 300 different Sign Languages [1] and each one is different from another.

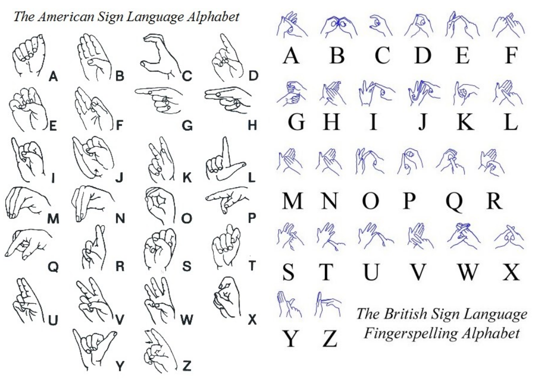
The American Sign Language Alphabet
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z
The British Sign Language Fingerspelling Alphabet

Fig. 1. Comparison between ASL, on the left, and BSL, on the right

Currently, the responsibility of bridging this gap falls upon the deaf individuals themselves. The population proficient in Sign Language remains relatively small since many perceive no personal advantage in learning it. As a result, the significant disparity between the number of people who know Sign Language and those who do not intensifies the likelihood that, when these two groups interact, they must mutually adapt. Typically, it is the deaf person who takes the initiative by employing a smartphone or even an interpreter to facilitate communication.

However, why should this be the norm and why can we not be the ones who try to help others?

To bridge this gap, our goal is to develop a model that can efficiently translate Sign Language into text. By enabling deaf individuals to express themselves in their native language, we aim to significantly reduce communication barriers and promote inclusive interactions.

To better define what can be done to build this model, we must first know what is Sign Language.

First, it is important to emphasize that nearly every country has its own Sign Language, and the majority of these Sign Languages are not mutually intelligible [9]. A great example of this is the American Sign Language and the British Sign Language. Despite English being the official Spoken Language in both countries, these two sign languages possess unique characteristics on their own. Taking a closer look at Figure 1, which displays the distinct alphabets of each language, it becomes evident that American Sign Language (*ASL*) and British Sign Language (*BSL*) are entirely different. ASL utilizes a single hand, whereas *BSL* employs both hands, as depicted in the visual representation.

Second, a Sign Language is not bound by the structure of Spoken Languages. In the case of Portuguese Sign Language, it originated from Swedish Sign Language rather than being directly derived from Spoken Portuguese. As a result, the sentence structure can differ between the two languages. This variation is evident in Portuguese Sign Language, where expressing someone's British nationality involves combining three gestures representing "Person," "born," and "England".

Lastly, Sign Language goes beyond mere hand gestures, just as Spoken Languages encompass more than words. Just like in Spoken Language, when altering the tone of voice, speed or facial expressions can give different nuances to the same word, Sign Language possesses similar capabilities. For instance, consider the case of anger in communication. When an individual who can speak is angry, they tend to speak louder and faster. Similarly, in the case of a deaf person experiencing anger, their movement speed increases and their gestures become more pronounced and sharp.

Therefore, a skilled interpreter must consider a wide range of information transmitted by the Sign Language user. This includes factors such as head position, facial expressions, hand movements, and even mouth movements (known as *mouthings*), all of which contribute to conveying different meanings in Sign Language.

Having this in mind, we understand that there is a big way to automatize the translation of Sign Language to a Spoken Language, and vice versa. A perfect model would have to capture the face, hands and posture of the Sign Language user and process all that information to give an accurate translation in real-time. To simplify our problem, we can use a *"Divide to Conquer"* approach and decided to start by trying to translate the information transmitted by the hands.

To achieve this, there are different technologies capable of gesture recognition. Models like CNN's [5], LSTM's [6] and pre-trained models (VGG16 [3], ResNet [4]), that have these architectures as theirs cores, are good examples for this task.

In this paper we propose a method to translate the Portuguese Sign Language Alphabet only taking in account hand signs.

Our solution makes use of models that are capable of recognizing hands and a simple Machine Learning model, namely Multi Layered Perceptron. Additionally, we explored an alternative approach known as Transfer Learning to tackle this task, and we will delve into the results at a later point.

II. Related Work

With the recent developments in Machine Learning, there have been different proposals on how to solve Hand Sign Translation.

Having in mind this problem involves receiving an image as input, there can be two main strategies:

A. Feature engineering + Classifier

The conventional approach involves manually extracting features (Feature Extraction) from the provided image by employing various techniques such as hand landmarks recognition, background removal, image cropping, and more. Additionally, new features may be derived from these existing ones to enhance or introduce new characteristics (Feature Generation). Following this process, a classifier is trained to comprehend all these features. The classifier can be any type of Machine Learning model. Usually, the most used are:

- Support Vector Machines (SVM's hereinafter) [7] are a great choice when a problem is linearly separable. Even if that is not the case, they can be used in non-linear problems using the kernel trick [18].
- Multi-Layer Perceptrons (MLP's hereinafter) [8] is a fully connected class of feedforward artificial neural network. This model can solve linear and non-linear problems.

Several papers have adopted this approach, with the most prevalent method involving the utilization of a model capable of hand landmark recognition, followed by training another model, such as SVM and MLP, to comprehend these extracted features.

This approach suffers from the drawback that the classification process is heavily reliant on the quality of the employed feature extraction and feature generation methods. Moreover, there exists a trade-off between performance and the level of complexity involved in feature engineering.

B. Deep Learning

With the progress of Machine Learning, it has become increasingly viable to integrate feature engineering with classification. In other words, allowing the model to learn the feature extraction process autonomously has rendered it more dependable.

There are two approaches that are relevant to our problem:

1) CNN's: CNN's are usually used to learn specific details in images, and there is no surprise that they are vastly used in this problem.

A CNN is composed with convolution layers intercalated with pooling layers. The convolution layers are responsible to extract the features from the image and the pooling are a way to decrease the number of parameters of the model. Due to these properties, CNN's are a perfect candidate to solve this problem, since they can extract features from the images and use them to predict unseen data.

2) Transfer Learning: Transfer Learning [19] is a technique that transfers the knowledge acquired from a task to a new one. This knowledge is usually acquired from a large and general dataset so the model's weights are trained properly. There are examples of Transfer Learning in this area, making use of big pre-trained models like VGG16 and ResNet, that have CNN's in their core. Both these models were developed to participate in ImageNet contest [10], where they got impressive results.

These models can be fine-tuned to adapt to a new task, *i.e.*, training the model again making small changes to it, so it performs better in the new task. As these models are good at detecting and classifying objects of 1000 different classes, they have been used in problems similar to ours.

III. Dataset

In this project our main limitation was not having a previously done dataset. Consequently, we had to generate our own.

We took about 1500 photographs of each letter and proceeded to apply data augmentation techniques, such as: random contrast [11], random brightness [12] and random rotation [13].

Bear in mind that we are no experts in this field, and as a result, there may be gestures that were done incorrectly.

The dataset is also very similar (between each letter), since we took every photograph in one session. This factor probably influenced our final results.

IV. Proposed Methodology

In this section, we present our model to be used in Portuguese Hand Sign Translation. The solution consists of 3 phases:

- Feature Extraction - Involves extracting features from an image
- Feature Generation - Additional features are generated.
- Classification - The model classifies the input based on the extracted and generated features.

In the following sections, we will provide a more detailed explanation of each phase.

A. *Feature Extraction*

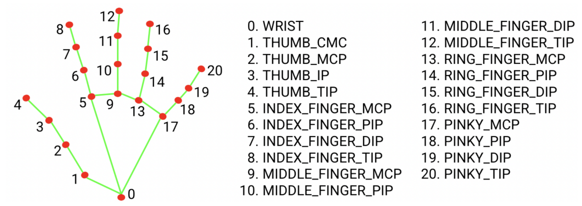

Fig. 2. MediaPipe hand landmarks

We get our features based in hand landmarks. These landmarks are key points of the hands that allow us to get a general idea on how the hand and fingers are positioned. This information is very precious since for obvious reasons.

There are different ways to get these points, but we opted for Hand Landmark from MediaPipe.

This model has been trained ”on approximately 30K real-world images, as well as several rendered synthetic hand models imposed over various backgrounds.” [2], making it very reliable. This model receives an image in RGB format and outputs a set of 21 points, as illustrated in Figure 2, representing the 3D coordinates of hand landmarks. These coordinates are normalized, ranging between 0 and 1, ensuring consistency across different hand sizes and positions.

B. *Feature Generation*

To get additional information and better distinguish between letters, we did a small feature generation relatively to the key points previously obtained.

The first generated feature was the distance between the fingertips and the wrist. This feature helps in a lot of cases, for instance, the distance between the index finger and the wrist is much smaller if the hand is closed in a fist than if the hand is totally opened.

The second feature is the angle of each finger joint. This gives a precise information about how closed a finger is.

These 2 features along with the coordinates captured by feature extraction, give a good and precise information about how is the hand sign being done.

C. *Classification*

In the classification phase, we feed these crafted features to a model that learns to differentiate letters based in the information inputted.

This part could be done using different models, such as MLP, SVM, Decision Trees [20], Random Forest [21], and more. Consequently, we experimented some of these models to identify which one fits the best for our problem considering the captured features.

TABLE I
RESULTS IN PERCENTAGE OBTAINED USING TRANSFER LEARNING

	Accuracy
VGG16 with 1 FCC	13
VGG16 with 3FCC	12
VGG16 with 1FCC (1/3 of Base Model frozen)	10
VGG16 with 3FCC (1/3 of Base Model frozen)	8
ResNet50 with 3FCC (1/3 of Base Model frozen)	8

V. RESULTS

We experimented with various combinations of feature generation techniques and the application of Transfer learning, as previously mentioned. The outcomes of these experiments are presented in Table I and II, providing a clear observation of the results.

A. *Configuration*

In the next sections, we will describe the configuration of each model tested in both approaches.

1) Transfer Learning: As mentioned previously, we made use of two pre-trained models, VGG16 and RESNet50, both utilizing CNN architectures.

We experimented involving various combinations of the following strategies:

- Replacement of the last fully connected layer.
- Replacement of all fully connected layers.
- Freezing all layers with the weights from VGG/ResNet.
- Freezing one-third of the layers with weights from VGG/ResNet.

We ran all models for a mere two epochs, which proved sufficient to understand the progression of the model.

These models underwent training using the train dataset, with 10% of the data reserved for validation. Subsequently, they were evaluated using the test dataset.

2) Feature Extraction + Classification: The fundamental feature generation techniques we experimented are the following:

- Distance between fingertips and wrist (Distance).
- Angle between each joint (Angle).
- Normalization of the coordinates with respect to the wrist (Norm Wrist).

The models we tried for classification are all from *SKLearn* [14] library and have the following configuration:

- Support Vector Classification (SVM) [15] - The ”gamma” parameter was set to ”auto”. Before inputting the features into the model, we applied standardization using the *StandardScaler* from the *SKLearn* library. All other parameters were set to default.
- Multi-Layer Perceptron (MLP) [16] - It consisted of two hidden layers, each containing 100 neurons. The model was trained with a maximum of 300 iterations and utilized early stopping. All other parameters were set to default.

TABLE II
RESULTS IN PERCENTAGE OBTAINED USING FEATURE EXTRACTION AND FEATURE GENERATION

	SVM	MLP	DT
Distance	71	88	46
Distance & Angle	70	88	49
Distance & Norm Wrist	66	84	59
Norm Wrist	61	88	53
Norm Wrist & Angle	64	74	59
Angle	63	70	37
None	62	71	28
All	67	79	64

- Decision Tree (DT) [17] - All parameters were set to their default values.

The models were trained with the train dataset and were evaluated using the test dataset. The MLP model utilized 10% of the train datasest for validation purposes.

B. *Discuss of Results*

The results obtained in Transfer Learning really surprised us. We were expecting this approach to be the best one, since the model was already trained with a large dataset, and with some fine tuning it would adapt almost perfectly. Unfortunately, this was not the case. We think the bad outcome can be explained by the the poor dataset. The images were all alike within each letter, which cause the models to predict wrongly when a hand sign is done slightly different. We would like to test this model in a good dataset and confirm our theory. It is also interesting the fact that the ResNet50 had identical results in comparison with the VGG16. We were not expecting it, due to the significantly different accuracy obtained in the ImageNet contest [10].

The Feature Generation approach also surprised us. We were hoping for good results, but not this good. Since the results were this positive, we would like to delve into exploring new feature generation techniques that are more informative. However, it is worthy to mention that more is not necessarily better, as we can see that the best results were with only the distance feature.

The reason why the approach utilizing all the generated features did not yield the best results for the same model could be attributed to the model's simplicity in comparison to the abundance of features. As the number of features increases, so does the dimensionality, which can pose challenges for the model. Furthermore, it would be worthwhile to try to increase the complexity of the model to assess its impact on performance.

Regarding the classifiers, we can observe that the Decision Trees are not good to classify based on these features. We can also observe that the MLP has perform way better than the SVM. This can be a evidence that the problem is likely Non-linear.

VI. CONCLUSION

In this paper, we present a model for translating the Portuguese Sign Language Alphabet to text. Our model incorporates three essential components: Feature Extraction, Feature Generation, and Classification. Initially, hand landmarks are extracted from an image, followed by the application of various strategies to enhance these features. Finally, the model predicts the corresponding text based on these features. The model demonstrates strong performance in this task, achieving an accuracy of approximately 88% on the test dataset. Moreover, a notable advantage of this approach is its lightweight computational requirements, allowing real-time translation.

VII. FUTURE ENHANCEMENTS

As mentioned extensively throughout the paper, there are several areas we would like to improve in order to develop a more dependable solution.

Firstly, as said before, a more diverse dataset from experts is mandatory to get better and reliable results. It would be great if the dataset would be split in train and test, made by different individuals.

Second, our model is designed to only recognize static signs, whereas Sign Language is much more than that. Many words are made with movements, and each movement can be done differently to transmit different feelings. Additionally, facial expressions are used to communicate by everyone, but it has a special important role in this type of languages. Therefore, we would like to train a model capable of not only detecting and recognizing different emotions through facial expressions, but also performing a temporal analysis to comprehend gestures and the way they are done. To achieve this goal, it is likely necessary to use more complex models that can account for temporal analysis, such as LSTM's and Transformers.

REFERENCES

[1] https://www.un.org/en/observances/sign-languages-day
[2] https://developers.google.com/mediapipe/solutions/vision/hand_landmarker
[3] https://datagen.tech/guides/computer-vision/vgg16/
[4] https://datagen.tech/guides/computer-vision/resnet/
[5] https://datagen.tech/guides/computer-vision/cnn-convolutional-neural-network/
[6] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[7] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[8] https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141
[9] https://en.wikipedia.org/wiki/Mutual_intelligibility
[10] https://www.image-net.org/challenges/LSVRC/
[11] https://www.tensorflow.org/api_docs/python/tf/image/random_contrast
[12] https://www.tensorflow.org/api_docs/python/tf/image/random_brightness
[13] https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomRotation
[14] https://scikit-learn.org/stable/
[15] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC
[16] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier
[17] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[18] https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f
[19] https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/
[20] https://www.ibm.com/topics/decision-trees
[21] https://www.ibm.com/topics/random-forest